

Bauhaus-Universität Weimar  
Faculty of Media  
Degree Programme Computer Science and Media

# Visualizing and Editing the History of Countries in Time and Space with HistoGlobe

## Master's Thesis

Marcus Kossatz  
b. 21.08.1989 in Spremberg, Germany

Matriculation Number 90487

1. First Referee: Univ.-Prof. Dr.-Ing. habil. Volker Rodehorst
2. Second Referee: Prof. Dr. rer. nat. Sven Bertel

Submission date: 10. June 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Motivation . . . . .	10
1.2	Problem Domain . . . . .	10
1.3	Overview. . . . .	13
<b>2</b>	<b>Basics</b>	<b>14</b>
2.1	Countries . . . . .	15
2.1.1	Problematic Cases. . . . .	15
2.1.2	Declaratory vs. Constitutive Theory . . . . .	17
2.1.3	Territory of a Country. . . . .	18
2.2	Historical Geographic Information Systems . . . . .	19
2.2.1	History vs. Geography . . . . .	20
2.2.2	Geospatial Data . . . . .	22
2.2.3	Temporal Data. . . . .	26
2.2.4	Existing Applications . . . . .	28
2.2.5	Data Sources. . . . .	28
2.3	Spatio-Temporal Data Models . . . . .	30
2.3.1	Snapshot Model . . . . .	31
2.3.2	Simple Time-Stamping . . . . .	31
2.3.3	Event-Based Spatio-Temporal Data Model. . . . .	32
2.3.4	Three-Domain Model . . . . .	33
2.3.5	History Graph Model . . . . .	33
2.3.6	Spatio-Temporal Databases . . . . .	35
2.4	HistoGlobe . . . . .	37

<b>3</b>	<b>Development</b>	<b>39</b>
3.1	Hivent Model . . . . .	41
3.1.1	Elements . . . . .	41
3.1.2	Preconditions . . . . .	42
3.1.3	Hivent Operations . . . . .	44
3.1.4	HistoGraph . . . . .	46
3.2	Editing Hivent Data . . . . .	48
3.2.1	Edit Operations . . . . .	48
3.2.2	Edit Workflow . . . . .	49
3.2.3	Retrospective Updates . . . . .	52
3.2.4	Backward Operations . . . . .	59
3.3	User Interface Design Process . . . . .	60
3.3.1	Paper Prototype . . . . .	61
3.3.2	Mockup Prototype . . . . .	62
3.3.3	Web-based prototype . . . . .	63
3.4	Implementation . . . . .	66
3.4.1	System Architecture . . . . .	66
3.4.2	Server-Side Application . . . . .	67
3.4.3	Client-Side Application . . . . .	69
<b>4</b>	<b>Extensions</b>	<b>78</b>
4.1	Evaluation . . . . .	78
4.1.1	Analysis of the Data Model . . . . .	79
4.1.2	Analysis of the Application . . . . .	82
4.2	Modelling Uncertainty . . . . .	83
4.2.1	International Recognition . . . . .	84
4.2.2	Uncertain Borders . . . . .	84
4.2.3	Special Areas . . . . .	86
4.2.4	Content from Wikipedia . . . . .	87
4.2.5	Extended Hivent Model . . . . .	89
<b>5</b>	<b>Summary</b>	<b>91</b>
5.1	Results . . . . .	91
5.2	Problems . . . . .	92
5.3	Prospect . . . . .	92

# List of Figures

1.1	The snapshot approach for modelling time and space . . . . .	11
1.2	The event-based approach for modelling time and space . . . . .	12
2.1	The Declaratory Theory (left) and the Constitutive Theory (right) of Statehood . .	17
2.2	Enclaves and exclaves <sup>1</sup> . . . . .	18
2.3	Signs, data, information and knowledge . . . . .	19
2.4	The geoid model, differences are exaggerated, [Bol08, Fig. 3-6, p. 75] . . . . .	22
2.5	Geographic coordinates using latitude and longitude . . . . .	23
2.6	The basic geometric primitives point, polyline and polygon . . . . .	24
2.7	Types of polygons . . . . .	24
2.8	Boolean set operations on polygons . . . . .	25
2.9	Comparison between equivalent, conformal and compromising map projections . . .	26
2.10	Comparison between a linear, a logarithmic and an irregular timeline . . . . .	27
2.11	Semi-automatic extraction of a border from a map of the Roman Empire <sup>2</sup> . . . .	30
2.12	The Snapshot Model by [Lan88] . . . . .	31
2.13	The Simple Time-Stamping method by [HW90] . . . . .	32
2.14	The History Graph model . . . . .	34
2.15	Types of changes in the History Graph model . . . . .	35

2.16 The HistoGlobe School project (status: April 2015) . . . . .	38
3.1 Human Centered Design process with five project phases . . . . .	39
3.2 Relevant models for an information system . . . . .	40
3.3 Three event types that change Areas, resulting in two different area states . . . . .	42
3.4 The initial state of the world map at time point $t_0$ . . . . .	44
3.5 The concept of the HistoGraph at the example of the history of Germany since 1945	46
3.6 Example for a simple conflict due to a retrospective update . . . . .	52
3.7 Conflicts after a retrospective incorporation . . . . .	54
3.8 Conflicts after a retrospective secession . . . . .	55
3.9 Conflicts after a retrospective unification . . . . .	57
3.10 Conflicts after a retrospective separation . . . . .	58
3.11 The two iterations of the paper prototype for the Edit Mode . . . . .	61
3.12 Two iteration stages of the mockup prototype for the Edit Mode . . . . .	62
3.14 Initial state of the Browsing Mode . . . . .	64
3.15 Initial state of the Edit Mode . . . . .	64
3.16 1) Select old Areas . . . . .	64
3.17 2) Set a new territory . . . . .	64
3.18 3) Set a new name . . . . .	65
3.19 4) Add the Operation to a Hivent . . . . .	65
3.20 4) Create a new Hivent . . . . .	65
3.21 The final state with Scotland . . . . .	65
3.22 The system architecture of HistoGlobe . . . . .	66
3.23 The Hivent Database Model . . . . .	67

3.24 Visualization of the German Reunification in the Hivent Database Model . . . . .	68
3.25 Class Diagram of HistoGlobe . . . . .	70
3.26 Detecting the next Hivent that happens in the HiventController . . . . .	71
3.27 The Within-Tree (left) for the set of polygons (right) . . . . .	74
3.28 The resulting labels on the map in Europe 2016 . . . . .	76
4.1 The difference between accuracy and precision . . . . .	79
4.2 Newly designed and extended buttons for Edit Operations. . . . .	84
4.3 New Edit Operation: Recognition – sets up the recognition of one Area to another. .	84
4.4 Three different methods to visualize uncertain courses of a border . . . . .	85
4.5 Drawing historical borders instead of full Areas and defining a level of certainty. .	86
4.6 Defining a special status or relationship to a territory. . . . .	87
4.7 Getting suggestions for the name name from Wikipedia. . . . .	87
4.8 Creating a new Hivent and adding the newly created Edit Operation. . . . .	88
4.9 The new data model to support the developed approaches regarding uncertainty . .	89

# List of Tables

2.1 Differences between history and geography [KH08, pp. 2-4] . . . . .	21
3.1 The five Hivent Operations . . . . .	45
3.2 Historical events in German state history since 1945 . . . . .	47
3.3 The six Edit Operations . . . . .	48
3.4 The requirements of each step for the Edit Operations . . . . .	49
3.5 Translation from Edit Operations to Hivent Operations. . . . .	51
3.6 All possible conflicts on retrospective updates regarding their resolvability . . . . .	59

## **Selbstständigkeitserklärung**

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig und nur unter Zuhilfenahme der angegebenen direkten und indirekten Quellen erstellt habe. Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keinem anderen Prüfer als Prüfungsleistung eingereicht und ist noch nicht veröffentlicht.

## **Statement of Authorship**

Hereby I declare that I completed this Master's Thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this, nor a similar work, has been published or presented to an examination committee.

Weimar, 10. June 2016

---

Marcus Kossatz

## Acknowledgements

The work on this Master's Thesis began in October 2015 and ended in June 2016. I want to thank my supervisors Prof. Volker Rodehorst and Prof. Sven Bertel for the continuous advices, ideas and the motivation to work on this topic of my choice.

The majority of the time of this thesis (October-December 2015 and February-May 2016) I have spent working at the *Scholars' Lab* of *University of Virginia*. I was provided with an excellent workspace, very cooperative people and delicious oatmeal raising cookies that all contributed to a great work atmosphere in Charlottesville, Virginia. I especially want to thank all the members of the *Scholars' Lab* that were always eager to solve my problems regarding the concept and the implementation of this thesis, espacially Ammon, Eric, Scott, Purdom, Jeremy, Chris, Wayne, Ronda, Laura, Becca, Shane and Veronica. The perspective of humanity scholars on my project were very valuable for me personally and for my work. Thank you!

Since this thesis was an advancement of HistoGlobe, the project I have founded in 2010, I want to thank everybody that has ever been involved in this project. In the previous 6 years I have learned a lot about software engineering, history and geography, project management and entrepreneurship – and how to (not) make a project successful that almost everybody likes. Special thanks goes to Frej which whom I developed the first prototype in 2011. From 2012 on, my coworkers, fellow students and friends Simon and Felix deserve great credit for everything they have done to HistoGlobe and me. Without the software they have developed until 2014 the result of this thesis would be much different. I also want to thank the other developers that have spent their free time, ideas and coding skills on the HistoGlobe project: Sascha, Sebastian, Lene, Max, Chris and Sebastian. Thank you Christian for always burning for HistoGlobe and trying to make the project profitable. Im also thankful for Carola and Katrin who have spent significant free time on HistoGlobe in 2013. Last but not least a huge Thank You to my friend and colleague Tobias for your high efforts for the HistoGlobe project, great concepts and stunning designs. I have learned from you how valuable good design is.

Neither HistoGlobe, this thesis, nor both of my academic degrees would be possible without *Bauhaus-Universität Weimar* and all the associated professors and academic staff at this great instiution. Thank you for the support throughout the previous seven years. The history and the values of the Bauhaus have always inspired me and are a big part of how I think, develop and create. It has been a great time in Weimar.

Finally, I want to thank my family that has always been there for me in the first 26 years of my life. May it be my studies, my three foreign exchanges or my entrepreneurial idea – my parents, my brother, my sister and the rest of my family have always supported me. That feeling is invaluable.

# Chapter 1

## Introduction

*Imagine there's no countries  
It isn't hard to do  
Nothing to kill or die for  
And no religion too  
Imagine all the people  
Living life in peace*

– John Lennon, Imagine (1971)

The song of John Lennon is an anthem for peace on Earth, for brotherhood of people, for the end of materialism – but also for the end of countries. He connects the concept of a country to nationalism that encourages people to fight and die for. John Lennon wrote the song in the 1970s, in the midst of the Cold War between the capitalistic and the socialistic bloc, only 30 years after World War II and 50 years after World War I. Especially in Europe this time point would have probably not been described as “peaceful”. And not just John Lennon connected this lack of peace with the existence of national states divided by artificial borders.

Now, another 45 years later, the situation in Europe looks much different: Most countries are united in a confederation of a largely shared economy. While there are still countries with clearly defined borders, they are mostly of legal nature, but citizens of the European Union can travel freely within large parts of Europe. This concept is celebrated as a major achievement, but it is mostly forgotten that the concept of nations with solid borders has not been there 200 years ago. While travelling back then was probably not as pleasant as it is today, Goethe at least did not need a passport when he travelled to Italy and back to Weimar. He also did not travel from country “Germany” via “Austria” to “Italy”, but he rather crossed several duchies and principalities that do not exist anymore.

## 1.1 Motivation

What we might call “our country” today has changed a lot in the past. Hardly any of the current 193 member states of the United Nations are in their same border as 100 years ago. The countries have evolved in time and space. Would it not be nice to see this development? Would it not be nice to have a map that shows the state of the world at any point in history? With that map we can see how our country looked like 100 years ago. 200 years ago. 1000 years ago. We could see how settlements became cities and principalities became national states. While there are many historical sources describing one point in history, may it be governmental bills, historical maps or diary entries of kings, there is no such thing as a comprehensive historical world atlas that lets you travel back in time and space and explore *when* our country changed, *where* it changed – and most importantly *why*? This thesis is all about that: How can the historical development of countries be shown, for the benefit of a better understanding of how we became what we are today.

This is a very complicated undertaking, given that countries have changed frequently. But there are severe conceptual problems: How do we know how a country has looked like in 1600? And if we find an historical map of this time, can we trust it? How certain can we be that the countries and their borders are true? The next problem is that the history of countries can also be contradictory. There is not always *one story* which is supported by all sides. There are contested territories, even today, from which it is not clear who they belong to. There are “places”, even today, which are not clearly a “country” because some might disagree. There is a whole lot of uncertainty and disagreement in the history of countries that this thesis deals with.

Finally, the state of the world can not be visualized at any point in history just like this – because there is no freely available dataset. It is not just a visualization problem, it is a data problem. And to go even further: It is a data model problem because it is not even straightforward to say what kind of information is actually necessary to show the history of countries. And if we found a data model and got some data, we do not want to write it into a database table. The third goal for this thesis is to develop a well-designed user interface to edit the history of countries directly on the map.

## 1.2 Problem Domain

*All human actions takes and makes place.*

*The past is the set of places made by human action.*

*History is a map of these places.*

*The past thus exists not in time but in space.*

– Philip J. Ethington in [Eth07, précis]

*Time* and *space* are everywhere. They are highly related to our lives and the objects we perceive. The temporal perception of the world is driven by events, may they be personal life events like a wedding or world events like the Fall of the Berlin Wall. While a point in time can be described by a date and a time stamp, it is not always easy to scale and grasp. This is mainly because some temporal developments happen suddenly, like a natural disaster, and some happen very slowly throughout years, decades or even centuries, like climate change. Time is not tangible. For space, the situation is different, because it can be perceived as physically existing: A place is just there, we can go there and see it. Each point on this planet can be exactly described by a pair of geographic coordinates.

The combination of both concepts in one information system would allow to say how something has developed in time and space. *Geographic Information Systems* (GIS) allow to manage and visualize data with a spatial relation to the Earth, mostly on a map. Most GIS answer two basic questions about an object: *Where* it is in relative or absolute location and *what* it is – an object with certain properties. As an example, a country can be expressed by a set of borders consisting of border points in geographic coordinates and by a name. However, most of the current GIS are limited to the spatial dimension. They can not answer to the question *when* a country was founded or how its borders have developed in the previous fifty years. For that purpose ***Historical Geographic Information Systems*** (HGIS) were developed. They extend general GIS with the dimension of time.

There are several *spatio-temporal data models* that deal with the temporal development of spatial objects. The straightforward approach immediately derives from the concept of historical maps: At certain time points a *snapshot* is taken: a map showing the current state at this point in time. Snapshots can immediately answer the question how the world has looked like at this date. However, they fail to answer the next question: What has changed since last time, when and why? Given two historical maps of Germany, one at 1871 after the formation of the German Empire and one 1919 after the Treaty of Versailles – how did it look like at the beginning of World War I in 1914?

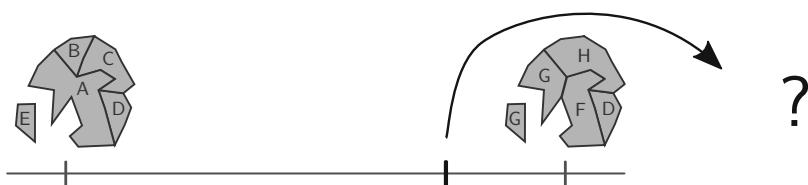


Figure 1.1: The snapshot approach for modelling time and space

As it is seen in figure 1.1, this is impossible to say, because there is no information about an arbitrary time point between two snapshots. Also, if the map only shows Germany and its neighboring countries, what about Russia, Sub-Saharan Africa or South East Asia? For an interactive historical world atlas, the snapshot approach is neither suitable nor feasible, because it requires a whole new world map every time some country changes on Earth.

The key problem is that snapshots can not say what has changed, because they do not store changes. This is the approach of another class of spatio-temporal data models: *Event-based* models. They store two things: one reference snapshot and a set of events that happen at a certain time point and trigger changes on the map relative to the last event.

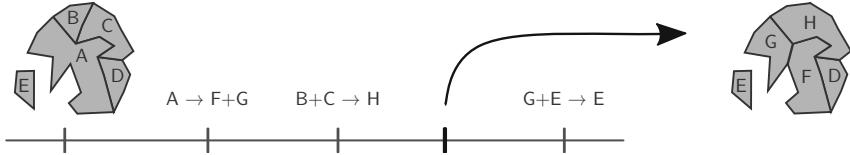


Figure 1.2: The event-based approach for modelling time and space

Figure 1.2 shows an example of an event-based approach. There are three changes that consecutively happen to the snapshot on the left. The question how the world looked like at an arbitrary point in time can be answered like this: It is the state of the reference map and all the changes of all events since that time point accumulated. In the case of figure 1.2, country *A* split up into *F* and *G* and *B* and *C* have unified to *H*. This approach is suitable for modelling a countries history, because each change to the state of a country was introduced by some historical event, may it be a declaration or a peace treaty.

**Research Questions** The goal of this thesis is to lay a theoretical foundation for an Historical Geographic Information System that deals with the development of countries in time and space. The domain is limited to countries, their names and their borders and historical events that change them. In addition, the model is to be developed in *HistoGlobe*, open-source Web-based application that aims to visualize the course of history. It should provide a well-designed user interface for editing historical data about countries allowing the user to directly manipulate the countries on the map.

For that matter there are three research questions to be answered throughout the thesis:

1. What type of historical changes can happen in the development of countries in time and space?
2. How can these changes be
  - (a) modeled in an information system?
  - (b) edited by humans in a user interface?
3. How can the model handle uncertainty and disagreement in history?

## 1.3 Overview

The remaining part of this Master's thesis is structured in four chapters. The second chapter introduces the basic concepts of the problem domain: First of all the surprisingly difficult concept of a *country* is introduced in section 2.1. Afterwards the term *Historical Geographic Information Systems* is clarified in section 2.2), followed by state of the art of *spatio-temporal data models* (2.3). The last section 2.4 presents *HistoGlobe*, the application that the work of this thesis is developed in.

Chapter 3 is the main part of this thesis. It describes the Human Centered Design process to answer the first two research questions: The *Hivent Model* in section 3.1 introduces a set of five *Hivent Operations* that can model all possible changes to the development of countries in time and space. The next section 3.2 presents approaches to edit data in the Hivent Model using *Edit Operations*, a different set of operations that is well-understood by humans. The iterative design process of the user interface is illustrated in section 3.3. The chapter closes in 3.4 with an insight into the implementation of the data model and the user interface in the HistoGlobe application.

The fourth chapter tackles the last research question: How to deal with uncertain historical data? The model and the implementation of chapter 2 are evaluated in section 4.1. From the identified shortcomings the thesis developed approaches to handle some aspects of uncertainty regarding the historical development of countries. The designed extensions are presented in section 4.2.

In the last chapter the work of the thesis is summarized.

# Chapter 2

## Basics

The title of this Master's Thesis is:

***Visualizing<sup>1</sup> and Editing<sup>2</sup> the History<sup>3</sup> of Countries<sup>4</sup> in Time and Space<sup>5</sup> with HistoGlobe<sup>6</sup>***

This chapter will form the theoretical foundation to understand these terms and present related work to the topic of this thesis.

<sup>1</sup> The purpose of a *visualization* is to present information to a human in a comprehensible way. The visualization is one component of an *Historical Geographic Information System* introduced in section 2.2.

<sup>2</sup> An information system can allow the user to modify, correct or generally *edit* the information in the system in order to improve it.

<sup>3</sup> History is the study of our past, to understand the present and reason about the future. Its main ideas are introduced in section 2.2.1.

<sup>4</sup> A *country* is probably a political entity with a clearly defined territory, a permanent population and a government. But as section 2.1 shows, its definition is surprisingly difficult.

<sup>5</sup> The work in this thesis focuses on data models of *time and space*. Existing *spatio-temporal data models* are discussed in section 2.3.

<sup>6</sup> The chapter closes in 2.4 with a presentation of *HistoGlobe*, a Web-based Historical Geographic Information System in which the data model of this thesis was developed in.

## 2.1 Countries

Almost everybody in the world is familiar with the term “country”, because almost everybody has at least one home country he or she can potentially hold a passport from. However, the reality is very complex. If the information system of this thesis deals with countries, it must be possible to decide for each current and historic political entity in the world if it is or was a country or not. This requires a clear and non-conflicting definition of a country. This section will show that this is impossible.

The Oxford Dictionary reads as follows: “The *territory* of a *nation*; a *region* constituting an *independent state*, or a region, province, etc., which was once independent and is still distinct in institutions, language, etc.”<sup>1</sup> This definition includes many different concepts and terms: the territory or region that the country is on, a nation or state, a population and a culture of the territory in terms of institutions or languages. *Nation* and *state* are commonly used as synonyms for countries.

To understand what a country really is, it is helpful to consult the United Nations, an intergovernmental organisation found in October 1945 that promotes international peace keeping, security, protection of human rights. The committee currently has 193 full member states and two permanent observers [Uni]. But these 195 members do not cover all places in the world – and also a membership in the United Nations does not guarantee being a undisputed country.

### 2.1.1 Problematic Cases

Examining the list of the UN member states yields several special cases, which can be classified by their membership status in the United Nations and their degree of international recognition.

**UN observer states** The *Holy See* is the juridical and spiritual entity representing Vatican City. It is a fully recognized and sovereign state but not a full member of the UN, because it has never applied. It is the by far smallest sovereign state in the world ( $0.44 \text{ m}^2$ ), an enclave inside the city of Rome with a population of only 800 people, including 30 women [Vat].

The *State of Palestine* with a population of 4.8 million people [Pal, as of 2016] has a totally different situation, because it does not have a clearly defined territory. The West Bank, East Jerusalem and the Gaza Strip were created in the 1949 Green Line Armistice Agreement but were never intended to be used as international boundaries [Amn]. Moreover, while 114 states officially recognize the Palestinian state, almost all current main economic powers do not, including the Germany and the United States. Unlike the Holy See, Palestine is not a fully sovereign and recognized country.

---

<sup>1</sup> *country*, Oxford Dictionary, URL: <http://www.oed.com/view/Entry/43085?>, last access: 25.04.2016

**UN non-members with limited recognition** Kosovo declared independence from Serbia in 2008. It has a clearly defined territory and a permanent population and is recognized by 111 UN member states. In order for Kosovo to become a full member of the United Nations, all permanent members of the security council (United Kingdom, France, Russia, China and the United States) must agree. But since Russia and China strongly support the territorial integrity of Serbia, they would veto Kosovos membership. Therefore, Kosovo is not even a UN observer state, although having about the same degree of international recognition as Palestine [Peo].

The status of *Taiwan* is a very complicated issue. Two territories and two political entities are involved in the conflict: The *People's Republic as China* (commonly known as China) has full control over mainland China, and the *Republic of China* governs the island of Taiwan. The problem is that both states claim the exact same territory: whole China. Since 1971 the People's Republic of China is the only representative of China in the United Nations, including the island of Taiwan. It is part of the Security Council and can successfully veto membership requests of the Republic of China. However, Taiwan operates like an independent country by international standards: They have an own jurisdiction, issue own passports and have unofficial diplomatic relations to most countries in the world. Officially only 22 UN members uphold diplomatic relations to Taiwan [Rep]. To all of these states the People's Republic of China does not uphold any diplomatic relations.

There are other places with limited international recognition: the Sahrawi Arab Democratic Republic (recognized by 84 UN member states) [Wes], Abkhazia (6) [Glo], South Ossetia (5) [BBCc], the Turkish Republic of Northern Cyprus (1) [Leo15], Nagorno-Karabakh Republic (0) [BBCa], Transnistria (0) [Gut14] and Somaliland (0) [BBCb].

**UN members with limited recognition** In addition to the Republic of China, there are five other member states of the United Nations that are not fully recognized by all other UN members: Armenia is not recognized by Pakistan [Tod], Turkey does not recognize the Republic of Cyprus [Eur]), North and South Korea mutually do not recognize each other [Dav] and the State of Israel is not recognized by 32 UN member states [Isr].

**Special Territories** There are also territories belonging to fully sovereign countries with a varying degree of sovereignty: Greenland is an autonomous country within the Kingdom of Denmark, but not a sovereign state. The same applies to numerous overseas territories of the United Kingdom, the French Republic or the Kingdom of the Netherlands. Moreover, there are five quasi-independent countries in a so called *Free Association*: Niue and Cook Islands are associated to New Zealand and not part of the United Nations. The Marshall Islands, the Federated States of Micronesia and Palau are associated to United States, but are in contrast full UN members [Won].

This incomplete and simplified list of special cases manifests the big problem that is associated with the terms "country", "state" or "nation": There is neither a *de jure* consistent definition nor a *de facto* consistent usage of these terms.

### 2.1.2 Declaratory vs. Constitutive Theory

Officially there are two different concepts that define what a country is: The *declaratory theory*, established in the Montevideo Convention 1933 [Yal], gives each entity the right to declare a state if it matches all of the four requirements: a clearly defined territory, a permanent population, a political representation / government and the *capacity* to enter diplomatic relations. These four requirements ensure that a state can exist physically and politically, independent from its recognition by others. In contrast, the *constitutive theory* requires exactly that: A state can only be considered as such if it is recognized by other states. However, it is not defined anywhere by how many other states [Law]. In short: "A country is a country when other countries think that country is a country." [CGP]

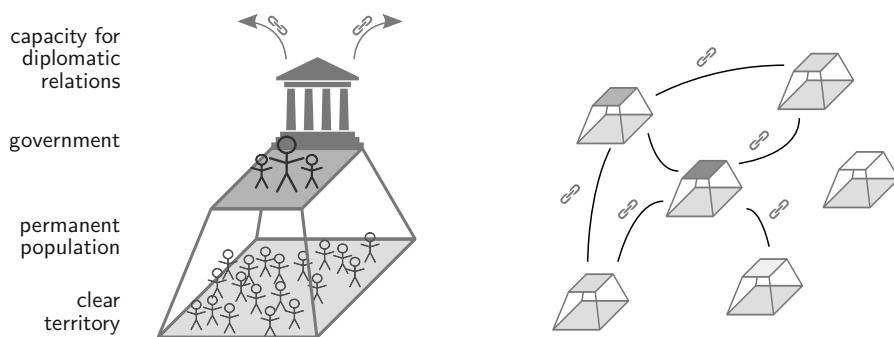


Figure 2.1: The Declaratory Theory (left) and the Constitutive Theory (right) of Statehood

Both theories have advantages and disadvantages, but the main problems are:

1. Following the declarative theory, countries are self-classifying and potentially conflicting entities. The application of this measure would grant Kosovo, the Republic of China and Abkhazia statehood. But this would lead to overlapping territories with Serbia, China and Georgia.
2. There is no super-national organization that can judge if a country is a country or not. Even the United Nations fail to do so, because their membership requirements prevent states like Kosovo or the Republic of China from becoming members.

Nobody can clearly say if Kosovo, Taiwan or Abkhazia are countries or not – and if they do, there will be people that disagree. This is a big problem for the information system, because it is impossible to objectively classify a place as a country or not. And this section only covers the current countries. 100 years ago these two theories were not existing. For the time before that, a conflict-free decision of what is country is not just impossible, but also not justifiable because of a lack of jurisdiction.

That means, the HGIS developed in this thesis inevitably deals with uncertain information that some parties disagree. Its data model can not perfectly fit self-classifying data and can not rely on an objective data source. The system has to contain approaches that deal with this problem.

### 2.1.3 Territory of a Country

The declaratory theory introduces four main properties of a country. The data model developed in this thesis focuses only the territory, since it is easy to present it on the map. It can just as well be a complex issue, as it has already been explained for the case of the State of Palestine in section 2.1.1. In the usual case, the three-dimensional territory of a country consists of the *landmass* it occupies, its *territorial waters* and its *controlled airspace*. This thesis considers only the two-dimensional part of the territory on the Earth's surface. It contains land, islands, inland water, river outfalls, and fjords.

The *land territory* is bounded by the countries *borders*. A border consists of a set of straight lines between clearly defined *border points*. A border is either *interior* when it directly borders the territory of another country on land or a *coastline* if it borders international water. In line with [Uni82], the territory of a country extends in a range of 3 to 12 miles (5 to 20 kilometers) into international waters and forms the *sea territory* of a country. The territory of a country is *continuous* if each point on the territory can be reached by any other point without leaving the territory. Non-continuous territories are usually the case if the country has islands. However, there is a special case to be considered:

If a territory is completely surrounded by the territory of exactly one country, it is an *enclave*. On the contrary, an *exclave* is a part of a country's territory that is geographically separate from the main part of the territory. In some cases the exclave of one country is automatically the enclave of another country – however, if the exclave is surrounded by more than one other country, it is not an exclave. Enclaves can be nested and form a hierarchy, e.g. a *second-order enclave* is an enclave within an enclave [Tho].

In the example in figure 2.2 *A* has three exclaves: *A*<sub>1</sub>, *A*<sub>2</sub> and *A*<sub>3</sub>. Only *A*<sub>3</sub> is also an enclave, within *B*. *E* is an enclave of *A*. Within *E*, there are two second-order exclaves for *A*: *A*<sub>4</sub> and *A*<sub>5</sub>. Inside *A*<sub>5</sub>, there is *E*<sub>1</sub>, a third-order enclave from the perspective of *A* and a second-order enclave for *E*. *D* is the only fully enclaved territory.

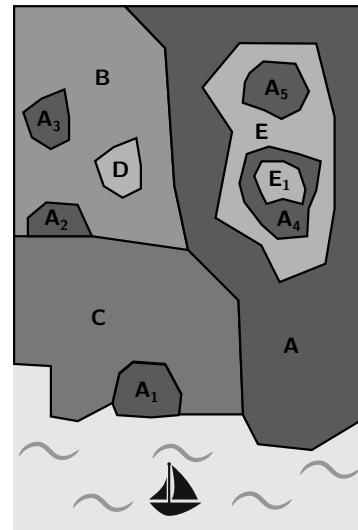


Figure 2.2: Enclaves and exclaves <sup>2</sup>

Second-order exclaves infrequently appear in the real world, as in the example of Baarle-Nassau and Baarle-Hertog at the border between the Netherlands and Belgium <sup>3</sup>. There are three countries in the world that are complete exclaves: San Marino, Vatican City and Lesotho.

<sup>2</sup> example based on: *Explicative diagram of territorial discontinuities: Enclaves and exclaves*, Wikipedia, [https://en.wikipedia.org/wiki/Enclave\\_and\\_exclave](https://en.wikipedia.org/wiki/Enclave_and_exclave), last access: 02.06.2016

<sup>3</sup> *The Curious Case of Baarle-Nassau and Baarle-Hertog*, kaushik, 06.11.2012, <http://www.amusingplanet.com/2012/11/the-curious-case-of-baarle-nassau-and.html>, last access: 31.05.2016

## 2.2 Historical Geographic Information Systems

A *system* is an organized structure containing *elements* that are directly or indirectly *related* to each other. At any point in the system's existence there is an *internal state* that changes when it gets influenced by stimuli of from the outside [Bus]. An *information system* (IS) is an application that is dealing with the acquisition, management, analysis and presentation of information [Zwa]. The terms "signs", "data", "information" and "knowledge" are sometimes used interchangeably and there is no coherent definition for any of them. However, all describe different concepts. This explanation seen in figure 2.3 is based on the work of [Dra].

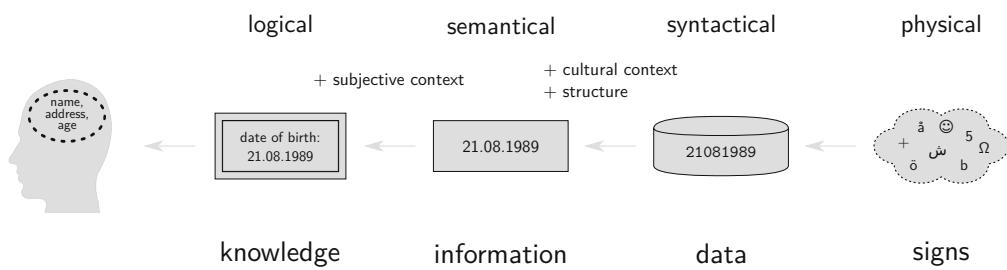


Figure 2.3: Signs, data, information and knowledge

A *sign* is the physical representation of something in the real world. Since the real world is continuous, literally anything can be seen as a sign, so there are uncountably infinitely many different signs. *Data* is a subset of all possible signs and represents the syntactical level of what an information system deals with. Data itself does not have any meaning, but as soon as it is organized, it becomes *information*. However, information is sensitive to its cultural context. The string `14.07.1789` is useful and understandable for people in countries that use the date format `DD.MM.YYYY`. However, for people in Belize and the USA this might just be a random string of numbers – it does not have any information although it is the same data. When a human understands the visualization of information he or she can integrate it into the larger subjective context, the *knowledge* [Nak]. The goal of a visualization is to present the information in a way that it can be transformed into knowledge by the viewer.

If the majority of the information in a system has a spatial relation to the Earth, its surface, its atmosphere or the social structure of its habitation, it is a *Geographic Information system* (GIS). The data objects in the system are called *geo-objects* [Bol08]. If the information additionally has a temporal dimension, e.g. via time stamps or time spans, which enable to trace developments of geo-objects, it becomes an *Historical Geographic Information System* [GG14] or alternatively *Spatio-Temporal Information Systems* (STIS) [PTKT04]. HGIS help scholars in *Digital Humanities* to analyze how "spatial patterns change over time in order to better understand large-scale Earth processes" [Peu99]. Their purpose is to situate "history in its geographical context and using geographic information to illuminate the past" [KH08, p. 3].

## 2.2.1 History vs. Geography

*La Géographie n'est autre chose que l'Histoire dans l'espace,  
de même que l'Histoire est la Géographie dans le temps.*

*Geography is nothing but History in space,  
the same way as History is Geography over time.*

– Élisée Reclus: “*L'Homme et la Terre*” (1908)

**History** Alike many other fields in humanities, *history* is “an ideal field for thinking long and hard about important questions” [AHA]. It originates from the Greek word *Iστορία / historia*: “finding out, learning through research, narration of what is learned” <sup>4</sup> and it signifies the two main modern usage forms of the term: To research about something and to tell a story. Historians interpret primary sources, such as written documents, photographs or historical maps to explain complex phenomena [KH08, pp.4-7]. The main goal of history is to study processes in the past to understand the situation in the present and make reasonable decisions for the future. The American Historical Association has developed the “five C's of historical thinking [that] together describe the shared foundations of [the] discipline” [AHA]:

**Change over time** The lives of people, their languages and cultures are continuously changing. One goal of history is to describe these historical changes triggered by historical events. Snapshots in the form of historical maps or historical photography are used to tackle this task.

**Context** The goal is to travel back in time to a moment in the past and recreate the world. Therefore it is crucial to understand the historical context via primary sources.

**Causality** The overall goal of each science is to answer the question *why* something is the way it is. Historians want to explain an historical event or process based on evidence. The problem is that history can not run experiments, because the same condition can not be repeated. Therefore historians have to focus on the interpretation of primary sources.

**Contingency** Each event has a network of prior conditions, because the world is highly interconnected. A slight change in one prior condition could have led to a completely different outcome of the event and a different state of the world.

**Complexity** The intrinsic human need for order conflicts with the contingency of history. It is questionable if all details about events in the world are scientifically explainable.

---

<sup>4</sup> *History*, Dictionary.com, based on Random House Dictionary, 2015, URL: <http://dictionary.reference.com/browse/history>, last access: 23.10.2015

**Geography** (Greek *γεωγραφία / geographia*) literally means “describing the earth.”<sup>5</sup> It is a science that studies the interplay between the landscapes and environments of the Earth (*physical geography*) on the one hand and the people, their cultures, societies and economies (*human geography*) on the other. It is an interdisciplinary field between natural and social sciences [RGS]. Geographical research aims to understand where things are found, why they are there and how they developed over time. It focuses on the interconnectivity between elements of physical and human geography, which gets expressed in Tobler’s First Law of Geography: “Everything is related to everything else, but near things are more related than distant things.”<sup>6</sup>

Geographers use different techniques to answer their research questions. One important technique is a *map*: A graphical expression of something that is not tangible. A map shows the physical, environmental, political, economical or social properties of the Earth. The “art and science of making maps” is the field of *cartography*.<sup>7</sup> Since maps visualize a model, they have a natural constraint: “No map can perfectly replicate the real world, since it inevitably generalizes, abstracts and approximates the complexity of the reality” [KH08, p. 181].

<i>geography</i>	<i>aspect</i>	<i>history</i>
where?	question	when?
space	dimension	time
exact, statistical	character	complex, fuzzy
mainly quantitative	research	mainly qualitative
spatial proximity of conditions	causal explanation	temporal sequence of events
spatial differentiation	explanation	temporal differentiation
clustering	organization principle	periodization
mostly visual (maps)	expression	mostly verbal (texts)
high (GIS)	digitalization potential	low (digital humanities)

Table 2.1: Differences between history and geography [KH08, pp. 2-4]

<sup>5</sup> *Geography*, Dictionary.com, based on Random House Dictionary, 2015, URL: <http://dictionary.reference.com/browse/geography>, last access: 23.10.2015

<sup>6</sup> “A computer movie simulating urban growth in the Detroit region”, Waldo Tobler, 1970 *Economic Geography*, 46(2): 234-240.

<sup>7</sup> *History of maps and cartography*, James S. Aber, URL: [http://academic.emporia.edu/aberjame/map/h\\_map/h\\_map.htm](http://academic.emporia.edu/aberjame/map/h_map/h_map.htm), last access: 24.10.2015

## 2.2.2 Geospatial Data

HGIS need to unambiguously locate geo-objects on, underneath or close to the Earth's surface using *geographic coordinates*. They express an object directly in the coordinate system of the Earth. To understand that, a model of the Earth has to be developed, the *geodetic datum*, that needs to fit the real shape of the Earth as accurately as possible.

**The shape of the Earth** measured in the field of *geodesy* is very complicated. In the Babylonian Empire ( $\approx$  2000-539 BC) the theory of the Earth being a flat disc surrounded by an infinite body of water evolved<sup>8</sup>. The Greek scientists Pythagoras and Aristotle (340 BC) rejected this theory and proved the Earth to be a three-dimensional spherical object. It took almost 2000 years until Sir Isaac Newton (1687) reasoned that due to the centrifugal forces of the rotating Earth the shape has to be flattened at the poles and is therefore better described as an *ellipsoid* with two radii: the polar radius ( $r_p$ ) and the slightly larger equatorial radius ( $r_e$ ) [Bol08, pp. 69-77]. However, the model disregards that the surface of the Earth is not flat but consists of deep oceanic trenches and high mountains. Therefore the gravitational field of the Earth is not homogeneous either: the actual *mean sea level*, the reference surface for the height of objects varies from 106 meter below to 85 meter above the uniform sea level of the ellipsoid model. These discoveries in the 20<sup>th</sup> century led to the complex model of a *geoid* (see figure 2.4). The latest and most accurate measurements for the shape of the Earth are the result of the GOCE satellite launched in March 2009 [Uot,Fra].

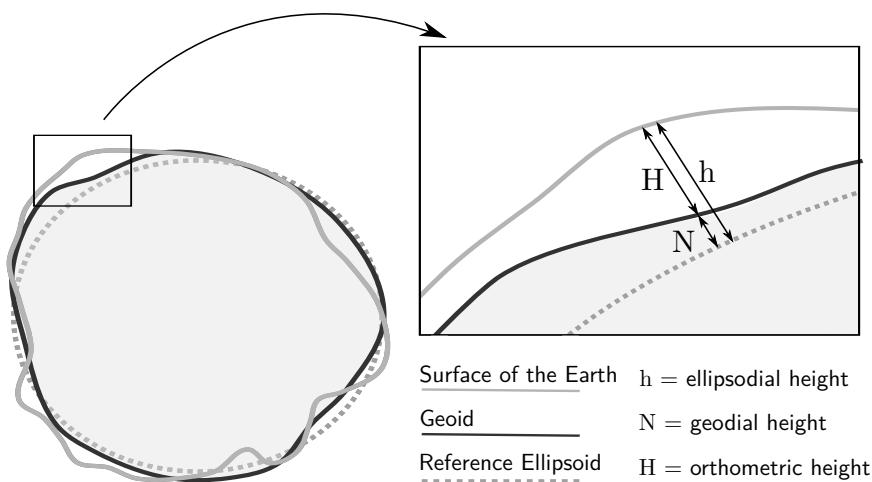


Figure 2.4: The geoid model, differences are exaggerated, [Bol08, Fig. 3-6, p. 75]

<sup>8</sup> A theory still valid in some southern parts of the United States of America

**Geographic coordinate system** The basis for the geospatial data model is the reference ellipsoid. It is represented in a three-dimensional *spherical coordinate system*. The *North* and the *South Pole* are defined as the two surface points closest to the Earth's center opposite to each other. The *Equator* is the line equidistant to the two poles and dividing the world in a *Northern* and *Southern Hemisphere*. Additionally, the *Prime Meridian* is defined as the line perpendicular to the Equator, running from the North to the South Pole. Since there are infinitely many lines like this, its definition is arbitrary, but by convention, the line running through Greenwich is used. Based on these two lines, each point in the spherical coordinate system can be unambiguously defined by [Bol08, pp. 26-28]:

1. The rotation angle along the Equator, defining its longitude:  $\gamma = [-180^\circ \dots + 180^\circ]$
2. The rotation angle along the Prime Meridian, defining its latitude:  $\phi = [-90^\circ \dots + 90^\circ]$
3. The distance to the origin:  $r \in \mathbb{N}_0$

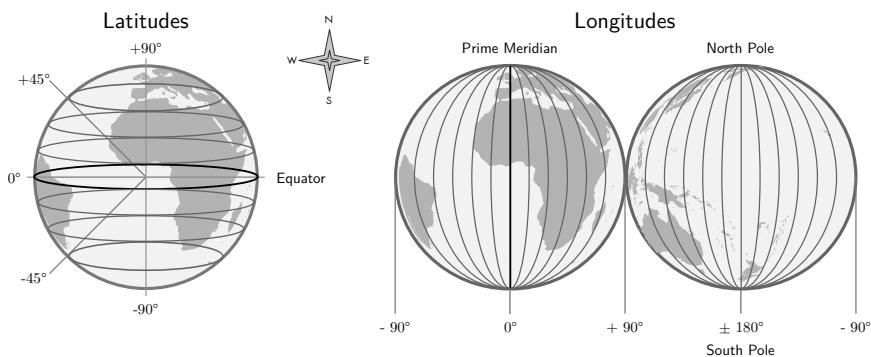


Figure 2.5: Geographic coordinates using latitude and longitude

Lines of constant latitude are running horizontally and are called *parallels*, lines of constant longitude in vertical direction are *meridians*. All parallels are circles with their center on the axis between the poles. No two parallels intersect. The longest parallel is the Equator ( $0^\circ$  latitude). All meridians have the same length. Geographic coordinates are usually recorded either in degree-minutes-second (DMS, e.g.  $50^\circ 58' 22''$ ) or in decimal degree (DD, e.g.  $50.973$ ) notation [Bol08, pp. 30, 79].

**The Geodetic Datum** is the discrete digital model of the continuous Earth that is described by the geoid. It consists of two parts: The approximation of the Earth's surface in a the Cartesian coordinate system with the origin in the Earth's center and a set of reference points used to accurately locate a point. Geodetic datums can be very accurate in one region of the world, i.e. the model fits the real geoid very well, but inaccurate in another region. This is the main reason why there are a lot of different geodetic datums used in the world. The same coordinates in two different geodetic datums define two different points on Earth. In order to be accurate is essential to know the geodetic datum of the coordinates [Bol08, p. 80]. The *World Geodetic System 1984 (WGS84)* is a model that found worldwide acceptance and is used in all major Web-based mapping services like *OpenStreetMap* and in the GPS unit of major mobile devices.

**Vector Model** The real world is infinite in detail, but storage in a computer is finite. In order to model continuous geographical phenomena in an information system, a relevant subset of them has been sampled to create discrete spatial data. It can be represented either in a raster or vector model. In the vector model each spatial object is expressed by three basic geometric primitives.

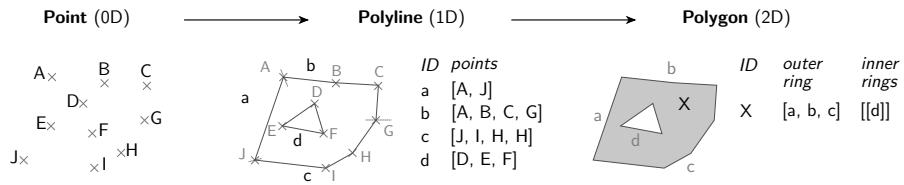


Figure 2.6: The basic geometric primitives point, polyline and polygon

0D A *point* is the fundamental object in vector geometry. It has no dimension and is only defined by its position, specified in geographic coordinates. A point is independent from all others.

1D A *polyline* is constructed by an ordered set of points with at least one start and one end point.

2D A *polygon* describes a closed surface and is constructed by one set of polylines forming a closed *outer ring*. Additionally, there can be multiple sets of polylines as *inner ring* representing holes in the polygon. If the polygon has no inner rings it is a *simple* polygon, otherwise it is *weakly simple* or *complex*, if it is even self-intersecting (see figure 2.7). *Polygons* represent multiple separate polygons belonging to one logical entity [Esr].

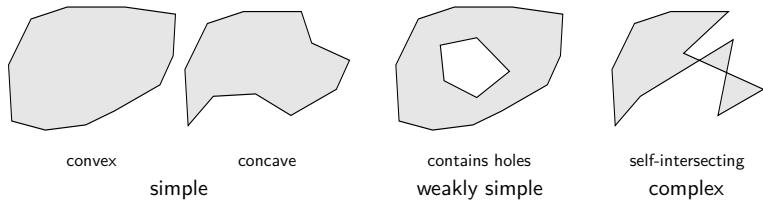


Figure 2.7: Types of polygons

There are mainly three boolean *set operations* that can be performed on two polygons: *Union*, *intersection* and *difference*. A *symmetric difference* is a combination of two differences. Their definition is visualized in figure 2.8. If multiple polygons are unified at once, it is a *cascaded union* [Bol08].

Common file types for geospatial vector data are the open file format GeoJSON (.geojson)<sup>9</sup> or ESRI Shapefiles (.shp)<sup>10</sup>.

<sup>9</sup> *GeoJSON*, IETF GeoJSON Working Group, URL: <http://geojson.org/>, last access: 30.10.2015

<sup>10</sup> *ESRI Shapefile Technical Description*, ESRI White Paper, July 1998, URL: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, last access: 30.10.2015

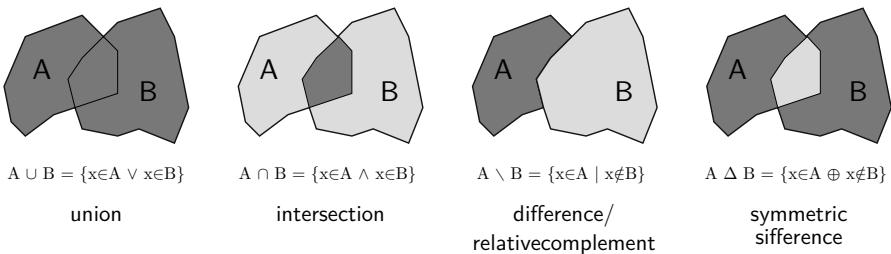


Figure 2.8: Boolean set operations on polygons

**Maps** The most common ways to present geographic space are two-dimensional *maps* and three-dimensional *globes*. The HGIS in this thesis will use a map to show how countries have developed over time. A map is structured according to the *layer* principle: Each layer is a transparent film showing one specific aspect, e.g. a physical layer for landmasses and water and a political layer for international borders and names of countries. The layers are interchangeable and can be switched on and off. A *legend* including the scale bar and north arrow should explain all symbols used on the map and give orientation. Interactive maps additionally have control options for panning and zooming, switching map layers on and off or changing the color scheme of the map [Bol08, pp. 159-166]. An example of such an interactive map service is *Leaflet.js* is “an open-source JavaScript library for mobile-friendly interactive maps”<sup>11</sup> that offers embedding a map on the client-side of a Web-based information system and additionally use own map layers, symbols and markers on the map.

The Earth is three-dimensional, but the map on the computer screen has only two dimensions. The model of the Earth has to be projected onto the map. But as previously discussed in subsection 2.2.2, the Earth is an inhomogeneous spherical object with a curved surface whereas the map is flat [Bol08, p.79]. That is why some features of the Earth will be distorted on the map: An *equivalent projection* preserves the area sizes of features on the map, whereas a *conformal projection* preserves angles and the shapes of objects. Every map projection that is area-preserving distorts shapes at the same time, and each shape-preserving map distorts areas to some degree. There is no perfect map projection that correctly presents all features of the real world [Geo]. A compromise between preserving areas and shapes is the *Robinson projection*. It is neither conformal, nor equivalent, but provides a reasonable trade-off between both properties.

<sup>11</sup> Leaflet - JavaScript library for interactive maps, URL: <http://leafletjs.com/>, last access: 02.11.2015

<sup>12</sup> Tissot indicatrix world map equirectangular proj, Eric Gaba (Wikimedia), 2008 URL: [https://commons.wikimedia.org/wiki/File:Tissot\\_indicatrix\\_world\\_map\\_equirectangular\\_proj.svg](https://commons.wikimedia.org/wiki/File:Tissot_indicatrix_world_map_equirectangular_proj.svg), last access: 28.10.2015

<sup>13</sup> Tissot indicatrix world map Mercator proj, Eric Gaba (Wikimedia), 2008 URL: [https://commons.wikimedia.org/wiki/File:Tissot\\_indicatrix\\_world\\_map\\_Mercator\\_proj.svg](https://commons.wikimedia.org/wiki/File:Tissot_indicatrix_world_map_Mercator_proj.svg), last access: 28.10.2015

<sup>14</sup> Tissot indicatrix world map Robinson, Eric Gaba (Wikimedia), 2008 URL: [https://commons.wikimedia.org/wiki/File:Tissot\\_indicatrix\\_world\\_map\\_Robinson\\_proj.svg](https://commons.wikimedia.org/wiki/File:Tissot_indicatrix_world_map_Robinson_proj.svg), last access: 28.10.2015

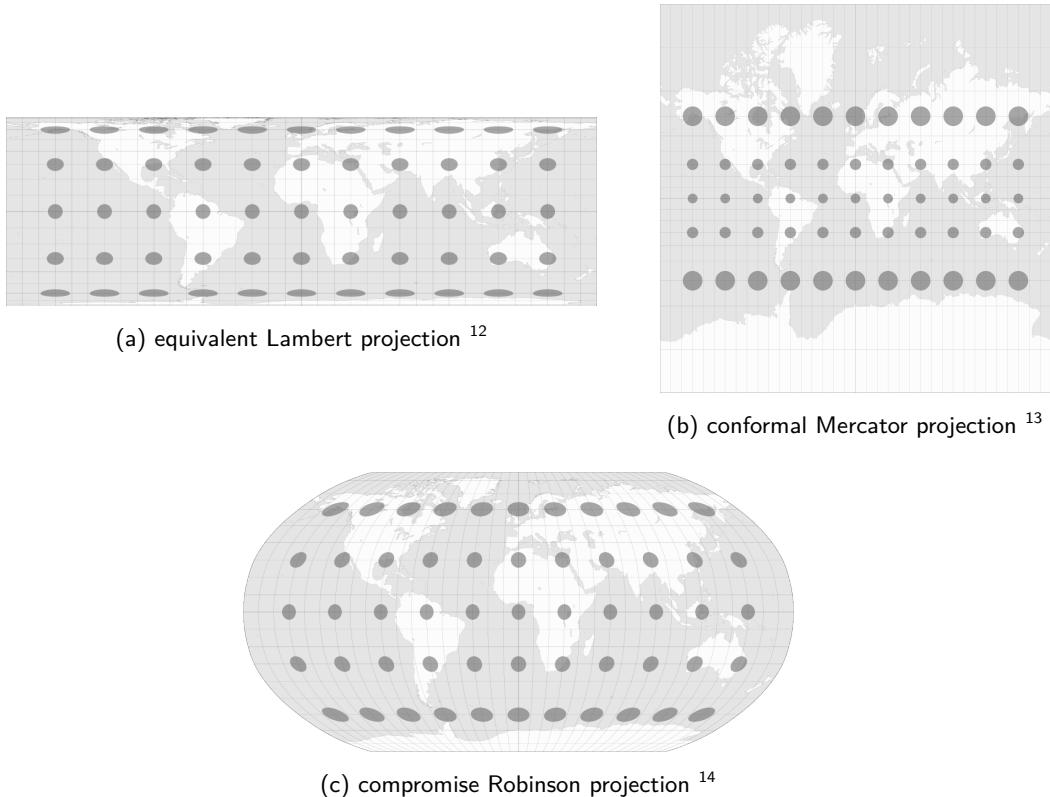


Figure 2.9: Comparison between equivalent, conformal and compromising map projections

### 2.2.3 Temporal Data

Time is an abstract concept that “can be perceived only by its effects” [Lan89, p. 27]. Many philosophers and scientists have developed models to deal with time. for this thesis, the model needs to be appropriate to represent time in an historical sense and in interplay with geographical space. A popular model is *Cartographic Time*, where time is seen as the “fourth cartographic dimension” [Lan89, p. 28]. Unlike space, time knows only one dimension, but relative to one point two directions: historically *forward* into the future and *backward* into the past.

Whereas space is represented by geo-objects, time may be represented by discrete *events* and continuous *processes*. Events can happen at a certain *time point* or like processes in a *time interval* or *time period*, defined by two time points [Sol14, chapter 2, pp. 47-49]. The Taxonomic Model of Time by [Fra98] classifies time also by its *nature* or *time order*: a consecutive development on the time axis, defined by start and end, defines *linear time*. On a contrary, *cyclic time* has no predefined order and events reoccur on a regular cyclic basis. The other two types, *branching time* and *multi-dimensional time*, are more complex and not relevant for this thesis.

The topological relationship between two time points  $t_1$  and  $t_2$  is straightforward. Since they are discrete elements and therefore isomorphic to the number space of integers, there are three different order relations:  $t_1 < t_2$ ,  $t_1 > t_2$  and  $t_1 = t_2$ .

**Timelines** In contrast to space, time does not have an intrinsic representation. However, the most common form to visualize cyclic time is on a cyclic display, e.g. a clock. Linear time is mostly visualized on a *timeline*. The purpose of it is to show events as time points or processes as time intervals in chronological order. A timeline additionally shows time markers on a certain date to support orientation. The position of an event on the timeline is described by its date using a reasonable sampling unit like century, year or day [Lan89, p. 32]. A timeline uses a certain time scale:

- On a *linear* timeline, the distance between any two time points is directly proportional to their actual temporal distance.
- A *logarithmic* timeline uses a logarithmic function to scale the depicted time. There is a reference point on the timeline, e.g. the timeline center. Just like for linear timelines, the further away a time point from the reference point, the further away it is positioned on the timeline. However, the distance between the time point and the reference point does not increase linearly, but logarithmically. That means, events that are further away do not appear as far. This time scale accounts for logarithmic human perception: events that happened 20 years ago do not seem to be twice as long ago as events happening 10 years ago [DISP08].
- A timeline can also have an *irregular* scale, e.g. to get the same absolute distance of events on the timeline. This is useful if the distribution of the events on the timeline are far from homogeneous.

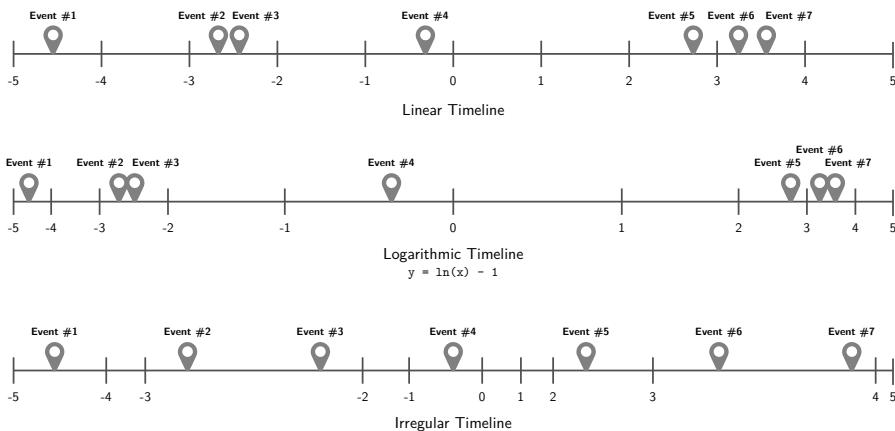


Figure 2.10: Comparison between a linear, a logarithmic and an irregular timeline

## 2.2.4 Existing Applications

“Today, operational temporal GIS does not exist”. This quote nicely summarizes the state of the art in this field. The main reasons are “the complexity of integrating space and time and the lack of standards” [Raz12, p. 5].

However, there are numerous HGIS projects for one specific research question. A large collection them can be found in [KH08] and [GG14]. One example related to the topic of this thesis is the *Great Britain Historical GIS Project (GBHGIS)*<sup>15</sup>. It maps statistical data on historical territorial units of the United Kingdom using *aerial interpolation* [Arc]. The purpose is to analyze for example net migration in the districts in the UK. The data is collected by the *British Ordnance Survey* that automatically detects spatial changes to the geography of the United Kingdom using aerial photography<sup>16</sup>. The *National Historical Geographic Information System (NHGIS)*<sup>17</sup> provides the digital boundaries of the United States of America and census data for each year since 1790. While the data in the system is extensive, the interface to analyze and use the data is very frustrating to use.

HGIS are not widely accepted in the humanities. One reason is the nature of the qualitative historical research: historic sources are subjective and biased, their content may be fuzzy and they are definitely incomplete. Therefore the knowledge that can be extracted from a source bears the integral problem of *uncertainty*. Information systems on the other hand have a logical architecture and try to be as precise and accurate as possible. Analysis is based on mathematical functions – an information system is quantitative in its entire nature [KH08, p. 2].

## 2.2.5 Data Sources

The HGIS developed in this thesis needs historical data about countries, their names and borders and historical events and their changes to countries. There are a lot of free and open sources for geographic data about the current countries, their names and borders. One of the most exhaustive collections of geographic data in public domain is hosted by Natural Earth<sup>18</sup>. There is physical data (e.g. coastlines and rivers) and cultural data (e.g. political borders, cities and roads). Data about historical countries and events are not as straightforward to acquire, because of the mostly qualitative nature of historical research (see section 2.2.1). The most exhaustive free and open source of historical

---

<sup>15</sup> *Great Britain Historical Geographical Information System (GBHGIS)*, Ian Gregory & Humphrey R. Southall, University of Portsmouth, since 1994, URL: <http://www.port.ac.uk/research/gbhgis/>, last access: 02.11.2015

<sup>16</sup> *British Ordnance Survey*, URL: <https://www.ordnancesurvey.co.uk/education-research/research/automatic-change-detection.html>, last access: 02.11.2015

<sup>17</sup> *Welcome to NHGIS*, Minnesota Population Center, University of Minnesota, since 2007, URL: , last access: 02.11.2015

<sup>18</sup> *Natural Earth*, URL: <http://www.naturalearthdata.com/downloads/>, last access: 30.10.2015

data is the *Wikipedia* and their article categories, e.g. armistices or treaties<sup>19</sup>. All sorts of historical events can be found. Some information is structured in information boxes, e.g. some historical treaties have a name, location, a signature and an effect date. Particularly interesting for this thesis are articles about historical countries<sup>20</sup>, because they contain the name and important meta information, e.g. their historical successors and predecessors.

Building an open-source Historical Geographic Information System on the basis of Wikipedia would be a huge project with significant impact on open education – however, it would also be a big challenge: Not all historical countries and events necessary to model the history of the world are available on Wikipedia. It is also inconsistent, because not all articles are structured, especially not to those of events that actually have an influence on a territorial change of a country, e.g. a border agreement. Retrieving, parsing and processing this information is challenging. Also accuracy and quality of information in the Wikipedia due to their open source nature has to be considered. Overall, using the Wikipedia as a data source for this thesis is not feasible, but is subject to further research.

**Historical maps** The most problematic data to acquire are historical borders of countries. There is no primary data source for that, so the most promising way is to extract a border from an historical map. They can also be found on Wikipedia, or in historical map collections, e.g. *OldMapsOnline*. The project is developed “out of a love of history and heritage of old maps” and stores about 400 000 historical maps<sup>21</sup>. There are five steps to retrieve the points of a border in geographic coordinates from an historical map. This process was developed in a preceding *HiBo* project (see figure 2.11).

1. **Digitization:** If the map is on paper, it has to be scanned in the best possible quality. The result is a raster graphic.
2. **Georeferencing:** The historical map has to fit as good possible on the reference map. This requires to manually define a set of reference points which are used to transform the map into the geographic coordinate system. This process is error-prone, especially if the projection of the historical map is not known and the map itself is not accurate [Kno02, pp. xvii]. The outcome is a raster graphic in which each pixel is assigned a geographic coordinate.
3. **Preprocessing:** The raster image has to be processed so that the desired border stands out and can be traced in the next step. This happens via greyscale conversion, thresholding or the

---

<sup>19</sup> Category:Treaties, Wikipedia, the free encyclopedia,

URL: <https://en.wikipedia.org/wiki/Category:Treaties>, last access: 13.05.2016

<sup>20</sup> List of former sovereign states, Wikipedia, the free encyclopedia, URL: [https://en.wikipedia.org/wiki/List\\_of\\_former\\_sovereign\\_states](https://en.wikipedia.org/wiki/List_of_former_sovereign_states), last access: 13.05.2016

<sup>21</sup> Old Maps Online, URL: <http://www.oldmapsonline.org/>, last access: 13.05.2016

<sup>21</sup> HiBo - semi-automatic extraction of borders from historical maps, Project of: B. Weber, N. K. Dankwa, K. Singh and T. Kashyappan, supervised by: Prof. Volker Rodehorst and Marcus Kossatz, Bauhaus-Universität Weimar, Oct 2014 - Feb 2015, URL: [https://bitbucket.org/bastian\\_weber/hibo](https://bitbucket.org/bastian_weber/hibo), last access: 29.10.2015

Canny Edge Detector. This results in a monochrome graphic in which the desired border must be uninterrupted and clearly be seen.

4. **Line detection:** By selecting a start and an end point of the border, the line gets traced automatically. This step vectorizes one particular feature, a borderline, from the raster graphic and produces a polyline in geographic coordinates.
  5. **Postprocessing:** In the last step, the polyline can be adapted: The line can be simplified to reduce unnatural artifacts and the position of border points can be manually edited. The final output of the whole process is a polyline whose points are expressed in the geographic coordinate system which can be used in the system as a border of an historic country.

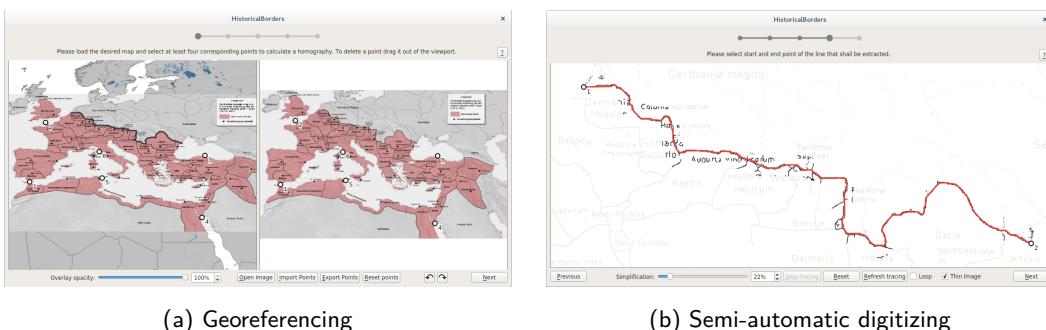


Figure 2.11: Semi-automatic extraction of a border from a map of the Roman Empire<sup>22</sup>

## 2.3 Spatio-Temporal Data Models

*“Geography differs from geometry because  
in geography, space is indivisibly coupled with time”*

– Don Parkes & Nigel Thrift (1980)

A data model abstracts a part of the real world, identifies the most essential elements and their relation to each other. Historical Geographic Information Systems use spatio-temporal data models to explain the historical development of geographic phenomena. Based on the theory of the *Triadic Framework*, three components are involved: space (3 dimensions), time (1 dimension) and attribute (1 dimension for each). All of these dimensions can change independently from each other [OS01, p. 53]. However, in order to trace spatial and attribute changes over time, the dimensions have to be related to each other. Spatio-temporal data models establish these relations.

Throughout the lifetime of a geo-object, it appears at some point, can undergo several changes and can disappear at some other time point. *Discrete changes* are based on the idea of a *state machine*: At any point in the lifetime, an object is in a certain state. It stays there until an event occurs that suddenly changes the object into a new state, e.g. the German Reunification in 1990 unified East and West Germany to present-day Germany. On a contrary, an object can gradually change according to a *continuous process*, e.g. the change of the coastlines due to the Sea Level Rise [Peu99].

In the previous 30 years many spatio-temporal data models were developed. The basis for all models is the concept of *Time Geography* by [Häg70]: There is an orthogonal relationship between time and space. At each time point an object is at exactly one location. The models can be classified by their organizing dimension: In *location-based* models time is an attribute of a geo-object. On a contrary, *event-based* approaches focus on events and processes that change geo-objects. This section introduces different spatio-temporal data models that are relevant for this thesis.

### 2.3.1 Snapshot Model

As it has already been introduced in 1.2, the *Snapshot Model* stores the full state of all geo-objects at certain time point  $t_i$  in a snapshot. It is one of the simplest, oldest and most frequently used spatio-temporal models despite its severe disadvantages [Lan88].

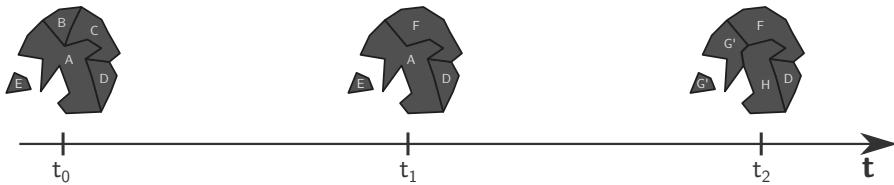


Figure 2.12: The Snapshot Model by [Lan88]

For all other time points  $t \neq t_i$  that are not covered by a snapshot, it is impossible to retrieve the state of the system, because the data model does not record any changes. This is an integral problem of the model and can not be solved which makes it unsuitable for the domain of this thesis. The original model is also redundant, because objects that have not changed from one snapshot to the next one are duplicated. However, there have been improvements made, e.g. by [Arm92].

### 2.3.2 Simple Time-Stamping

The simplest approach to trace the history of a geo-object is to assigning a period of existance to it. This happens by adding two attributes: at the *start date*  $t_{start}$  the object is created and at the *end date*  $t_{end}$  it ceases. If an object still exists its cessation date gets a special value, e.g. NOW [HW90].

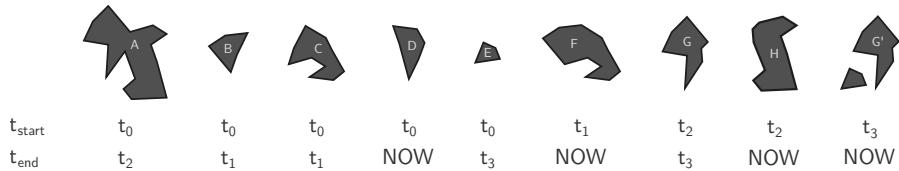


Figure 2.13: The Simple Time-Stamping method by [HW90]

The *Simple Time-Stamping* method is also location-based and tracks discrete changes of objects. Given full and integer information, the state of the system at each time point  $t_i$  can be retrieved: All geo-objects for which  $t_{start} \leq t_i < t_{end}$  are active, all others are inactive. However, this retrieval is cumbersome, because without efficient data structures every time the date changes, it has to be checked for each geo-object if its state has changed.

Another problem of the model is that it does not allow for tracing historical relations between geo-objects. As an example, at time point  $t_1$   $B$  and  $C$  cease and  $G$  starts. Visually,  $G$  is a successor of  $B$  and  $C$ , but this historical relationship can not be deducted directly from the model. This shortcoming can be resolved by adding a reference to the predecessor and the successor of the object.

This model alone is not suitable for the domain of this thesis, because it is impossible to say what exactly has happened at a certain time point. Given the example above, it is unclear if two objects unified to a new one ( $B + C \rightarrow G$ ) or if two are successors ( $B \rightarrow G$ ) and one just stops to exist ( $C \rightarrow \emptyset$ ). The model is also redundant: if a geo-object replaces another one ( $B \rightarrow G$ ), then  $t_{end}$  of  $B$  is the same as the  $t_{start}$  of  $G$ .

### 2.3.3 Event-Based Spatio-Temporal Data Model

A time-based approach addresses exactly those shortcomings: They explicitly represent events or processes in the data model and associate all objects that change according to them. One example of this approach is the *Event-Based Spatio-Temporal Data Model* (ESTDM) for geospatial raster data by [PD95]. At one defined time point  $t_b$ , a snapshot gets stored. This *base map* contains the current state of the map, i.e. the current value of each raster cell  $(x, y)$ . From that moment on, the system stores events that change the values of certain cells. Such an event has a time stamp ( $t$ ) and a list of components associated with it. A component represents a new value ( $v$ ) and knows which raster cells  $(x, y)$  change their value to  $v$ .

The method uses the following data structures: a header file contains information about the thematic domain, a pointer to the base map and to the first and last element of the event list. This doubly-linked list stores all events chronologically. Therefore, each event knows its preceding and succeeding event. If the time point of an event is reached, all its components are executed, i.e. the relevant

raster cells change their value. The system follows the next pointer to know which event is waiting to be executed next. Since a change is relative to the previous change, not to the base map, change tracking is efficient.

The concept of the ESTDM suits the problem domain really well: An historical event changes the geometry of certain objects suddenly. The model explicitly represents these discrete changes. However, it does not work for vector data. The authors have explicitly stated that “the design of such a [vector-based] model is not seen as a straightforward task”, because of the problem “how to maintain the integrity of spatial topology as it changes [...] The solution will require a more complex definition of components within individual events” [PD95, p. 21].

### 2.3.4 Three-Domain Model

An event-based STDM for vector geometry including lines and polygons has to answer the following questions: What uniquely identifies a geo-object? What kind of spatial, topological and attribute changes can happen to an object? Which of these maintain the identity and which create a new object? This problem is addressed in the *Three-Domain Model* by [Yua96a, Yua96b]. The model is based on abstract entities that represent a spatio-temporal object. It handles the three domains identity, space and time separately:

- The *semantic domain* holds an entity uniquely identifiable. An object in this domain corresponds to a human concept, e.g. a “country”.
- The *spatial domain* represents geospatial objects in vector format, e.g. a polygon describing the territory of a country.
- The *temporal domain* stores all temporal objects, e.g. time points of an historical event.

The model is not specific, but more a general abstract framework to handle space, time and identity. This makes the model very flexible, e.g. it can handle discrete and continuous changes, relative and absolute time, world and database time. One limitation of the model is that it only traces spatial attributes over time. In an alternative model by [CT95], the *thematic domain* is added to fully describe a spatio-temporal object and trace also aspatial attributes that can change over time, e.g. the name of a country. Since countries, their territories and attributes can change independently over time, the data model used in this thesis will be organized according to the Three Domain Model.

### 2.3.5 History Graph Model

Most of the data models introduced so far cover only static changes of geo-objects. [Ren96] identified three different types of temporal behaviour of changing objects:

- Dynamic objects that change continuously.
- Static objects that change according to events with duration (processes).
- Static objects that change according to sudden events.

Based on this observation a data model that can handle all three kinds of temporal behaviour was developed: The *History Graph Model*. It manages objects and events separately from each other. An object can only be in three different states:

1. An object is *static*, if it currently does not change. This is called an *object version*. The version has an interval associated to it representing the duration of the object version, until it changes the next time. If the object is dynamic and changes continuously, the duration is zero.
2. If an object is currently *changing*, it is in an *object transition*. The transition has an associated interval as well, whose duration is zero if it is a sudden change. Additionally, a transition links the relevant objects to each other creating a historical predecessor-successor-relationship.
3. An object that is currently not active, is *ceased* and not visible on the map.

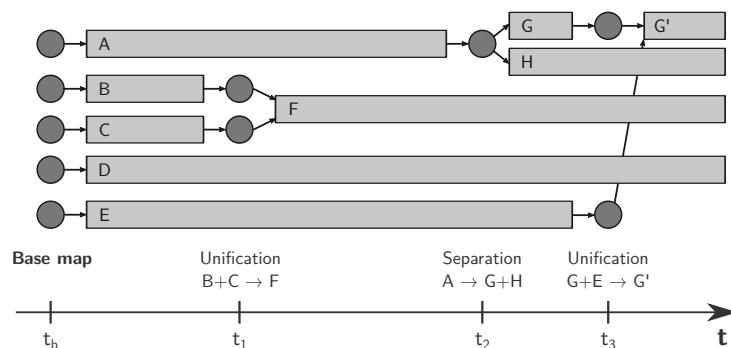


Figure 2.14: The History Graph model

The history of a geo-object is a chronologically ordered set of versions and transitions, that can be visualized in a graph like in figure 2.14. The model defines six basic types of temporal changes that can happen. They can be seen in figure 2.15:

- **Creation:** A new object is created.
- **Alteration:** A property of an object (e.g. geometry) changes.
- **Cessation:** An object is ceased.
- **Reincarnation:** An object that has previously been ceased is recreated.
- **Split/Deduction:** An object is divided into two or more new objects or one or more objects are deducted from an existing one.
- **Merge/Annexation:** Two or more objects are joined together to a new object or one or more objects are annexed to another object.

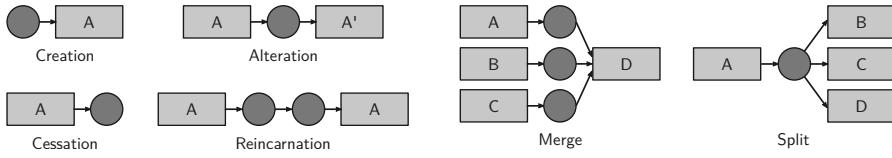


Figure 2.15: Types of changes in the History Graph model

The History Graph model can be seen as an extension to the ESTDM. It combines the advantages of event-based and location-based STDM and supports discrete and continuous changes. The main improvement is that the historical development of a geo-object can directly be derived from the model, because objects are linked to their predecessors and successors — the History Graph Model can tell a story. This is the reason why this model is particularly suitable for this thesis.

Other popular spatio-temporal data models that are not covered in this work, because they were not seen as relevant for the domain, include the *Space-Time Composite* model, the *Grid Model* and the *Amendment Vector Model*. Overviews about these and other spatio-temporal data models can be found in [Zha11], [PTKT04] and [Peu99].

### 2.3.6 Spatio-Temporal Databases

The data models that have been presented in this section have to be converted to a database model in order to implement them as the management component of the information system. A *Database Management System* (DBMS) is a software system for the administration of data, mainly storage and retrieval. There are mainly two types of DBMS: the oldest and most common ones are *Relational DBMS*. *Object-Oriented DBMS* were developed to adapt concepts of object-oriented programming into the database world. The combination of both approaches are *Object-Relational DBMS*.

**Relational DBMS** RDBMS are built upon the concept of *entities* and *relations*. An entity represents an object in the real world with a set of *attributes* and attribute values of a simple data type (e.g. string or int). Entities are represented in a table with one row for each *tuple* and one column for each attribute. An entity has one attribute that unambiguously identifies each tuple, the *primary key*, mostly a contiguous number. Entities can be related to each other in three different kind of relations:

- 1:1 Direct attributional relation, e.g. one country has one head of state and vice versa.
- 1:n One-to-many relation, e.g. one country can have many cities, but each city can belong to only one country.
- m:n Many-to-many relation, e.g. one country can have many rivers, but each river can also flow through multiple countries.

The first RDBM developed was *Oracle*, released in 1979 [Ora]. Since then, the concept has been established as the state-of-the-art for databases. An example for a RDBMS used for Web-based systems is *MySQL*, the “world’s most popular open source database”<sup>23</sup>.

**Object-Oriented DBMS** One problem with RDBMS is that attributes can only have simple data types. Developers using object-oriented programming need to map the objects used in the application to tuples in the relational database and vice versa. This process can be cumbersome. OODBMS solve this problem by adopting the concepts of object-oriented programming for database management purposes [Dar]. OODBMS support the following and many more object-oriented principles:

- *Classes* are structured representation of things in the real world of the same kind of properties, e.g. a country with a name and a territory. Classes in OODBMS relate to entities in RDBMS.
- An *object* is an instance of a class, one specific thing with defined properties, e.g. the country of Germany with its territory. This relates to a tuple in RDBMS.
  - The *attributes* of an object can not just be of a simple data type, but also instances of other classes, e.g. `country.territory` can be a `polypolygon` object.
  - Objects also have *methods* that can be called to do something with the object, e.g. `territory.getCenter()` returns the geometrical center of the polypolygon.
- The internal state of an object can not be accessed from the outside. Methods are the only way to interact with an object. This is called *encapsulation* and maintains control over what can be done to and with an object and prevents corruption.
- According to the concept of *inheritance*, classes can be hierarchically structured, whereas the attributes and the methods of a *base class* are inherited to its *derived class*. As an example, an *Area* has a name, a *territory* and the method `getArea()` associated to it. A *Country* can be derived from the *Area*, inheriting both attributes and the method. Additionally, it can get an attribute `head_of_state`, which is specific to *Country*, but not to *Area*. The class *Ocean* can just as well be derived from *Area* and it does not need a `head_of_state`.

**Object-Relational DBMS** combine the advantages of both worlds. Internally, it uses the established and efficient relational database for the data storage. The database model and the interaction with the data happens in an object-oriented way while supporting all of the concepts mentioned in the previous subsection 2.3.6. The most popular ORDBMS example for Web-based systems is *PostgreSQL*, “the world’s most advanced open source database”<sup>24</sup>.

---

<sup>23</sup> MySQL :: About MySQL, URL: <https://www.mysql.com/about/>, last access: 31.10.2015

<sup>24</sup> PostgreSQL:, The world’s most advanced open source database, URL: <http://www.postgresql.org/>, last access: 31.10.2015

**Spatio-Temporal Database Models** The question is: How to implement the spatio-temporal data models introduced in section 2.3 in a relational, object-oriented or object-relational database management system introduced in this section? While the implementation depends on the data model, there are common concepts and issues that have to be addressed. When storing time related data, it is important to distinguish between the time that was true in reality (*world time*) and the time it was stored in the database (*database time*). For this thesis, only world time is relevant.

Object-oriented concepts are more appropriate than relational ones, because of the complex nature of spatio-temporal data [PTKT04, section 3.9]. One of the first implementations was the *spatio-temporal object* combining geometrical and bitemporal properties in one object [WHM90]. A similar approach by [Raz12] is the *Spatio-Temporal Data Type* (STT): Time is not considered an attribute of space, but a separate class. They are aggregated in the SpatioTemporal class, using both spatial and bitemporal attributes. The model also provides spatio-temporal operators, e.g. STT.intersects returns true if two SpatioTemporal objects intersect in time and space, i.e. their geometries intersect and the time intervals in which they are active overlap. These operators are very helpful when analyzing spatio-temporal data or checking for data integrity.

Finally, a severe issue is *version management*. Given a database model that stores geo-objects that are created, updated and destroyed by events. Events can not only be appended to the end of the timeline, but also in between. This is called a *retrospective update*. It might create conflicting situations, e.g. if it ceases a geo-object that would be manipulated in a later event. The question is how to maintain data integrity on insertion, update and deletion from a spatio-temporal database? This issue has to be addressed using formal logic for temporal reasoning [Peu99, section 6].

## 2.4 HistoGlobe

*Imagine a globe that you can rotate and see from all sides.*

*Imagine a timeline which you can scroll to any point in time.*

*Imagine you can see how the world changes when you move the timeline:*

*country borders shift, troops move on the planet or kingdoms make way for democracies.*

*This is HistoGlobe, a revolution of teaching and learning of history!*

HistoGlobe is a Web-based Historical Geographic Information System that aims to visualize the history of the Earth on a globe with a timeline. It shows historical events and geopolitical changes. The goal is to provide a freely accessible open-source tool for a

- better *understanding* of history for learners,
- more appropriate *presentation* of history for teachers and
- interesting *exploration* of history for scholars.

The HistoGlobe project was founded in Weimar in 2010 and has since then evolved from a single student project via a tech-startup with up to 15 contributors to an academic project for the sake of research for geography, history and computer science. The current prototype of HistoGlobe was developed for students in school to understand German, European and World history since World War I with a focus on the time of the Cold War. The prototype can be seen at <http://histoglobe.com>.

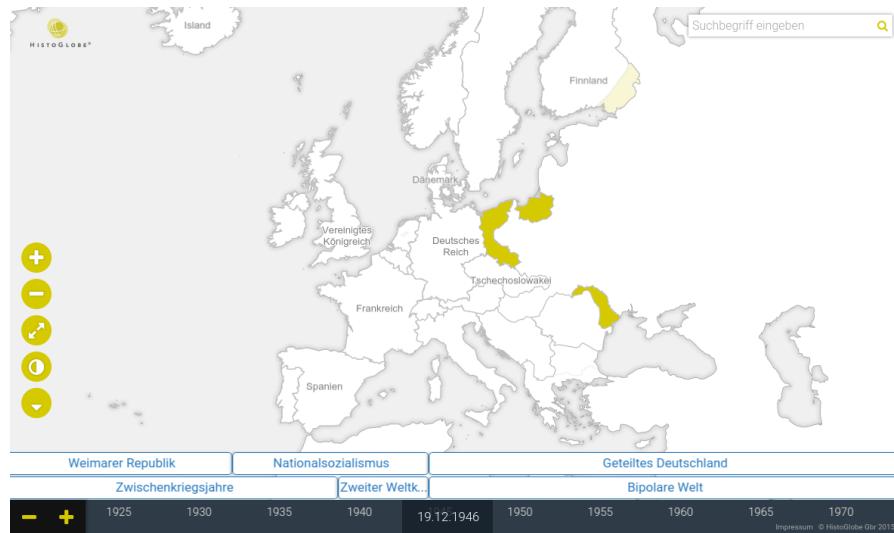


Figure 2.16: The HistoGlobe School project (status: April 2015)

Figure 2.16 shows the user interface of HistoGlobe: a 2D map, a timeline, control buttons for the map and the timeline, a topic bar extending the timeline showing historical periods in German and World history and a search bar for finding historical events. HistoGlobe is built upon a module system for interface components: The map can be exchanged with a globe, control buttons can be switched on and off or additional visualization approaches, e.g. for the history graph introduced in section 2.3.5, can be added. The idea is that the components are independent from each other, e.g. exchanging the map with a globe has no effect on the timeline or the control buttons. The main problem of HistoGlobe is the data: there is very little data in the system and editing historical information about countries and events is very complicated due to a complex domain and a lack of a proper backend.

This chapter introduced the problem domain of Historical Geographic Information Systems for the history of countries. Also existing spatio-temporal data models their advantages and disadvantages have been presented. The purpose of this thesis is to develop a model for managing the historical developments of countries in time and space. This model will combine some of these approaches and also consider complex cases due to the uncertain nature of historical research. It will be implemented as the foundation of HistoGlobe. Additionally, an editor for historical data is to be developed as a new HistoGlobe module to enable users to edit the course of history directly on the map. The next chapter will present the development process for the data model and the user interface.

# Chapter 3

## Development

This thesis aims for a comprehensive Historical Geographic Information System. The previous chapter has shown that this is a very complicated undertaking, because both the reality and the human using the system are complex. For such complex applications the methodologies of *Human Centered Design* are promising to create an interface that humans can easily understand.

The development process is iterative and divided in several phases. The outcome of each phase is a prototype of the interface that gets closer to the desired solution by increasing the fidelity of the prototype. A phase starts with an initial set of requirements. In multiple iterations, a prototype is developed that solves the problem. Each iteration has four steps: The requirements for the system are analyzed in the *planning* step. Afterwards, they translated into an abstract *design* which is realized in a specific *implementation* of the prototype. Finally, this prototype is tested with humans to find out how well it works. Based on the results of this *testing* step, the requirements are updated and the next iteration starts. This is repeated until a version of the interface is created that sufficiently solves the problem. Then the fidelity is increased, starting the next development phase. The five phases in this thesis are shown in figure 3.1.

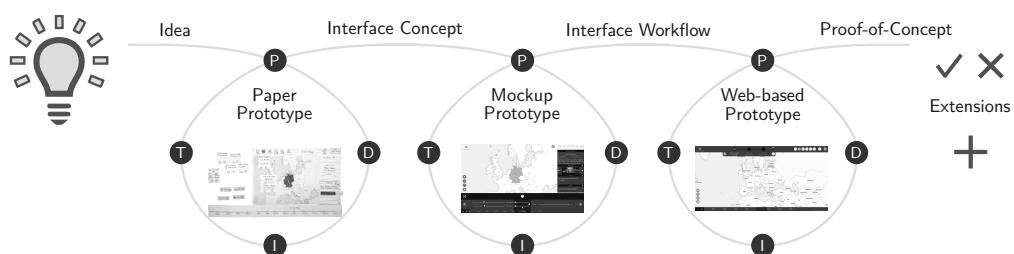


Figure 3.1: Human Centered Design process with five project phases

1. **Idea:** The initial idea how to edit and visualize the history of countries.
2. **Paper prototype:** The concept of the interface realized and tested on paper.
3. **Mockup prototype:** The concrete workflow developed in a slide-based presentation.
4. **Web-based prototype:** The final version developed in a Web application.
5. **Extensions:** Design approaches to account for the uncertain nature of history.

This chapter covers the first four phases of this design process, focusing on the results of the Web-based prototype and its underlying data model. Everything in this chapter is based on the assumption of full certainty about the data: For each time point in history there is a clear and undisputed state of the world. The data model and the application presented in this chapter is open to extension to tackle the actual problem of uncertainty in history. This is the topic of the next chapter 4.

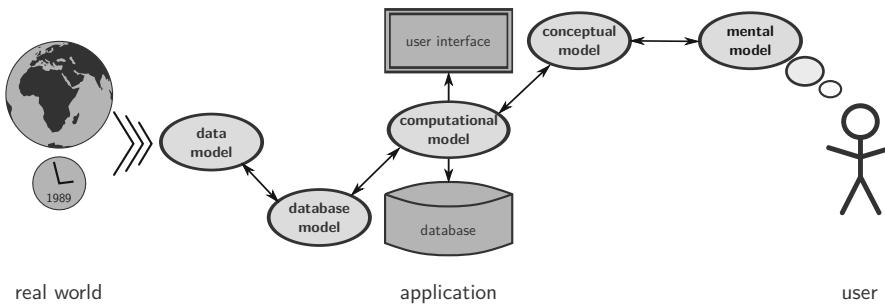


Figure 3.2: Relevant models for an information system

There are several models involved in the development of the software. The *data model* is an abstraction and simplification of the real world. The *Hivent Model* developed in this thesis is explained in the first section 3.1 of this chapter. It is followed by section 3.2 with methods to *edit* the spatio-temporal data in the system. In interactive computer systems, the *mental model* is the representation in the mind of the human about how the interface should work – the *conceptual model* describes the way the interface actually works. The goal of Human Centered Design is to match the conceptual model to the mental model. Section 3.3 outlines the gradual design process to reach this goal. In the application, the data model is implemented in the *database model*. The task for the *computational model* is to translate between the database model and the conceptual model. The implementation of HistoGlobe including the latter three models is presented in the last section 3.4 of this chapter.

## 3.1 Hivent Model

This section proposes the spatio-temporal *Hivent model* to represent countries and their history in time and space. In section 2.3, different spatio-temporal data models were introduced. The *Snapshot Model* is unsuitable for the problem space. *Simple Time-Stamping* is helpful to link countries to their history, but it does not explicitly model historical changes, which is desireable. For that purpose, the idea of the *Event-Based Spatio-Temporal Data Model* was developed, but since it only works for raster data, it does not fit the problem space. The *History Graph Model* fills this gap and additionally introduces temporal changes and their influences on geographic entities directly in the model. Finally, the *Three-Domain Model* presents a helpful concept to separate the spatial, temporal and thematic dimension of a spatio-temporal object.

The Hivent Model is constructed from components of some of these models: It is event-based and supports vector data. It is organized in four domains and allows to visualize data on a graph. The first section 3.1.1 introduces the main elements of the Hivent model. Afterwards, the preconditions are defined in section 3.1.2. One major contribution of this thesis is proposed in section 3.1.3: the set of five *Hivent Operations* that describe all possible changes of countries in time and space. This section closes with the *HistoGraph* (section 3.1.4), an aspatial visualization of historical developments.

### 3.1.1 Elements

**Hivents** – an acronym for ***Historical event*** – represent historically significant happenings, e.g. a treaty, bill or declaration. An Hivent happens at one particular point in time and space and is therefore the main organizing element of the eponymic data model. The focus in this work is on events that influence the geopolitical situation on Earth.

**Areas** represent one identical current or historical country. They are an abstract entity on the map with a *name* and a *territory*. In the real world, a country has a common *short name*, e.g. “Germany” and a potentially longer *formal name*, e.g. “Federal Republic of Germany”. Both attributes are part of the Area model. The *territory* of the Area is described by a polypolygon, a set of weakly simple polygons to account for enclaves and exclaves. The polylines of a polygon consist of an ordered set of points that represent the borders of the country.

**Historical Changes** The idea of the Hivent Model is that Areas can change over time. This happens via *historical changes* that are part of exactly one Hivent. Throughout the lifetime of an Area, it is created at some point  $t_s$ , its territory and short name can change multiple times  $t_i : t_s < t_i$  and at some point  $t_e : t_s < \forall t_i < t_e$  it ceases. Each Area is **uniquely identified by its formal name**. That means as soon as the formal name of an Area changes, it is considered a “new” Area. Since all changes in this model are sudden, there are only two possible states an Area can be in: It is *active*, if at the current time point it is historically existing, otherwise it is *inactive*.

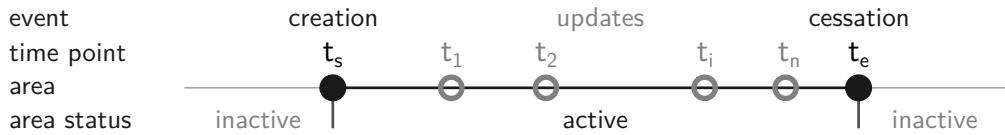


Figure 3.3: Three event types that change Areas, resulting in two different area states

Historical changes and Areas are mutually linked, i.e. an Area keeps references to the changes creating, updating and ceasing it and an historical change stores a set old Areas that cease, a set of new Areas that are created and a set of update Areas that are manipulated in the change.

**Four-Domain Model** Following the idea of the Three-Domain Model and its extension introduced in section 2.3.4, the Hivent Model is as well organized by four domains that are modelled independently from each other:

- The *semantic domain* holding the Area uniquely identifiable is represented by its formal name.
- The *spatial domain* is the territory of the Area that is visible on the map.
- The *thematic domain* is represented by the short name of the Area
- The *temporal domain* is the entirety of all historical changes associated to the Area.

The main difference between the spatial and the thematic thematic is that there is no relation between the names of two Areas: While it does not make sense, there could potentially be two Areas active at the same time with the same name. The update of the name of one Area is independent from any other Area. This is not true for the spatial domain: The territories of two Areas are highly related to each other. Geospatially each territory has at least one neighbor and two neighbors share the same border. An update of the territory of one Area results in the update of at least one other territory.

### 3.1.2 Preconditions

*In the beginning God created the heavens and the Earth  
Now the Earth was formless and empty [...]  
And God said, “Let there be light” — and there was light.*

– Genesis 1:1, The First Book of Moses, Old Testament

There are five axioms and two assumptions the Hivent Model is based on. The theoretical foundation is the model of the Earth and its curved surface that can be projected on a two-dimensional map using a map projection, as introduced in sections 2.2.2.

**Axiom 1** *The Earth's surface has an invariant area size, i.e. it does not change over time.*

**Axiom 2** *Each Area in the spatio-temporal system is located directly on the surface of the Earth.*

These axioms set the spatial foundation of the system: a constant dimension of the map and Areas covering the map. The basis of the temporal part of the system is content of the following axioms:

**Axiom 3** *The spatio-temporal system has an initial state at time point  $t_0$ . At this initial state, there exists exactly one Area, denoted by  $\Omega$  and referred to as the universe Area. It has no name and its territory covers the whole surface of the Earth.*

**Axiom 4** *At each time point  $t_i \geq t_0$  multiple historical changes can be introduced.*

**Axiom 5** *At each time point  $t_i \geq t_0$  each point on the surface of the Earth is covered by exactly one territory of exactly one Area.*

As it has been defined in section 3.1.1, an historical change can create, manipulate and cease Areas on the Earth's surface. According to axiom 5, each change introduced in the system must maintain the spatial integrity on the map, i.e. if an Area is created on the map, the Area claiming this territory before has to cease it.

Formally, it can be said that each change consists of three sets of Areas: the old Areas  $A$  that are ceased, the new Areas  $B$  that are created and the Areas  $C$ , whose properties are updated in the change. Each  $A_i \in A$  and  $B_i \in B$  has a territory  $A_i^T$  respectively  $B_i^T$ . Each  $C_i \in C$  has an old territory  $C_i^{OT}$  that is updated with the new territory  $C_i^{NT}$ . Each change introduced in the spatio-temporal system maintains the spatial integrity of axiom 5 like this:

$$\left( \bigcup_{i=1}^n A_i^T = \bigcup_{i=1}^n B_i^T \right) \wedge \left( \bigcup_{i=1}^n C_i^{OT} = \bigcup_{i=1}^n C_i^{NT} \right)$$

The first changes introduced in the system at time point  $t_0$  are the creation of all bodies of water, including the oceans and lakes, denoted as  $W$ . Each Area  $W_i \in W$  is created with their name and territory cut out of  $\Omega$ . The result is that after  $t_0$ , the map is divided into water ( $W$ ) and land ( $\Omega$ ). Land can at any point in time be either *claimed*, i.e. it is currently occupied by the territory of exactly one active Area, or on a contrary be *unclaimed*, i.e. belonging to  $\Omega$ .

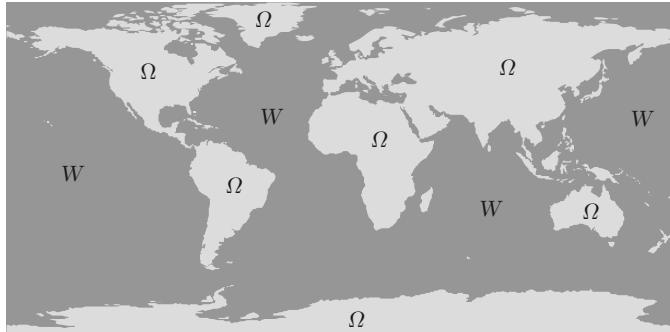


Figure 3.4: The initial state of the world map at time point  $t_0$

In the real world, the name of a country changes according to sudden events, e.g. a declaration or a governmental bill. The territory can change either because of a geographical processes, e.g. the sea level rise influencing the change of the coastline, or according to a historical event, e.g. a treaty. The Hivent model is based on two assumptions that simplify the model and keep the problem space clear:

**Assumption 6** *The territory of a country stops at the coastline.*

**Assumption 7** *The spatial configuration of water and the coastlines have not changed over time.*

Both assumptions are obviously wrong: The territory of a country stretches into international water (see section 2.1.3). Coastlines are constantly changing and so does the distribution of land and water on Earth. However, the assumptions allow the Hivent Model to focus only on discrete historical changes. It is subject to future work to extend the data model to account for long-term processes that change water and the coastlines. For now, the temporal behavior of an Area in the Hivent Model can be described as a *static object that changes according to sudden events*.

### 3.1.3 Hivent Operations

Respecting the preconditions, there are several different types of historical changes that transform a set of old Areas  $A$  to a set of new Areas  $B$  and update a set of Areas  $C$ . However, each possible historical change can be expressed with only one of five *Hivent Operations*. Four of them change the identity of at least one Area and therefore establish historical predecessor-successor-relationships. These relationships are always symmetric, i.e. if one old Area is replaced by one new Area, the old Area is the historical predecessor of the new Area and vice versa the new Area is the successor of the old Area. The last operation changes an aspatial property of an Area and is therefore identity-preserving.

**UNI – Unification** A set of old Areas unifies to one new Area. The old Areas cease, becoming the historical predecessors of the new Area. The territory of this new Area is the union of the territories of the old Areas. The new Area receives a new name.

In 1922, the Russian SFSR, the Transcaucasian SFSR, the Ukrainian SSR and the Byelorussian SSR unified and formed the Union of Soviet Socialist Republics (USSR).

**INC – Incorporation** One or more old Areas are incorporated into another Area that stays active. Its territory is enlarged by the union of the territories of the old Areas. The old Areas are historical predecessors of the Area that stays active.

In 1990, the territory of the German Democratic Republic (East Germany) became part of the Federal Republic of Germany (West Germany). Although this event is known as the *German Reunification*, it is historically an incorporation of East Germany into West Germany [Jan].

**SEP – Separation** As the inverse of unification, one old Area is separated into multiple new Areas. Each new Area gets a part of the territory of the old Area, receives a new name, and has the old Area as its only historical predecessor.

In 1993, the Czech and Slovak Federal Republic, commonly known as Czechoslovakia, dissolved into present-day Czech Republic and Slovak Republic, creating two new countries out of one old.

**SEC – Secession** As the inverse of incorporation, one or more new Areas are ceded from a previously existing Area that stays active. Each new Area gets a new name, receives the previously existing Area as the only historical predecessor and a part of its territory.

In 2008, the Republic of Kosovo declared independence from Serbia and has since then partially received international recognition. Serbia stays a country, keeping its name, but ceding a part of its territory to Kosovo.

**NCH – Name Change** An Area changes its short name but preserves its formal name and identity.

A recent change happened on 5. May 2016: The cabinet of Czech Republic approved that the country will now officially be called “Czechia”. However, the formal name stays “Czech Republic”, which preserves its identity.

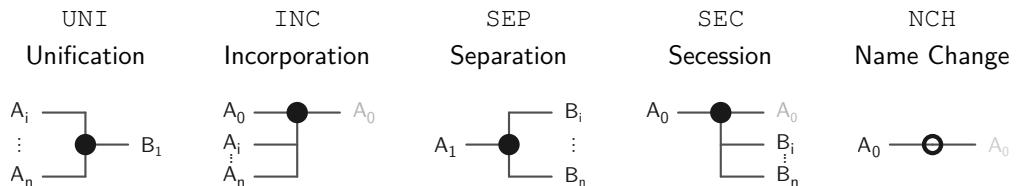


Table 3.1: The five Hivent Operations

### 3.1.4 HistoGraph

Based on the idea of the History Graph Model, introduced in section 2.14, the linguistically and conceptually related *HistoGraph* visualizes the temporal development of countries. The edges of the graph represent an Area, the nodes a Hivent Operation. The graph shows the predecessor-successor-relationships between Areas. This is easily possible, because as it has been explained in section 3.1.1, an Area keeps references to the historical changes creating, updating and ceasing it.

The two-dimensional HistoGraph has an horizontal orientation. The x-axis refers to one time point, the y-axis has no spatio-temporal relation. The graph uses the visualization approach of the five Hivent Operations in table 3.1, including the following symbols:

- Identity-changing Hivent Operation     UNI, INC, SEP, SEC
- Identity-preserving Hivent Operation     NCH
- A combination of both                        e.g. INC + NCH

Each uninterrupted horizontal line refers to exactly one Area. If an horizontal line leads straight through a circle, the identity of the Area is preserved in the operation. New Areas resulting from an identity-changing Hivent Operation emerge from the circle with a vertical line, indicating a sudden change with zero duration. From this line, the new Areas branch out right-angled. The HistoGraph is created from one particular reference Area. It visualizes historically related Areas in one direction: into the past, it recursively plots the predecessors on the graph, but not the predecessors successors. Into the future, the successors of the reference Area are plotted recursively, but not their predecessors.

The behavior of the HistoGraph is shown in figure 3.5 at the example of present-day Germany and its state history since the end of World War II. This history is driven by six historical events, which provide examples for all five Hivent Operations. They are listed in table 3.2.

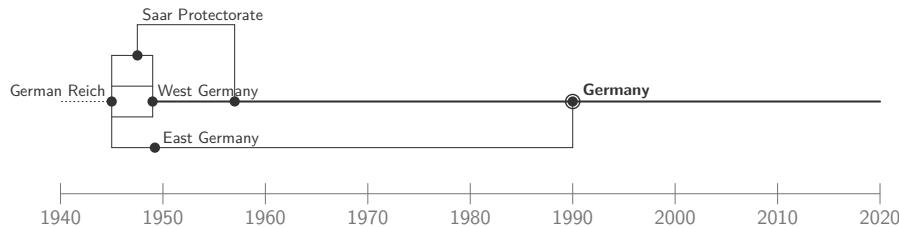


Figure 3.5: The concept of the HistoGraph at the example of the history of Germany since 1945

The example hosts a special case: in October 1949, East Germany was created from the Soviet Zone of Occupation. Both Areas have the same territory, but a different short and formal name. A NCH can not be performed, because the identity is not preserved: The German Democratic Republic is a new Area. However, the change can be described by a UNI of only one old Area (Soviet zone), creating a new Area (East Germany). This new Area occupies exactly the same territory as the old Area. Between both Areas a historical relationship is established.

Hivent date	Hivent description	Hivent Operations
05.06.1945	In the Berlin Declaration the total dissolution of the Third Reich is confirmed. It separates into multiple parts, returning the territories annexed by the German Reich in World War II. The rest is controlled by the British, French, American and Soviet occupation zone.	SEP
16.02.1946	The Saar Protectorate is entangled from the French Zone of Occupation Germany, creating an own country.	SEC
28.05.1949	The Federal Republic of Germany (West Germany) is created from the British, American and French Zone of Occupation.	UNI
07.10.1949	The German Democratic Republic (East Germany) is created from the Soviet Zone of Occupation.	UNI
01.01.1957	The Saar Treaty ("Little Reunification") joins the Saar Protectorate as the Bundesland Saarland in West Germany.	INC
03.10.1990	In the German Reunification, East Germany joins West Germany. The Federal Republic of Germany is now just called "Germany".	INC + NCH

Table 3.2: Historical events in German state history since 1945

The graph plots Germany first. Since it does not have any successors, the plot expands only one way, historically backwards: East Germany and the Saar Protectorate were incorporated into Germany, so they are plotted. They emerged from the four post-war occupation zones, visualized next. All of the four occupation zones themselves originated from the German Reich. However, the Reich dissolved into many more Areas, e.g. the Memel territory. They are not included in the graph, because they are not predecessors of any Area that is a recursive predecessor of present-day Germany.

Many problems of the graph visualization are apparent in this example: Circles may overlap if many operations happen in a short period of time – in this case between 1945 and 1949. The name "West Germany" collides with the vertical line indicating the incorporation of the Saar Protectorate, which should also be avoided. Additionally, the names of the Areas of the four post-war occupation zones can not be shown in the graph, because there is no space for them. One more important aspect can be seen in the creation of West Germany in 1949: A UNI operation unifies three old Areas to one new Area. This could be visualized symmetrically with a straight line from the midmost incoming Area line into the circle to the outgoing Area line of the new Area. However, this would give the wrong impression that this midmost Area has the same identity than the newly created Area. In general, the circle for UNI and SEP operations with an odd number of old respectively new Areas must be displaced off the center to emphasize that the identity has changed. All these issues are not in the scope of this thesis and subject to future work in the field of Information Visualization.

## 3.2 Editing Hivent Data

The previous section proposed the abstract Hivent Model, a set of Hivent Operations and the Histo-Graph visualization. However, one purpose of the HGIS developed in this thesis is to add, alter and delete historical changes. This section presents the tools and methods to edit spatio-temporal data about the history of Areas in the Hivent Model. Whereas the Hivent Operations are well-defined and specific, user studies have shown that they are not well understood by humans for edit purposes. This chapter introduces a different set of six *Edit Operations* in section 3.2.1. Afterwards, section 3.2.2 shows a *workflow* to perform an Edit Operation step by step. The Hivent Model needs to support editing historical changes in between other historical changes. The last section 3.2.3 explains the theoretical approach to *retrospective updates* of spatio-temporal data in the Hivent Model.

### 3.2.1 Edit Operations

The Hivent Operations are valuable, because they can describe all possible changes in the development of Areas in time and space. They are well understood by the system and form the basis for the Hivent Model. However, interviews with researchers in humanities at University of Virginia were conducted to understand their mental model about editing the history of countries on the map. It turned out that the Hivent Operations are not suitable for a human, because of their low-level nature. One example is that the operations do not provide a simple way to create a new Area on previously unclaimed land. Changing the formal name of an Area with a UNI operation is also not intuitive. Since a main goal of this thesis is to develop a user interface that humans can understand, a second set of six high-level *Edit Operations* is proposed. They are shown in table 3.3.

	CRE	Create	a new Area with a new name and territory on the map.
	MRG	Merge	two or more Areas to a new Area. The name has to be set manually, the territory is automatically unified.
	DIS	Dissolve	one Area into two or more new Areas, manually setting their new territory and name.
	CHB	Change Borders	between two neighboring Areas by defining the territory that changes sides.
	REN	Rename	an Area and set a new formal name, short name or both.
	CES	Cease	an Area by deleting it from the map, leaving unclaimed land.

Table 3.3: The six Edit Operations

**Error correction** Another use case for the interface is to correct wrong information on the map. For this purpose it is important to understand how correcting information in an event-based system works: Given an Area  $A$  at time point  $t_y$ . The name of  $A$  at this point should be  $Y$ , but happens to be  $X$ . That means the operation at  $t_x : t_x < t_y$  that created the name  $X$  for Area  $A$  is erroneous and has to be corrected. Correcting a state means correcting the operation that created this state.

### 3.2.2 Edit Workflow

The Edit Operations have proven to be understandable in several user studies. This section shows they can be internally expressed by a set of Hivent Operations. The creation of an Edit Operation happens in four steps of a simple workflow:

1. Select the Areas that will be changed in the operation.
2. For each new Area resulting from the operation, create a territory.
3. For each new Area create a name.
4. Add the Edit Operation to an Hivent to inherit the time point.

For each Edit Operation the requirements for the steps are different. Not all operations need all steps, since some data can be processed automatically. Table 3.4 presents an overview about the behavior of the Edit Operations in the first three steps. The last step is necessary for each operation alike.

	Select old Areas	Create new territories	Create new names
CRE	–	create a territory of the new country	create a name for the new country
MRG	select the countries to be merged	– <i>territories of selected countries are automatically unified</i>	create a name for the new country
DIS	select a country to be dissolved	create a territory for each new country	create a name for each new country
CHB	select two neighboring countries to change their border	create the new border between both countries <i>the territory for both countries will be created automatically</i>	–
REN	select a country to rename it	–	create a new name of the country
CES	select a country to cease it	–	–

Table 3.4: The requirements of each step for the Edit Operations

Depending on the input of the user in the steps for an Edit Operation, there are different possibilities to express it by a set of Hivent Operations. All possibilities are introduced in table 3.2.2. Hivent Operations are combined when they happen at the same time. In the example of the German Reunification, East Germany was incorporated into West Germany which at the same time changed its short name to “Germany” (INC + NCH).

EditOp. (case)	old Areas	update Areas	new Areas	expression by Hivent Operations	visualization
CRE (1)	$\forall P_i : \forall F_i$	$\forall P_i$	$B_1$	<p>The Area <math>B_1</math> is created with territory <math>T</math>. The part of <math>T</math> that is on previously unclaimed land (<math>T_\Omega</math>) is seceded as <math>B_1</math> from <math>\Omega</math>. If <math>T_\Omega</math> is empty, then <math>B_1</math> is initialized with an empty territory. The rest of <math>T</math> covers some Areas <math>P_i \in P</math> partially and some Areas <math>F_i \in F</math> fully.</p> <p><math>\forall P_i : P_i</math> is seceded and incorporated into <math>B_1</math>.</p> <p><math>\forall F_i : F_i</math> is completely incorporated into <math>B_1</math>.</p> <p><math>\text{SEC of } B_1 \text{ from } \Omega</math></p> <p><math>\text{SEC of } T_i \text{ from } P_i, \text{ INC of } T_i \text{ into } B_1</math></p> <p><math>\text{INC of } F_i \text{ into } B_1</math></p>	
MRG (1)	$\forall A_i$	-	$B_1$	<p>Multiple Areas <math>A_i \in A,  A  \geq 2</math> are unified to <math>B_1</math>. The new Area receives a name distinct from all the names of <math>A_i</math>.</p> <p><math>\text{UNI of } \forall A_i \text{ to } B_1</math></p>	
MRG (2)	$\forall A_i$	$A_0$	-	<p>Multiple Areas <math>A_i \in A,  A  \geq 2</math> are unified. The resulting Area reuses the short and formal name of one of the old Areas (<math>A_0</math>) and therefore preserves it. The remaining Areas <math>A_i</math> are incorporated into <math>A_0</math>.</p> <p><math>\text{INC of } \forall A_i \text{ into } A_0</math></p>	
MRG (3)	$\forall A_i$	$A_0$	-	<p>The same as the previous case, just that <math>A_0</math> receives a new short name and therefore an additional name change is required.</p> <p><math>\text{INC of } \forall A_i \text{ into } A_0</math></p> <p><math>\text{NCH of } A_0</math></p>	
DIS (1)	$A_1$	-	$\forall B_i$	<p>Multiple Areas <math>B_i</math> are separated from one initial Area <math>A_1</math>. Each <math>B_i</math> receives a part of the territory of <math>A_1</math> and a name. Each name is distinct from the name of <math>A_1</math>.</p> <p><math>\text{SEP of } A_1 \text{ into } \forall B_i</math></p>	
DIS (2)	-	$A_0$	$\forall B_i$	<p>Multiple Areas <math>B_i</math> are separated from one initial Area <math>A_0</math>. Each <math>B_i</math> receives a part of the territory of <math>A_0</math> and a name. One of the separated Areas has the same short and formal name as <math>A_0</math>, so it preserves its identity. The remaining new Areas secede from <math>A_0</math>.</p> <p><math>\text{SEC of } \forall B_i \text{ from } A_0</math></p>	

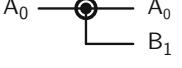
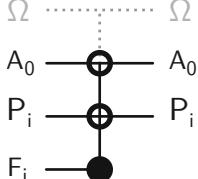
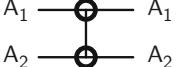
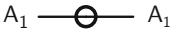
EditOp. (case)	old Areas	update Areas	new Areas	expression by Hivent Operations	visualization
DIS (3)	The same as the previous case, just that $A_0$ receives a new short name and therefore an additional name change is required.	$- A_0 \quad \forall B_i$	$\text{SEC of } \forall B_i \text{ from } A_0$ $\text{NCH of } A_0$		
CHB (1)	One existing Area $A_0$ is selected and its territory changes. Relative to the old territory some parts of the territory expands ( $T_e$ ) and some withdraws ( $T_w$ ). The part of $T_e$ that expands into unclaimed land ( $T_\Omega : T_\Omega \in T_e$ ) is seceded from $\Omega$ and incorporated into $A_0$ . The Areas $F_i$ fully covered by $T_e$ are incorporated into $A_0$ , the Areas $P_i$ partially covered by $T_e$ secede this territory $T_i \in T_e$ to $A_0$ . $T_w$ is be incorporated into $\Omega$ , resulting in unclaimed land.	$\forall F_i \quad A_0, \forall P_i \quad -$	$\text{SEC of } T_\Omega \text{ from } \Omega, \text{INC of } T_\Omega \text{ into } A_0$ $\text{SEC of } T_i \text{ from } P_i, \text{INC of } T_i \text{ into } B_1$ $\text{INC of } F_i \text{ into } B_1$ $\text{SEC of } T_w \text{ from } B_1, \text{INC of } T_w \text{ into } \Omega$		
CHB (2)	Two existing Areas $A_1$ and $A_2$ are selected and their common border changes. This results in a symmetrical change of territories, made up by two sets of territories: $T_2$ that previously belonged to $A_1$ and is now part of $A_2$ and $T_1$ for which the opposite is true. $T_2$ is seceded by $A_1$ and incorporated into $A_2$ , the opposite happens to $T_1$ .	$- A_1, A_2 \quad -$	$\text{SEC of } T_2 \text{ from } A_1, \text{INC of } T_2 \text{ into } A_2$ $\text{SEC of } T_1 \text{ from } A_2, \text{INC of } T_1 \text{ into } A_1$		
REN (1)	One Area $A_1$ is selected and both its short and formal name is changed. Therefore, a new Area $B_1$ is created as a direct successor of $A_1$ . This is a special case of a unification with only one Area.	$A_1 \quad - \quad B_1$	$\text{UNI of } A_1 \text{ to } B_1$		
REN (2)	One Area $A_1$ is selected and receives a new short name, but the formal name and therefore the identity is preserved. $A_1$ is updated.	$- A_1 \quad -$	$\text{NCH of } A_1$		
CES (1)	One Area $A_1$ is selected and ceases by incorporating into the universe.	$A_1 \quad - \quad -$	$\text{INC of } A_1 \text{ into } \Omega$		

Table 3.5: Translation from Edit Operations to Hivent Operations.

multiple Hivent Operations in one row happen exactly at the same time point, so they are combined

### 3.2.3 Retrospective Updates

A straightforward use case of the Hivent Model is to change the current state of the system with a new Hivent Operation into the future. Given the initial start point  $t_0$ , a current time point  $t_{now} > t_0$  and a set of consecutively added Hivent Operations at  $\forall t_i : t_0 \leq t_i < t_{now}$ . The accumulation of all changes make up the current state of the system at  $t_{now}$ . To change this current state, a new Hivent Operation can be inserted at  $t_{now}$  into the future. This state is valid until the next change is inserted. For historical research that use case alone is not sufficient, because the current state of the map at  $t_{now}$  is known to a large degree. The problem is to describe states and changes in the past. Therefore the system needs to support retrospectively inserting Hivent Operations in between existing operations and therefore updating the set of Hivent Operations in the system. Each Hivent Operation that is not entered to the end of the timeline must maintain the semantic, spatial and thematic integrity of the data, i.e. the changes to Areas, their territories and names must still work.

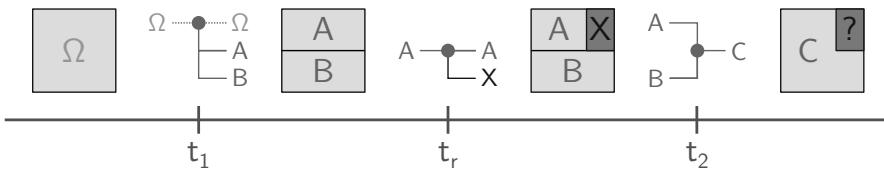


Figure 3.6: Example for a simple conflict due to a retrospective update

The simple example in figure 3.6 shows the problem: Given time point  $t_1$  with two Areas  $A$  and  $B$  and an UNI Hivent Operation at  $t_2$  unifying  $A$  and  $B$  to  $C$ . A SEC of a new Area  $X$  from  $A$  is inserted at  $t_r : t_1 < t_r < t_2$  in retrospective. The operation at  $t_2$  is not consistent anymore, because the old territory of  $A$  is not the same. It is not simple to say how the remaining territory ‘?’ should be treated.

**Conflicts** The way the Hivent Model works is comparable to a version control system like *Git*. There are different kinds of conflicts that can occur on retrospective updates. In the Hivent Model, they are classified regarding their resolvability:

- A) The conflict can be resolved *automatically* without the interference of the user.
- S) The conflict requires the user to choose between two alternatives (*semi-automatic* resolution).
- M) The conflict is complex and the user needs to resolve it *manually*.

The remaining part of this section examines all possible cases of conflicts and their resolvability. Each inserted Hivent Operation transforms a set of old Areas  $A = [A_i]$  to a set of new Areas  $B = [B_i]$  or updates an update Area  $A_0$  or does both. Each consecutive Hivent Operation that manipulates  $A_0, A_i \in A$  or  $B_i \in B$  has to be checked regarding three aspects of integrity:

1. semantic: Does  $A_0$  and  $\forall A_i \in A$  still exist? If not, can it easily be replaced by another Area?
2. spatial: Is the territory of  $A_0$  and  $\forall A_i \in A$  still the same? If not, can it easily be updated?
3. thematic: Is the name of  $A_0$  and  $\forall A_i \in A$  still the same? If not, can it easily be updated?

All cases can be simulated in the following simple scenario: Given the system with only two states: an initial state at  $t_1$  at which only three spatial entities are on the map ( $A_1, A_2, A_3$ ) and an Hivent Operation at  $t_2$  that manipulates some of these Areas with one of the five possible operations. This is called the original Hivent Operation ( $H_o$ ). Now, a retrospective update  $H_r$  is inserted in between the two states ( $t_r : t_1 < t_r < t_2$ ).  $H_r$  manipulates the same set of Areas with an Hivent Operation. The question is: What happens regarding the semantic, spatial and thematic integrity of  $H_o$ ? Is there a conflict and if so, how can it be resolved? There are 25 possible cases, because for both  $H_o$  and  $H_r$  there are five possible Hivent Operations.

**Retrospective Name Change** The first five cases are straightforward: Given NCH is inserted in retrospective ( $H_r$ ) to change the name of  $A_1$  from  $X$  to  $Y$ . This has no effect on the identity or territory of  $A_1$ . Therefore the system only needs to check for thematic integrity of  $H_o$ . If that operation is an INC or SEC, which both only change the territory of  $A_1$ , it is not conflicting. If  $H_o$  is a UNI or SEP, then there is a conflict:  $A_1$  is an old Area of the operation, but the name associated to  $A_1$  is still  $X$ . This is not consistent, because  $H_r$  just changed the name to  $Y$ . This conflict can be resolved automatically by updating the name in the old Area from  $X$  to  $Y$ . The same is true if  $H_o$  is a NCH operation:  $A_1$  is the update Area and the old name has to be updated from  $X$  to  $Y$ . To summarize what the system has to do if a NCH on  $A_1$  is inserted in retrospective: find the next UNI, SEP or NCH operation that manipulates  $A_1$  and update its old name.

**Retrospective Incorporation** An INC ceases a set of old Areas and changes the territory of one Area. In this scenario,  $H_r$  incorporates  $A_2$  into  $A_1$ . The question is what kind of conflicts can occur to the spatial integrity of  $H_o$ ? If  $H_o$  is a NCH, there is no conflict, because  $H_o$  changes the territory of  $A_1$  and NCH the name.

Figure 3.7 shows the conflicts that occur in the remaining four possibilities of  $H_o$ . In the case of an original UNI operation, it can still be performed with the same Areas, because  $H_r$  did not change the identity of  $A_1$ . However, the territory of  $A_1$  has been enlarged. This new territory has to be taken into account for  $H_o$ : To maintain spatial integrity, the system has to update the territory of incoming  $A_1$ . However, the territory of  $B_1$  has to be updated as well, because it is enlarged in the same way as  $A_1$ . This requires a recursive update into the future: the next Hivent Operation dealing with  $B_1$  needs to take into account that the territory has changed. This case can be treated as if  $H_o$  would be  $H_r$  with an INC operation. The system has to repeat this process until all conflicts are solved. The same is true if  $H_o$  is an INC operation: The system needs to update the old and the new territory of  $A_1$  in  $H_o$  and recursively update the territory of  $A_1$  into the future.

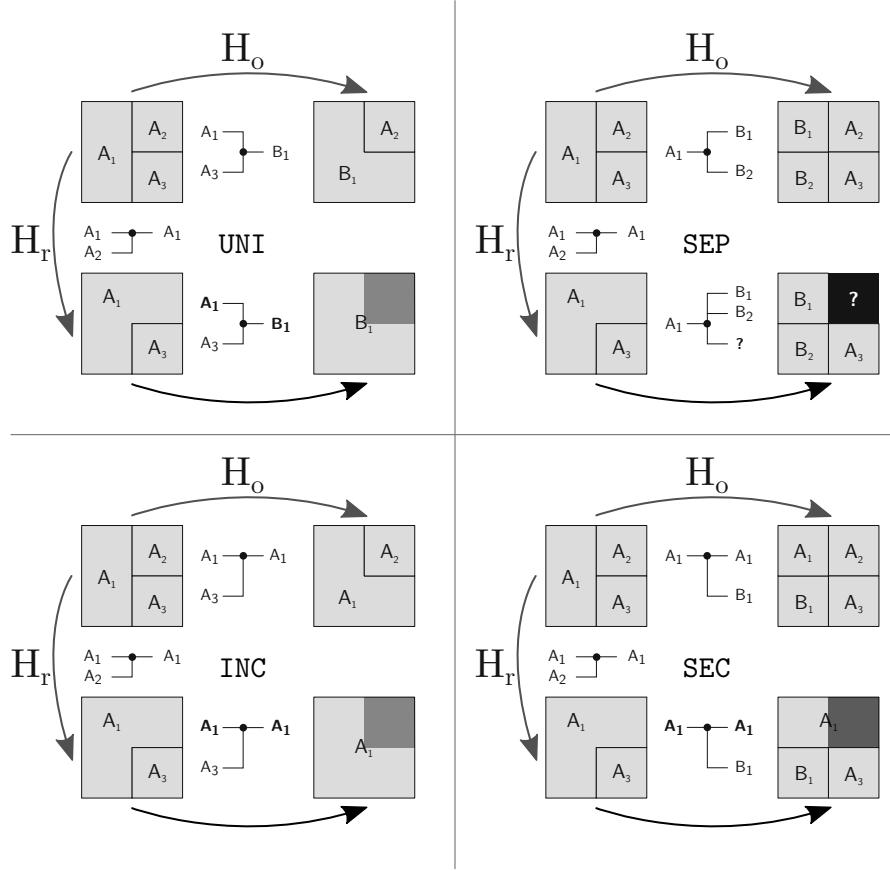


Figure 3.7: Conflicts after a retrospective incorporation

If  $H_o$  is a SEP operation, there is a major conflict: originally,  $A_1$  splitted into  $B_1$  and  $B_2$ . Due to  $H_r$ , the territory of  $A_1$  is larger.  $H_o$  can still separate  $B_1$  and  $B_2$  from  $A_1$  the same way as before, but it is unclear what should happen to the remaining territory of  $A_1$  that has just been enlarged. This conflict has to be resolved manually, because the system can not derive a decision from any existing information. The remaining part could become  $\Omega$ , it could be incorporated into  $B_1$  or  $B_2$  or stay  $A_1$ . However, the user has to decide it. In case of an original SEC, the situation is slightly different:  $H_1$  still exists like before, just with a larger territory.  $H_o$  can secede  $B_1$  like originally, but the system needs to update the old and new territory of  $A_1$  in this operation and recursively into the future.

The foregoing investigation relates only to  $A_1$  but not to  $A_2$  that has been incorporated into  $H_r$ . From the perspective of  $A_2$ , this change can be seen as a UNI, because its identity ceases and together with its territory it completely merges into  $A_1$ . Therefore this is treated like a retrospective unification examined later in this section.

**Retrospective Secession** Retrospective secessions are comparable to incorporations: The identity and the name of  $A_1$  does not change – but parts of its territory ceases to a new Area  $B_1$ . This section examines how the system has to treat  $B_1$  and the smaller territory of  $A_1$  in the original operation  $H_o$ . Exactly like for retrospective incorporation, there is no conflict if  $H_o$  is a NCH.

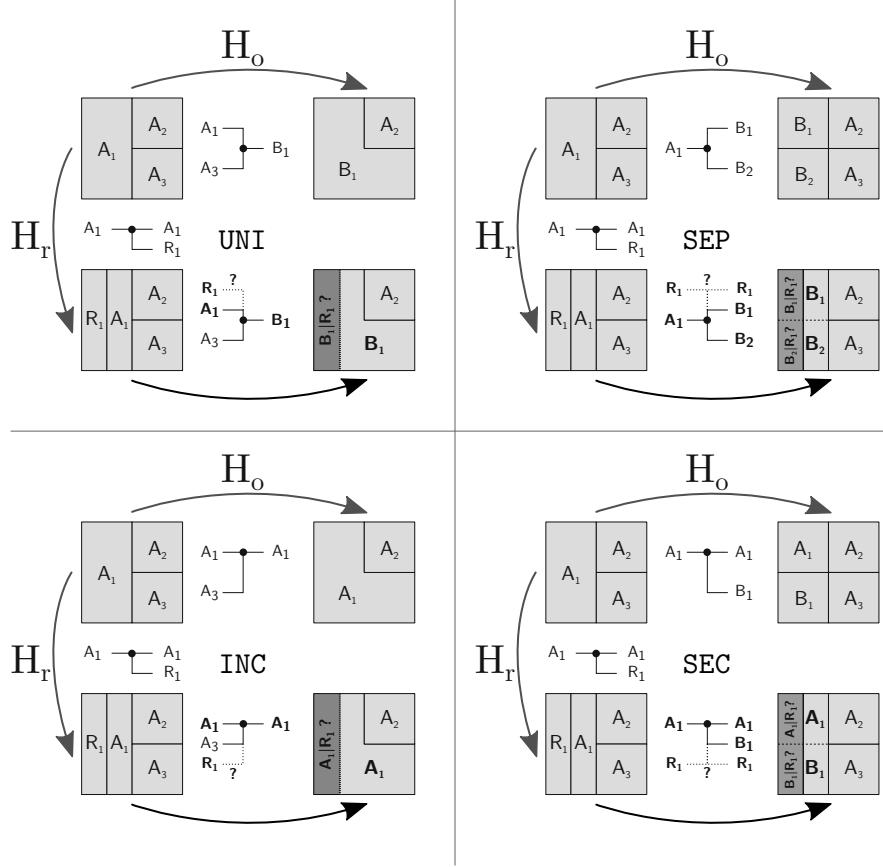


Figure 3.8: Conflicts after a retrospective secession

The other four cases are visualized in figure 3.8. For an original UNI, there is a conflict: Originally,  $A_1$  and  $A_3$  unified to  $B_1$ . Although  $H_r$  cedes parts of the territory of  $A_1$  to  $R_1$ ,  $H_o$  can still merge it with  $A_3$ . The question is what happens to the remaining territory of  $R_1$ ? There are two options:

1. *priority to  $H_r$* :  $R_1$  stays an own Area and is not affected by  $H_o$ .
2. *priority to  $H_o$* :  $H_o$  unifies  $R_1$  to  $B_1$  as well.

For the system it is hard to decide this, because it is unclear what the intention of the user is when creating  $R_1$  in  $H_r$ : Should the newly created Area be there for longer or is the territory of the united Area  $A_3$  more important? In order for the system to not behave unexpectedly, it will ask the user which choice he or she prefers. In case of the first choice, the territory of  $R_1$  has to be subtracted

from the territory of the new Area  $B_1$  in  $H_o$ . This update is again recursive, because the next Hivent Operation dealing with  $B_1$  needs to operate on the correct territory as well. In the second case,  $R_1$  simply has to be added to the old Areas of the UNI operation in  $H_o$ . No further recursive update is necessary. The same behavior is true if  $H_o$  is an INC. The user has to decide if he or she wants to incorporate  $R_1$  into  $A_1$  or keep it as a separate Area. In the latter case, the system needs to update the new territory of  $A_1$  in  $H_o$ .

For an original SEP, the situation is comparable: Originally,  $A_1$  splits into  $B_1$  and  $B_2$ , but after the retrospective secession of a part of  $A_1$  to  $R_1$ , the territory of  $R_1$  is conflicting with  $B_1$  and  $B_2$  in  $H_o$ . An important observation is that each part of  $R_1$  would be part of either  $B_1$  or  $B_2$ . There is no empty land, since both  $R_1$  and  $B_1 + B_2$  seceded from the same territory of  $A_1$ . Just like the other two cases above, the system will give the user for both conflicting territories the choice if either  $R_1$  should stay an Area or if  $B_1$  respectively  $B_2$  should incorporate  $R_1$  into their territory. In the first case, the territory of  $R_1$  has to be subtracted both from the incoming  $A_1$  and recursively from the outgoing territory  $B_1$  respectively  $B_2$ . The latter case needs at least one additional Hivent Operation: the part of  $R_1$  that shall be part of  $B_1$  respectively  $B_2$  needs to be seceded from  $R_1$  and the same time incorporated into  $A_1$  which then again in the same moment is separated into  $B_1$  and  $B_2$  ( $H_o = SEC+INC+SEP$ ). If whole  $R_1$  ceases in the operation, then it is incorporated into  $A_1$  completely, so it is only a INC+SEP at  $H_o$ . The case of an orginal SEC the system behaves in exactly the same way, just with an update of the territory of  $A_1$  and  $B_1$  instead of  $B_1$  and  $B_2$ .

To summarize, the main difference between retrospective incorporation and secession is that in the latter case, the user needs to choose between two predefined options. For incorporations, this is seen as unnecessary, because the conflicting territory of  $A_2$  has not been manipulated by  $H_o$ , so it can not be seen as a concious decision of the user to keep this Area.

**Retrospective Unification** If a UNI is inserted in retrospective, the semantic integrity of  $H_o$  is threatened, because in contrast to a INC all incoming Areas cease and unify to one completely new Area, in this example  $R_1$ . For each Area  $A_i \in A$  that was unified in  $H_r$ , the system needs to find the next Hivent Operation  $H_o$  manipulating  $A_i$  and update it accordingly. In this example,  $A_1$  is examined in place of each  $A_i \in A$ . If  $H_o$  is a NCH, there is a conflict: the name of  $A_1$  can obviously not be updated anymore, because  $A$  does not exist anymore. The only way to resolve this conflict is to automatically delete the NCH operation. All the other four cases behave regarding spatial integrity in exactly the same way as for a retrospective incorporation – with the only difference that the Area  $A_1$  is replaced by  $R_1$  as an incoming Area in the operation. In all four cases, the territory has to be updated in the same way as for retrospective incorporation and the same conflict occurs for the original SEP operation.

**Retrospective Separation** In contrast to the previous example, retrospective separations behave slightly different from secessions.  $H_o$  has to be checked both for spatial and semantic integrity, since

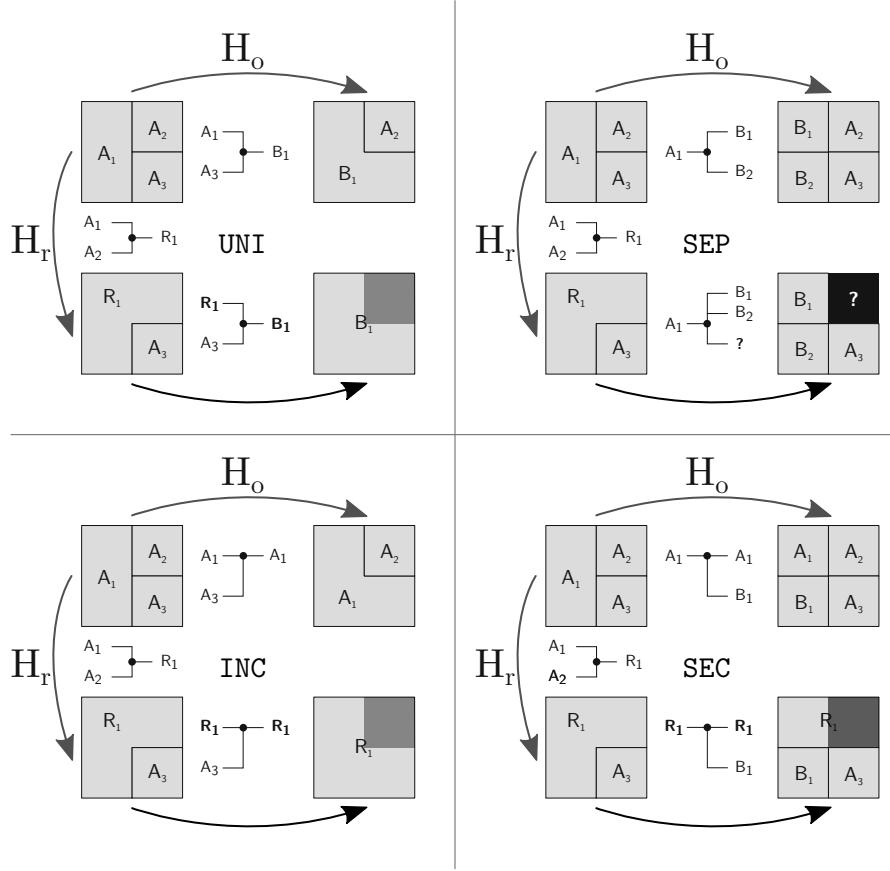


Figure 3.9: Conflicts after a retrospective unification

$A_1$  ceased in  $H_r$  to a set of new Areas  $R_i$ . In this scenario,  $R_1$  and  $R_2$  stay in place for  $\forall r_i \in R$ . If  $H_o$  is a NCH, the operation has to be automatically deleted, because  $A_1$  does not exist anymore. Figure 3.2.3 shows the conflicts that arise for the remaining four cases.

In case  $H_o$  is a UNI, the arising conflict can be solved semi-automatically: originally,  $B_1$  was created by unifying  $A_1$  and  $A_3$ .  $A_1$  just got separated into  $R_1$  and  $R_2$  by  $H_r$ , so the system can not know if  $R_1$ ,  $R_2$  or  $B_1$  is preferred at  $H_o$ . In any case, the system needs to remove  $A_1$  from the old Areas in the UNI operation. For each created Area in  $H_r$  – in this case only  $R_1$  and  $R_2$  – the system asks the user if he or she prefers  $R_i$  or  $B_1$ . If  $R_1$  is preferred, then its territory has to be recursively subtracted from the outgoing Area  $B_1$  in the UNI operation at  $H_o$ . Else  $R_i$  is added to the old Areas of  $H_o$ . If  $R_i$  is preferred every time, the remaining UNI operation only transforms  $A_3$  to  $B_1$ .

If  $H_o$  is an INC, the situation is more complex: The Area  $A_1$  in which  $A_3$  should be incorporated into does not exist anymore. Moreover, it is replaced by a set of new Areas  $R_i$  and it is not straightforward to see where  $A_3$  should be incorporated into – it is only clear that  $A_3$  will cease in this operation. Since

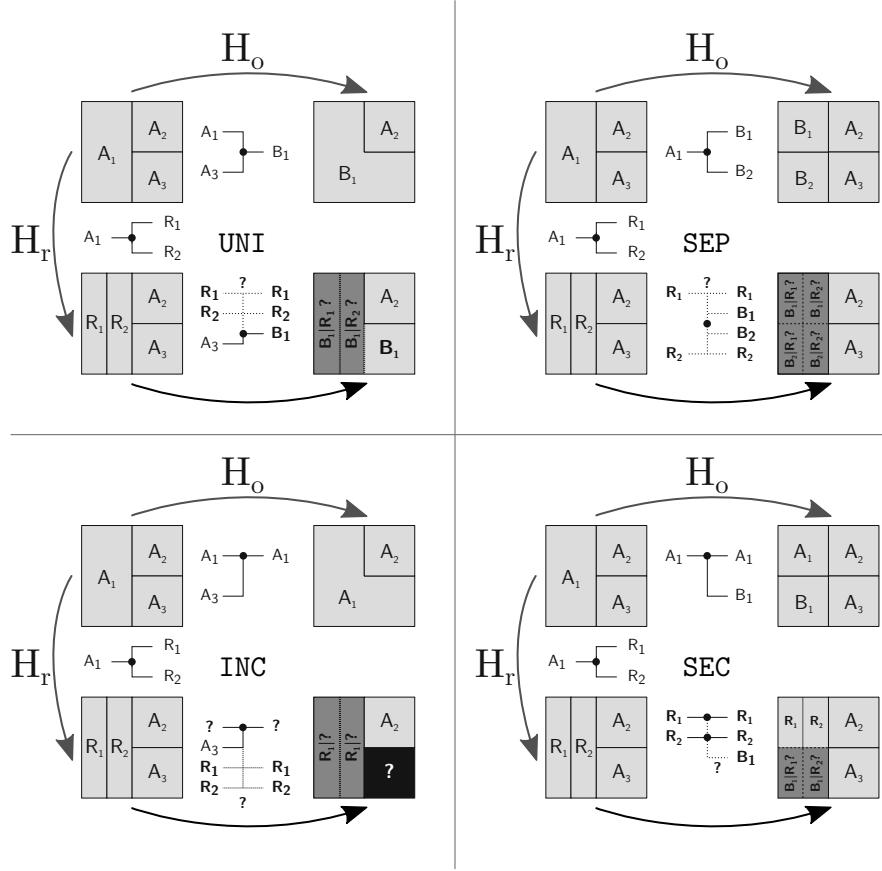


Figure 3.10: Conflicts after a retrospective separation

there is no information about what should be there instead, this conflict has to be solved manually. The situation is even more complicated, because on the one hand, in  $H_o$  the user intentionally incorporated  $A_3$  into  $A_1$  which means there is an obvious intention to keep  $A_1$ . On the other hand, he or she separated  $A_1$  in  $H_r$  to two new Areas  $R_1$  and  $R_2$ , so also they can be seen as intentionally created. To avoid an unexpected behavior of the system, the best approach is give him or her the choice to keep  $R_1$  and  $R_2$ , but in case of denial let the user resolve the whole conflict manually.

The case of an original SEP is likewise complex:  $A_1$  does not exist anymore to be separated – instead there are two sets of Areas that can be seen as reasonable to be the outcome of the operation: On the one hand the Areas  $B_i \in B$  created in  $H_o$  and on the other hand  $R_j \in R$  created in  $H_r$ . Since it is impossible for the system to know which Areas to prefer, the user should decide for each possible combination  $i \times j$  which Area it should be. If  $R_j$  is preferred, then its territory has to be recursively subtracted from each  $B_i$  that intersects with  $R_j$ . Else the territory of  $R_j$  intersecting with  $B_i$  has to secede from  $R_j$  and be incorporated into  $B_i$  with a combination of SEC+INC at  $H_o$ . The SEP operation itself is not necessary anymore, because there is not one simple old Area anymore.

The last case to examine is the influence of a retrospective SEP on an original SEC. Originally,  $\forall B_i \in B$  would secede from  $A_1$ , which has just been separated into  $\forall R_j \in R$  in  $H_r$ . The part of the territory of each  $R_j \in R$  not covered by any  $B_i \in B$  will certainly stay  $R_j$ , because there is no other reasonable alternative. The conflict is at the overlapping territory between  $B_i$  and  $R_j$ . Just like for the previous example, the user has to resolve these conflicts semi-automatically by choosing either  $B_i$  or  $R_j$  for each possible alternative. If  $B_i$  is preferred, it is seceded from the related  $R_j$  at  $H_o$ . If  $R_j$  is preferred, its territory has to be recursively subtracted from the related  $B_i$ .

		original Hivent Operation $H_o$				
		UNI	SEP	INC	SEC	NCH
retrospective operation $H_r$	UNI	A	<b>M</b>	A	A	A
	SEP	S(A A)	S(A A)	<b>M</b>	S(A A)	A
	INC	A	<b>M</b>	A	A	X
	SEC	S(A A)	S(A A)	S(A A)	S(A A)	X
	NCH	A	A	X	X	A

Table 3.6: All possible conflicts on retrospective updates regarding their resolvability

X = no conflict, A = automatic, S = semi-automatic, **M** = manual resolution

For semi-automatic resolution, the resolvability of the two options is stated like S(1<sup>st</sup>|2<sup>nd</sup>)

All possible cases and their resolvability are visualized in figure 3.6. It became clear in the extensive examination of the conflicting cases that inserting a retrospective update into a spatio-temporal system is not straightforward at all. Especially if a separation or secession is added somewhere not to the end of the timeline, in almost all cases the user has to decide between two alternatives. In three cases there is even a manual resolution necessary. A lot of cases also require recursive updates, meaning the retrospective update can lead to a potentially high number of semi-automatic or even manual resolutions by the user, which can potentially be frustrating. On top of that the examination completely disregarded the situation of combined cases, e.g. if the original operation is a combination of SEC+INC resulting from a user-defined border change (BCH) between two Areas. This might become even more complex. In summary, the Hivent Model needs to be checked for potential adaptions that might simplify retrospective updates and make the data handling less frustrating.

### 3.2.4 Backward Operations

From a relative time point  $t_i$  there are two historical directions: forward into the future with a predefined end at the current time point  $t_{now}$  and backward into the past until the predefined start point of the system  $t_0$ . Everything until now was focused on forward operations that change the current state of the system at the set time point  $t_i$  into the future, either until another operation changes it again or until  $t_{now}$ .

As it has been argued in the previous section 3.2.3 for retrospective updates, only forward operations to the end of the timeline are not sufficient for historical research. For an historian to edit a state in the past, a backward operation might be useful: A Hivent Operation is inserted at time point  $t_i$ , but into the past:  $t_0 < t_i < t_{now}$ . As an example: Given the initial state 10.06.2016 with present-day Germany created on 03.10.1990 on the map. The user wants to enter the German Reunification. The HGIS must support separating Germany into East and West, but indicating that this was the state *before* 1990 and the original state was *after* this date. This is complicated, because the conceptual, data and computational model have to adapt to this requirement.

The Hivent Operations themselves allow to be executed the other way, because each of them has an inverse operation: A UNI can be inverted with a SEP and a INC with a SEC operation. NCH can be inverted with itself by swapping the old and new name.

One problem is the conceptual model: The user interface has to provide a visual clue that inverting the direction of an operation is possible. Additionally, if an Hivent Operation is inserted backwards, another problems comes into play: all new Areas of the operation would now be active from  $t_i$  on backwards into the *past*. Each Area that is created in a backward operation has to be provided with another operation that ceases it in backward direction or creates it in forward direction, otherwise the Area would be active all the back to  $t_0$ . This is probably not desirable.

### 3.3 User Interface Design Process

The Hivent Model presented in the section 3.1 and the methods to edit the data in section 3.2 serve as the foundation of HistoGlobe. However, developing the system bottom-up from the data model to the interface might not lead to usable system. Human Centered Design promotes a top-down process from the user via the interface into the core of the application. This section illustrates the iterative design process for this thesis seen in figure 3.1

The two main use cases for HistoGlobe that are focused in this thesis are *understanding* the history of countries and *editing* it. The initial interviews confirmed that the combination of a map and a timeline are a very appropriate way to interactively visualize the history of countries. Thefore, the main interface concept of HistoGlobe introduced in section 2.4 does not need to be changed. It is extended by the *HistoGraph* introduced in section 3.1.4. This promising set of visualizations forms the *Browsing Mode* of HistoGlobe to understand the history of countries.

For editing purposes the idea of a second interface mode was developed: The *Edit Mode* is the main product of the iterative design process illustrated in this section. It is based on the Edit Operations, the workflow to edit the data and the concepts of retrospective updates and backward operations from section 3.2. The Edit Mode allows to intuitively edit Hivents, Areas and operations directly in HistoGlobe, without the need to write data into database tables.

### 3.3.1 Paper Prototype

The initial interviews it became clear that usability is the key to a successful interface that humanity scholars would actually use. In the second phase, the idea of the Edit Mode was developed in a paper prototype. It consisted of a map of Europe, a timeline and a set of buttons and dialogs for the Edit Mode. The prototypes were evaluated with three test subjects each that had to solve four tasks covering different use cases and operations:

1. 1300: Rename incorrectly spelled name of Switzerland on the map (*correction*)
2. 1990: Unite East and West Germany (*forward change*)
3. 1993: Separate the Soviet Union into Russia, Estonia, Latvia, etc. (*forward change*)
4. 1944: Change the border between Finland and the Soviet Union before 1944 (*backward change*)

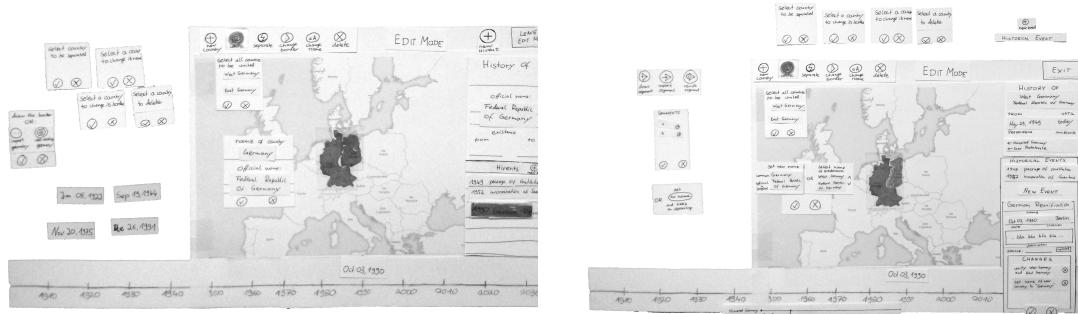


Figure 3.11: The two iterations of the paper prototype for the Edit Mode

Two paper prototype iterations were created. Each iteration took about three full work days including the creation of the prototype, the conduction of the study and the analysis of the results. Most parts of the interface concept were understood and all subjects were able to solve the first three tasks. However, paper prototyping was very helpful to identify the following flaws early in the design process:

1. The difference between Hivents, the history of a country and an historical change was unclear.
2. The backward change was not understood.
3. Correcting the name Switzerland by changing the event that created it in 1300 caused confusion.

The main finding of this step was that depending on the task, there is both an Hivent-based and an Area-based mental model of the task. This became apparent in the German Reunification Hivent: Some users started the unification operation first, and added West and East Germany afterwards – and some selected first West Germany, then initiated a unification operation and then added East Germany. From that finding arose that the interface has to support both an Hivent-based and an Area-based approach to introduce historical changes and correct information on the map.

### 3.3.2 Mockup Prototype

The main part of the design process was spent on the mockup prototypes. Their purpose is to rapidly develop an interface workflow that is understandable by the users. The prototypes were created in *LibreOffice Impress*, an open-source slide-based presentation tool. The interface is simulated on slides: the map is a background image, the timeline, the set of buttons and dialogs for the Edit Mode and HistoGraph are modelled with geometric elements: lines, circles and rectangles. Interactivity is simulated by linking a click on an element to a different slide that shows the effect of the operation. This allows to model sudden changes in the interface.

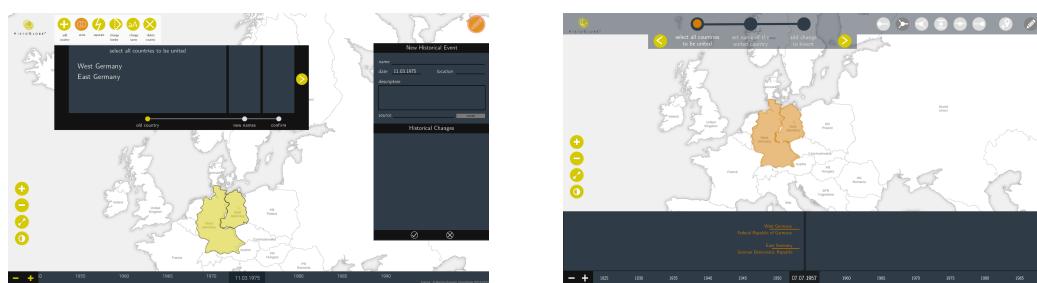


Figure 3.12: Two iteration stages of the mockup prototype for the Edit Mode

Each prototype iteration was tested with multiple subjects and similar tasks as for the paper prototype. From one test to the next one changes to the interfaces were made. A lot of design problems, e.g. position of buttons, font sizes or color schemes were solved, but also conceptual issues arose.

*"this was much easier than I thought"*

*"the interface is very clear  
and graphically pleasing"*

*"it's looking good"*

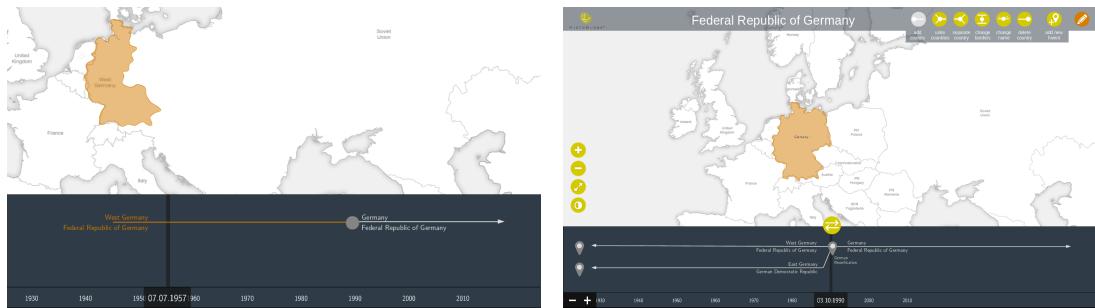
*"there is a training session needed"*

*"the logic makes sense,  
it is just very complex"*

*"a nice tutorial and a good  
documentation are necessary"*

– interesting quotes from the users of the mockup prototype

Especially the problem to initiate a retrospective update and a backward backward change proved to be very difficult. There was a design solutions developed for each problem: For retrospective updates, the interface needs to provide a visual clue where the next potential conflict arises: Figure 3.13a shows that West Germany can only be active until 1990, because then present-day Germany uses its territory. For backward operations a button that flipps an Edit Operation is introduced in figure 3.13b: the SEC operation introduced to secede East Germany from Germany will be flipped into an INC operation to incorporate East Germany into Germany. The two grey Hivent markers on the left side of the timeline indicate that West and East Germany need a creation event, otherwise they would be active backwards all the way to  $t_0$ , the initial state of the system.



(a) Retrospective updates: Visualizing the next conflict (b) Backward operation: flipping the Edit Operation

The prototype was very valuable for the development process. In a total of two weeks, an interface concept and workflow was designed the users understood. Its creation took longer than the paper prototype, but was much faster than implementing an interactive Web-based interface.

### 3.3.3 Web-based prototype

The main advantage of the design process so far is that it prevents major redesigns of the final Web-based prototype. After three months of implementation of the final system, the interface looks very similar to the last version of the mockup prototype. The main elements of the interface are the map, the timeline with the Now Marker the control buttons the map and the timeline and the Edit Mode.

However, not all desired features could be implemented: For the HistoGraph there were too many conceptual problems mentioned in section 3.1.4 that have to be solved first. Backward operations and retrospective updates are not supported as well, because of their complex nature behind the interface. The HistoGlobe version developed in this thesis only supports editing the history of countries with forward operations at the end of the timeline. The interaction and behavior is introduced in this section at the example of the fictional secession of Scotland from the United Kingdom in 2018.



Figure 3.14: Initial state of the Browsing Mode

The initial state of the user interface. Additional to the original elements, there is an edit button on the upper right corner. Clicking it enters the Edit Mode of the system.



Figure 3.15: Initial state of the Edit Mode

In the Edit Mode, a title bar and six buttons for the Edit Operations are revealed. Clicking a button starts the operation workflow introduced in section 3.2.2.



Figure 3.16: 1) Select old Areas

A *Workflow Window* guides the user step-by-step through the process of completing the Edit Operation. In the case of DIS, the user has to select the country to be dissolved by clicking it on the map. After the step is completed, clicking the next button in the workflow window proceeds to the next step. At each point in the workflow, clicking the back button reverts the previous action.



Figure 3.17: 2) Set a new territory

In step two, the user has to create a territory for each new Area. The *New Territory Tool* provides the functionality to create, manipulate and delete polygons directly on the map. The drawn polypolygon is intersected with the old territory to create the new Area. After at least one new territory is created successfully, the remaining old territory can be selected on the map to be used as the last territory. If the whole old territory is distributed among the new Areas, the workflow proceeds to the next step.

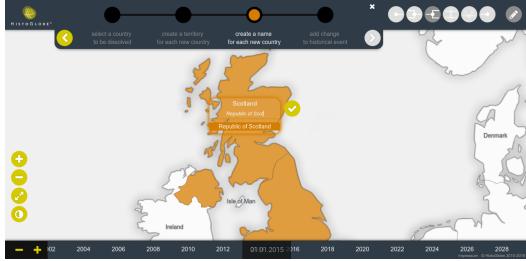


Figure 3.18: 3) Set a new name

In the next step, the user has to define a name for each Area that has just been created. The *New Name Tool* is a draggable box with two lines for the short and formal name. Via instant search, the user can select existing country names from the database to be put in. When clicking the confirm button, the short name is put directly on the map.



Figure 3.19: 4) Add the Operation to a Hivent

When all names are set, the Edit Operation is complete. In the last step of the workflow, it has to be added to an Hivent. The *New Hivent Box* offers two possibilities: search for an existing Hivent and add the operation to it or create a new Hivent.



Figure 3.20: 4) Create a new Hivent

The new Hivent created for that change is the “Scottish Independence” on 01.01.2018 with a description of the Hivent and possibly a location and a link to a wikipedia article. In the last line, the historical change “Secession of Scotland from United Kingdom” is noted. Clicking the confirm button finalizes the workflow.



Figure 3.21: The final state with Scotland

Clicking the edit button again leaves the Edit mode back to the Browsing Mode. Scotland and the United Kingdom are visible as separate Areas on the map after 2018. When moving the timeline before 2018, Scotland is still part of the UK.

## 3.4 Implementation

HistoGlobe is a Web-based Historical Geographic Information System. The data model and the conceptual model of the user interface were introduced in the first sections of this chapter. This section introduces the underlying database model, a specific implementation of the data model, and the computational model that translates between the conceptual model and the database model. The first part provides an overview about the architecture of the system in section 3.4.1.

### 3.4.1 System Architecture

HistoGlobe uses a classical client-server architecture of a Web-based information system. The user opens the application and interacts with it through the user interface in a Web browser, the *client* side of the system. The Web *server* is a remote computer that hosts the database and the middleware. The user interacts with the interface and the client-side application sends a request to the Web server for new data. The middleware checks the request and queries the necessary data from the database. It transforms the data and sends it back to the client. The interface shows the new information.

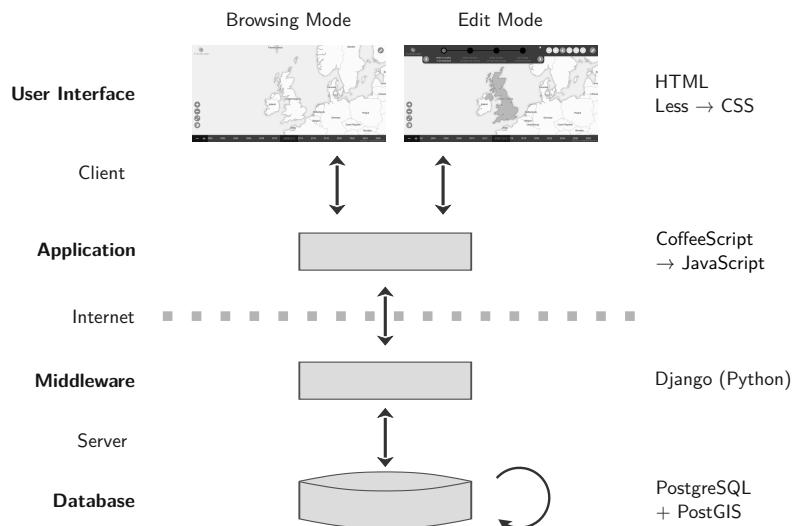


Figure 3.22: The system architecture of HistoGlobe

This clear separation between the data, the application and the user interaction in this chapter and in the system follows directly from the *model-view-controller* pattern: One part can be changed independently from the others parts: if the 2D map is replaced by a 3D globe, only the view changes, but the middleware and the database can stay untouched. Likewise, the implementation of a new database technology has no consequences to the view.

### 3.4.2 Server-Side Application

The underlying Hivent Model is implemented on the server-side of the application. HistoGlobe uses *Django*, a free and open-source web framework<sup>1</sup>, combined with *PostgreSQL*<sup>2</sup>, introduced in section 2.3.6. This allows HistoGlobe to take advantage of object-oriented concepts in a stable and fast relational database. Since the database is using a lot of geospatial data, *PostGIS* is used as a spatial database extension for PostgreSQL<sup>3</sup>. With these tools at hand, the Hivent Model from section 3.1 was implemented in a database model shown in figure 3.23. It is the final result of a highly iterative process that underwent many improvements and adaptations to new requirements introduced in the Human Centered Design process. The model is structured in two parts: The lower part of the Areas and the upper part of the Hivents and the historical changes.

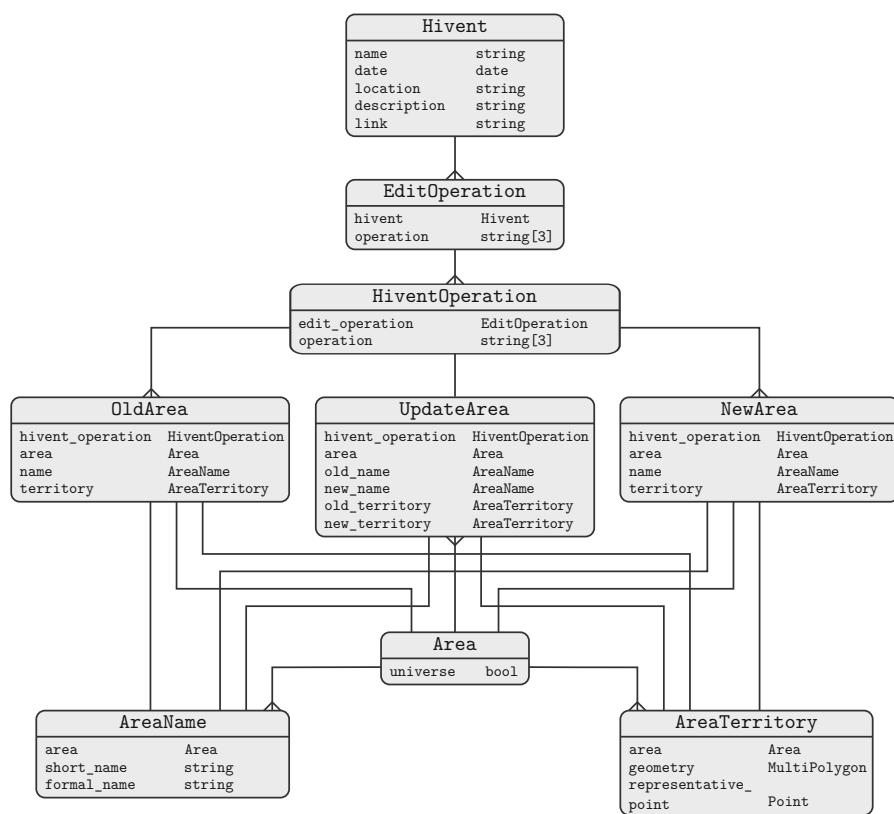


Figure 3.23: The Hivent Database Model

Each entity additionally has an `id` attribute, which is omitted for simplification purposes.

<sup>1</sup> *Django*, The Web framework for perfectionists with deadlines, URL: <https://www.djangoproject.com/>, last access: 27.05.2016

<sup>2</sup> *PostgreSQL*, The world's most advanced open source database, URL: <http://www.postgresql.org/>, last access: 31.10.2015

<sup>3</sup> *PostGIS*, Spatial and Geographic Objects for PostgreSQL, URL: <http://postgis.net/>, last access: 27.05.2016

**Semantic, Spatial and Thematic Domain** The Hivent Model introduces Areas as visible entities on the map with a name and a territory. In the database model they are represented by:

1. Area: semantic domain defining one identical Area with potentially changing name and territory. The universe attribute is true for  $\Omega$ , for the other Areas it is false.
2. AreaTerritory: spatial domain. A polypolygon describes the geometry of the territory and a representative\_point the position of the name label on the map.
3. AreaName: thematic domain. It is defined by a short\_name and a formal\_name.

**Temporal Domain** The main idea of the model is that the Areas can change over time. These changes are introduced by Hivents, the main entity of the eponymic model with five attributes: The name and a textual description of the Hivent, the point in time the Hivent happened (date), the Hivent location as a simple string and a link (URL) to the related article, serving as a historical source. Each Hivent consists of a set of EditOperations introduced by the user. They are an abstraction layer in the Hivent Model between the *Hivent* and the *HiventOperations*. They replace a set of OldAreas with a set of NewAreas and might update the name or the territory of one specific UpdateArea.

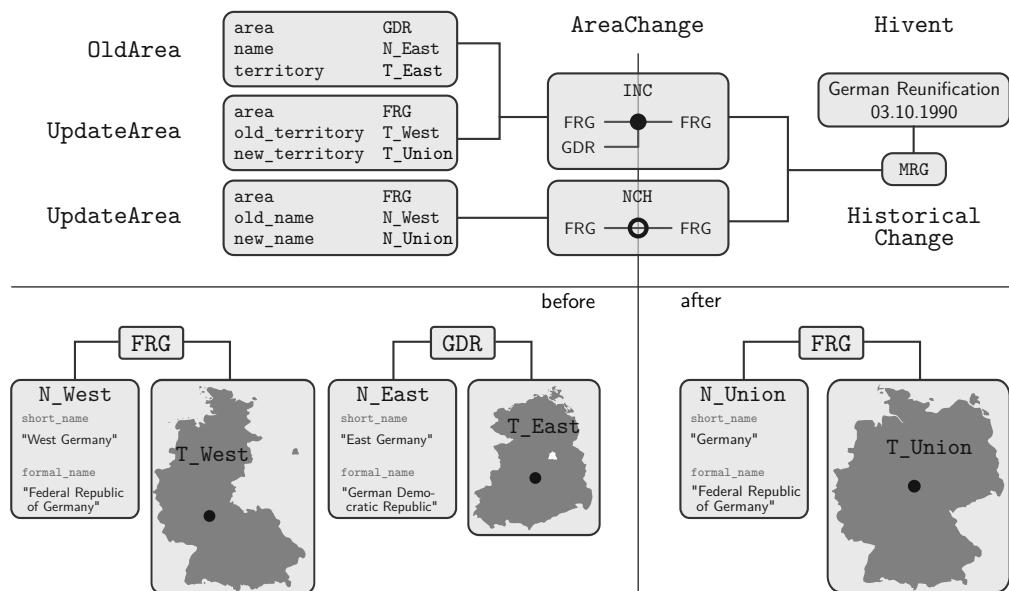


Figure 3.24: Visualization of the German Reunification in the Hivent Database Model

Figure 3.24 shows the Hivent Database Model at the example of the German Reunification on 3. October 1990. Before 1990, there were the Areas GDR (“German Democratic Republic”, East Germany) and FRG (“Federal Republic of Germany”, West Germany). A user introduced a Merge operation (MRG) in the Edit Mode between FRG and GDR. The new Area received the short name “Germany” and the same formal name “Federal Republic of Germany”, previously held by West Germany. Internally, the Edit Mode translates this to an INC of GDR into FRG and a subsequent NCH of FRG. One Area ceases, one Area is updated twice and no new Area is created.

**Initial Dataset** Section 2.2.5 explained the lack of data about historical countries. It is out of the scope of this thesis to create a large testing dataset with the historical countries in the world. The initial dataset consists of the following countries, their names and borders: the 193 UN members and 2 observer states – created by CRE operation – and seven countries with limited international recognition: Kosovo, Transnistria, South Ossetia, Abkhazia, Nagorno-Karabakh, Somaliland and Sahrawi Arab Democratic Republic. Their issue is explained in section 2.1.1. These seven Areas were created by DIS operations from their homeland on the day of their declaration of independence.

**Middleware** The Django web framework provides *view* classes that receive requests from the client, processes them, queries the necessary data from the database and returns an `HttpResponse` back to the client. In the naive implementation of the system, the middleware provides only two views for the two use cases:

1. `get_all` is initially called by the client side on loading the web service. The server responds to this `HttpRequest` with all data from the database in one `JSON` object.
2. `save_edit_operation` is called by the client after an Edit Operation has been completely created in the Edit Mode. In the last step, the client assembles the relevant data: the associated Hivent and HiventOperations, data about each OldArea, UpdateArea and NewArea. The view checks the data for integrity and stores them in the database. The method returns to the client a confirmation and a set of final `ids` for the entities stored in the database.

### 3.4.3 Client-Side Application

The main application of HistoGlobe runs on the client. As introduced in section 2.4, the software is built upon a module system. The modules used in this this implementation of HistoGlobe are emphasised in **bold** in the class diagram in figure 3.25. The remaining classes were created at run time by one of the seven modules. The classes are structured by their functionality regarding the Model-View-Controller pattern.

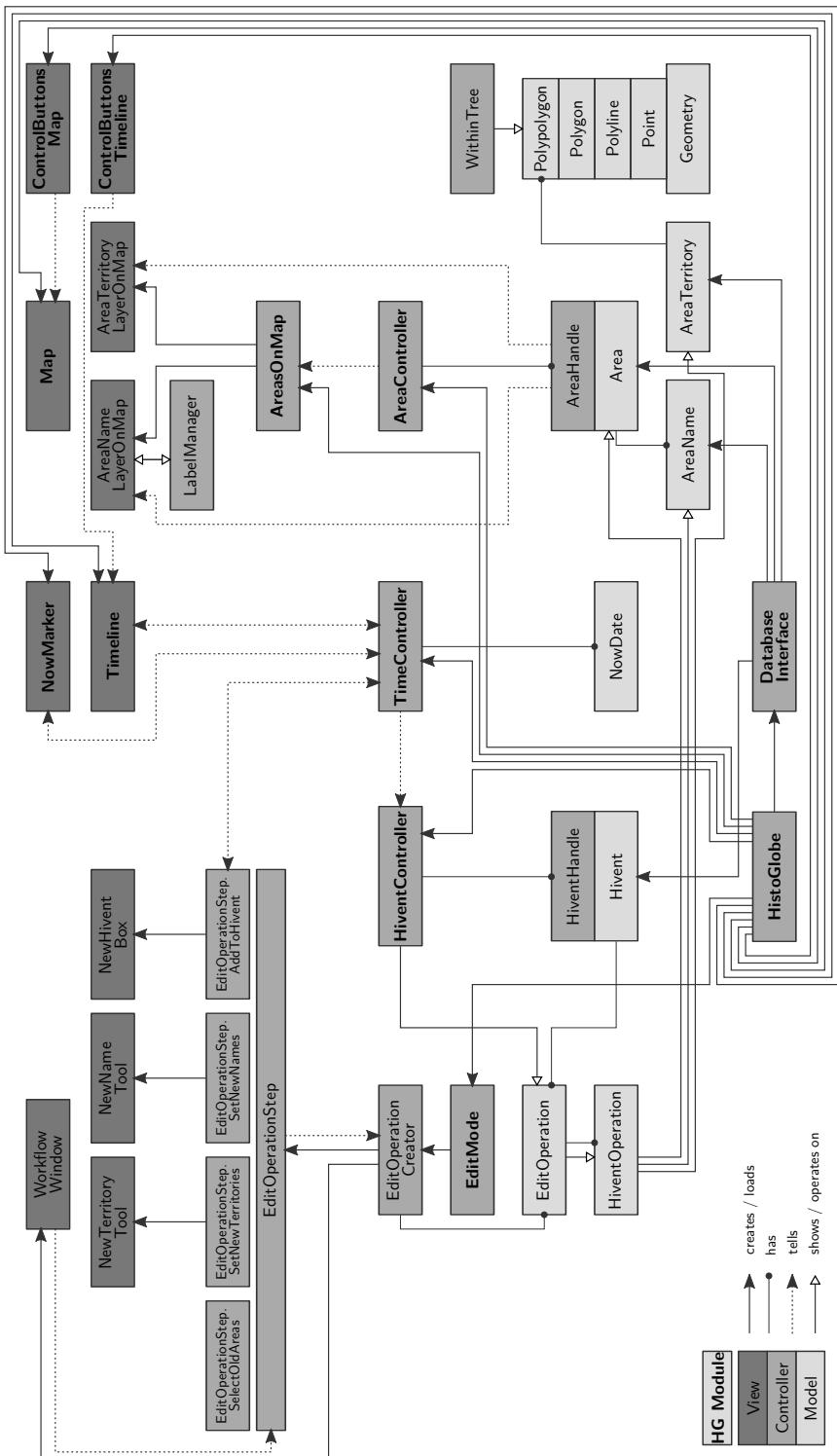


Figure 3.25: Class Diagram of HistoGlobe

**Initialization** The main HistoGlobe instance in the bottom initializes all modules, mainly the interface elements, the controllers and the DatabaseInterface. This class communicates with the middleware (section 3.4.2), loads data from and stores data in the database. Initially, each Hivent and the related EditOperations are created. Each HiventOperation is assembled by its associated set of OldAreas, NewAreas and the UpdateArea from the database model in figure 3.23. Afterwards, each Area, AreaName and AreaTerritory is loaded. A double-link is established to their associated HiventOperations via the startOperation, updateOperation and endOperation.

**Executing historical changes** HistoGlobe visualizes time on the interactive Timeline and the NowMarker showing the current date of the application: the NowDate. Both view classes can manipulate the current date by moving the Timeline or entering a date. The TimeController stores the NowDate and tells all other modules if the current visualization has changed.

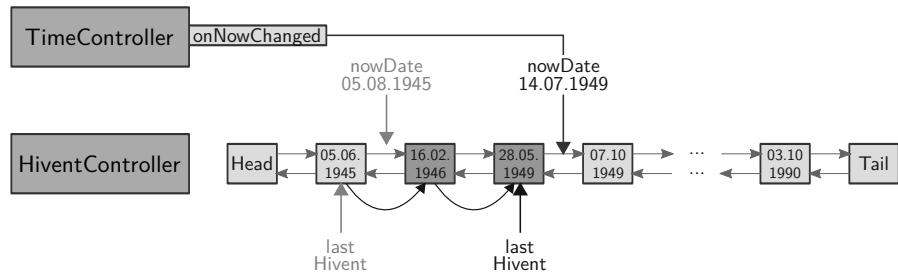


Figure 3.26: Detecting the next Hivent that happens in the HiventController

The core of the Hivent-based implementation is the HiventController. Figure 3.26 illustrates how it works: The controller stores a reference to each Hivent chronologically in a doubly-linked list, i.e. each Hivent knows the historically next and previous Hivent. Additionally, the controller stores a pointer to the last Hivent in the list that has happened and a copy of the current date. It listens to the TimeController – if the NowDate changes, the HiventController checks for the next Hivent if it has happened: The controller compares this new date with its current date checks if the Hivent.date is in between these two dates. If this is the case, the Hivent happens and all the EditOperations associated to this Hivent are executed on the map. The HiventController updates the pointer to the Hivent and checks for the next one until the next Hivent is outside this time span. If the nowDate from the TimeController is before the current date of the HiventController, it checks for Hivents backwards through the list and executes all the EditOperations backwards. This simple data structure allows to effectively and efficiently manage temporal changes of Areas on the map.

The initialization of the system works as follows: All Hivents in the list from the beginning until the NowDate of the TimeController are happening one after the other.

An EditOperation executes all its related HiventOperations. The main part of its source code, the execution function is shown in listing 3.1. If an HiventOperation is to be executed forwards, the following three steps happen:

1. For each newArea, the AreaName and the AreaTerritory that the Area has in the moment it gets historically created are associated to the Area. Afterwards, the Area is shown on the map: The AreaHandle associated to the Area has a function that shows the associated AreaNameLayerOnMap and AreaTerritoryLayerOnMap.
2. For each oldArea, the opposite happens: the name and territory are detached from the Area and the AreaHandle hides the Area from the map.
3. In the updateArea the AreaName or AreaTerritory is replaced by the newName respectively newTerritory. The update method of the AreaHandle updates the respective layers on the map.

In case the operation happens backwards, oldAreas and newAreas are swapped and the updateArea uses the oldName respectively oldTerritory instead. This is the same for all five operations.

```

1  class HiventOperation
2      #... initialization of the member variables
3
4      @oldAreas    = [] # {area, name, territory}
5      @newAreas    = [] # {area, name, territory}
6      @updateArea = {} # {area, oldName, newName, oldTerritory, newTerritory}
7
8      execute: (direction) ->
9
10     if direction is 1
11
12         for newArea in @newAreas
13             newArea.area.name =      newArea.name
14             newArea.area.territory = newArea.territory
15             newArea.area.handle.show()
16
17         for oldArea in @oldAreas
18             oldArea.area.name =      null
19             oldArea.area.territory = null
20             oldArea.handle.hide()
21
22         if @updateArea
23             if @updateArea.newName
24                 @updateArea.area.name =      @updateArea.newName
25             if @updateArea.newTerritory
26                 @updateArea.area.territory = @updateArea.newTerritory
27                 @updateArea.area.handle.update()
28
29         else # direction is -1 => backward change
30             # same as above, just each 'new' is replaced by 'old' and vice versa

```

Listing 3.1: Execution of an HiventOperation

**EditMode** The Edit Mode is the main contribution of this thesis to the HistoGlobe project. Its interface was introduced in section 3.3.3, this section shortly explains its implementation: When the user clicks a button of an Edit Operation in the upper right corner of the Edit Mode interface, internally a `EditOperationCreator` sets up the `WorkflowWindow` in the interface and the first `EditOperationStep` for this operation. Each of the four steps have different tasks that were explained in section 3.2.2. Throughout the workflow, the `EditOperationCreator` stores the data for the operation (selected old Areas, newly created Areas, update Areas, including the old and new territories and names) in an object. Each step can access this object and manipulate its content.

1. `SelectOldAreas`: The map is set in a `selectionMode` that allows requires the user to select a certain number of old Areas for the Edit Operation: one Area for DIS, REN and CES, one or two Areas for CHB and infinitely many Areas for MRG.
2. `CreateNewTerritories`: The map is set in an `editMode` that prevents any selection of existing Areas except for the ones that are manipulated. For each Area that is created or updated in this operation, the user needs to create the new `AreaTerritory` of the Area in form of a polypolygon with the `NewTerritoryTool`:

**CRE** Create one territory  $B^T$ , that is the territory of the new Area  $B$ . Check with the territory  $A^T$  of each existing Area  $A$  if the territories intersect ( $A^T \cap B^T \neq \emptyset$ ). If so, the territory  $A^T$  is updated by subtracting the new territory  $B^T$  from it via the complement (difference):  $A^T = A^T \setminus B^T$ . If the resulting territory  $A^T = \emptyset$  it means that  $A$  is completely covered by  $B$ , so it is wholly incorporated into  $B$  and is set as an old Area of the operation. If not, the territory of  $A$  is updated with  $A^T$  and  $A$  is set as an updated Area of this step that cedes parts of its territory to  $B$ .

**MRG** This step happens automatically without any need for user interaction. The new territory  $B^T$  is created by the cascaded union of the territories  $A_i^T$  of the selected Areas  $A_i \in A$  from the first step:  $B^T = \bigcup_{i=1}^n A_i^T$ . It is set as the territory of  $B$ .

**DIS** Each desired new Area  $B_i \in B$  needs an territory  $B_i^T$ . This is obtained by intersecting the temporary territory  $T_i$  that the user created in the `NewTerritoryTool` with the territory of the selected Area:  $B_i^T = A^T \cap T_i$ . This is repeated until the whole territory  $A^T$  is completely distributed among  $B_i \in B$ .

**CHB** If the user selected only one old Area  $A_1$  in the first step, the territory  $T_1$  created in this step by the `NewNameTool` will be set as the new territory of  $A_1$ . The surrounding Areas are handled in the same way as in the CRE operation. If two neighboring Areas  $A_1$  and  $A_2$  were selected, the new territories are obtained like this:  $T_1$  is seen as the basis of the territory  $A_1^T$ . It is temporariliy unified with its neighbor  $A_2^T$  to  $B^T = A_1^T \cup A_2^T$ . Then the new territory  $A_1^T$  is cut out off this temporary union with an intersection:  $A_1^T = B^T \cap T_1$ .  $A_2$  receives the remainig part as its new territory:  $A_2^T = B^T \setminus T_1$ .

In REN and CES no new territories are created.

3. CreateNewNames: For each Area that is created or updated in this operation, the user needs to create the new AreaName with a short name and formal name using the draggable NewTerritoryTool. The user writes the new names or selects them from the suggestions. Not only the name will be taken from the NewTerritoryTool but also the center position of the box will be the new representative point of the AreaTerritory. The user decides where the name of the Area should be positioned. This avoids the information visualization problem of automatic label placement.
4. AddToHivent: Using the NewHiventBox the user creates a new Hivent or selects an existing Hivent to which the Edit Operation is added to.

It was especially difficult to design each action to be fully reversible. For that purpose, an associated inverse of the action was stored in an `UndoManager` that works like a stack. If the user clicks the back button in the Workflow Window, the last action of the stack gets executed. When the last stage of the workflow (`AddToHivent`) is completed, the `EditOperationCreator` assembles the `HiventOperations` from the data gathered in the task. It sends it along with the associated Hivent to the server and finishes the operation.

**Within-Tree** The `NewTerritoryTool` in the second step of the workflow has to ensure that the drawn polygons are not self-intersecting and that no two polygons for one territory partially overlap each other. If they fully overlap, then they are holes in the polygon. A polygon consist of one *outer ring*, a closed polyline forming the boundary of the polygon, and a set of *inner rings*, closed polylines defining the holes in the polygon and representing first-order enclaves. Second-order enclaves are new polygons that happen to be positioned inside the inner rings of the other polygon. They can themselves have inner rings, which represent third-order enclaves, etc. In order to support nested holes, the *WithinTree* is introduced. The idea is to set up an hierarchical structure of polygons that contain each other. An example Within-Tree for a random set of polygons can be seen in figure 3.27.

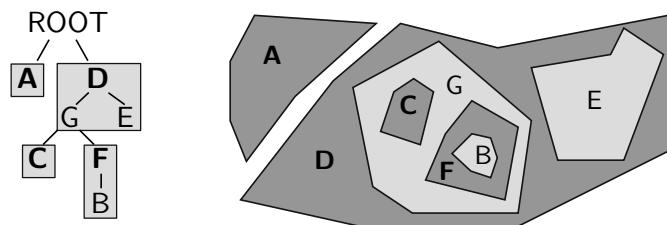


Figure 3.27: The Within-Tree (left) for the set of polygons (right)

The algorithm for inserting a polygon as a node into the tree is shown in listing 3.2. After the tree has been set up, the correct structure of polygons can be obtained by traversing the tree in the following custom order:

1. Remove the first child  $F$  of the root node and all its children  $C$  from the tree.
2. Insert each child of each  $C$  as a direct child of the root node.
3. Create a new polygon with  $F$  as the outer ring and each  $C$  as an inner ring.
4. Repeat until the tree is empty.

```

1  insert: (newNode, parentNode) ->
2
3      ## PREPARATION
4      # case 1) newNode also in 1 child of parentNode -> withinChild
5      # case 2) 1+ children of parentNode in newNode -> containChildren
6      # case 3) no hierarchical relation between newNode and any other node
7
8      withinChild = null
9      containChildren = []
10
11     for childNode in parentNode.children
12
13         if newNode.isWithin(childNode)                      # check if case 1)
14             withinChild = childNode
15             break # no other hierarchical relation to any other child possible
16
17         else if childNode.isWithin(newNode)                  # check if case 2)
18             containChildren.push(childNode)
19
20     ## EXECUTION
21
22     if withinChild                                      # case 1)
23         @insert(newNode, childNode)
24
25     else                                                 # cases 2 and 3)
26         # newNode is not in any child of parentNode => place it underneath
27         @_nodes.push(newNode)
28         newNode.setParent(parentNode)
29         parentNode.addChild(newNode)
30
31         for containChild in containChildren              # case 2)
32             # => detach from parent node and place them as children of newNode
33             containChild.setParent(newNode)
34             newNode.addChild(containChild)
35             parentNode.removeChild(containChild)

```

Listing 3.2: Insertion of a polygon node into the Within-Tree

**LabelManager** A major visualization problem that was sufficiently solved is the thesis is the label collision problem: Each active Area has both a territory and a name that should be shown on the map. Since no territories can overlap (precondition 5), they can all be shown. This is not true for the names of the Areas: The `short_name` of the `AreaName` is placed as a label in the `representative_point` of the `AreaTerritory`. Labels can overlap, because they can extend beyond their territory. To avoid this, some labels have to be hidden. A `LabelManager` decides for each label if it is shown or hidden. Each label gets an additional set of attributes:

- `isVisible`: status variable if the label is shown or not
- `priority`: the “importance” of the label determined by the size of the territory
- `boundingBox`: width and height of the text plus 5 pixel padding
- `coveredBy`: a list of higher-priority labels that cover this label
- `covers`: a list of lower-priority labels that are covered by this label

Label *A* covers label *B* if their bounding boxes intersect and *A* has a higher priority. The labels are stored in a doubly-linked `labelList` in a descending order by priority. The algorithm is based on the following heuristic: *A label is shown unless it is covered by an higher-priority label.*

When a new label is supposed to be shown on the map, it is inserted into the `LabelManager` like this: The correct position of the label in the `labelList` is found by checking with each element in descending priority if they overlap and if the priority is still higher. As soon as the first label with a lower priority is found, the new label is inserted before this label in the list. If there was a higher-priority label before that covered it, the new label is hidden – else it is shown. In the latter case all lower-priority labels are checked if they are covered by the new label – if so, they are hidden.

If an Area ceases also its name is hidden from the map. Additionally, the old label is removed from the `labelList`. Each label that was previously covered by the old label is not covered by it anymore. If no other label is still covering it, the label can be shown now.

If the user zooms the map, the `LabelManager` has to update the visibility status of each label. Zooming in means that each label has more space to its neighbors. No label has to be hidden, but a hidden labels can be shown if it is not covered by any other label anymore. Vice versa, if the user zooms out, the labels have less space to their neighbors. No label can be shown now, but a visible labels needs to be hidden if is covered by at least one other label now.



Figure 3.28: The resulting labels on the map in Europe 2016

Figure 3.28 shows the result of the LabelManager on the map of Europe in 2016. It is obvious that no two labels collide which was the main motivation for the algorithm. Also, the labels of the large countries Ukraine, Poland, Germany, France and the United Kingdom are shown. However, the label “Czech Republic” is hidden, because its bounding box intersects with the label “Germany”. On the other hand the labels of Monaco and Andorra are shown, although they are rather insignificant. But since there is enough space around them, they are shown. The LabelManager sufficiently serves the purpose of this thesis.

This chapter presented the iterative Human Centered Design approach to the core part of the thesis: The Hivent Model with the five low-level Hivent Operations describing an historical change. Additionally methods to edit these changes using six Edit Operations and a workflow were proposed. They formed the basis of the Edit Mode in the user interface of HistoGlobe. Finally the system architecture and some selected issues of the implementation in HistoGlobe were presented. However, this state of the application only covers forward changes at the end of the timeline and it is based on the wrong assumption of complete certainty about historical data. The next chapter evaluates this state and develops extensions that can cope with the uncertain nature of history.

# Chapter 4

## Extensions

This chapter evaluates the data model and the implementation developed in the previous chapter 3. It focuses on the important aspects of the uncertain nature of history in section 4.1. The second part of this chapter 4.2 develops extensions to the Hivent Model and the Edit Mode for HistoGlobe that deal with these problems of uncertainty and disagreement.

### 4.1 Evaluation

In order to evaluate a model it is important to understand the concepts of *disagreement* and *uncertainty*, *precision* and *accuracy*. The model in an information system tries to resemble the real world as good as possible and necessary. Uncertainty and disagreement are two different kinds of problems in two different domains. *Disagreement* describes a conflict in the real world. On a contrary, *uncertainty* is the imperfection of a model to describe the real world. It can be expressed by accuracy and precision: The better a model simulates the reality, the more *accurate* or *correct* it is. That means, the closer it gets to the target, the higher is the accuracy. *Precision* or exactness describes how similar the results of the model are compared to each other, independent from how well they resemble the real world. That means a precise model gets the same results over and over again (see figure 4.1).

If the border between the Principalities of Transsylvania and Wallachia is deducted from an historical map of 1600, the course of that border is inaccurate to a certain degree, because the map does not show the real world correctly. However, it can be modelled in the system very precisely, because the coordinates of the border points are stored as floating point numbers. The situation with Taiwan explained in section 2.1.1 is different: the disagreement is there already in the real world. The conflicting versions of the story can be modelled very precisely, but in order for the model to also be accurate, it needs to support declaring an Area as a contested territory.

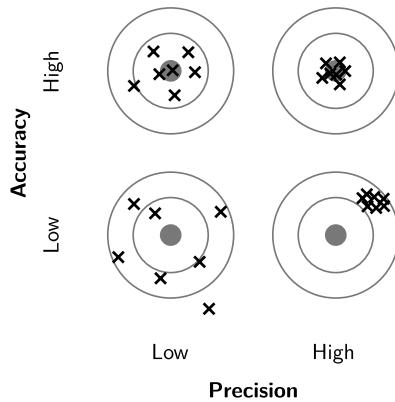


Figure 4.1: The difference between accuracy and precision

#### 4.1.1 Analysis of the Data Model

Each aspect of the Hivent Model proposed in section 3.1 is based on the prerequisite of full certainty of the data. The Hivent Model assumes that the dates of the Hivents, the names and territories of the historical and current Areas and the historical relations between Hivents and Areas are accurate and reasonably precise. However, this assumption is far from valid. Section 2.2.1 explained that in historical research, uncertainty is one of the major problems a historian has to deal with on a daily basis: primary sources can be biased towards the author, are certainly incomplete, inaccurate or conflicting with other sources. The acquisition of objective historical data is impossible. The further documents go back in time, the lower is the expected accuracy. Since all the information in the Historical Geographic Information System is based on primary sources, the data in the system inherits this integral lack of accuracy. This section evaluates the different components of the Hivent model in terms of how accurately and precisely they resemble the complex nature of history.

**Hivents** The model contains the name, date and location of an Hivent. In reality, the name can have different versions: a long, official version and a short common version. The commonly known “Treaty of Versailles” is officially called the “Treaty of Peace between the Allied and Associated Powers and Germany”. The name is different in other languages affected by the treaty. There can even be different versions of the name from different perspectives within the same language, e.g. the “American Civil War” was alternatively called “War Between the States”, “War for Southern Independence” or “War of Northern Aggression” depending on the perspective. The Hivent model does not account for different languages and versions and is therefore not very precise.

The `Hivent.date` is supposed to represent the temporal dimension of an historical event. While an historical change itself is discrete and happens at exactly one time point, the historical event yielding this change might not. The decisions of the “Congress of Vienna” came into effect on 9. June

1815, but the congress itself took place from September 1814 until June 1815. Another phenomenon becomes apparent in the “Convention for the Extension of Hong Kong Territory” (1898) which had a predefined length of 99 years. The treaty therefore has two dates in which historical changes happened: the date the treaty came into effect (Hong Kong becomes part of the United Kingdom) and the date it stopped being in effect (Hong Kong is handed over to China). Other interesting aspects are different calendar systems used in different parts of the world throughout history: the October Revolution in Russia (1917) happened in November in their Gregorian Calendar system, but in October in the Julian Calendar. Also timezones can play a crucial role: The German Instrument of Surrender ending World War II in Europe came into effect on 8. May 1945 at 23:01 Central European Time. In the Soviet Union that happened at 1:01 Moscow Time. That is why the celebration of the Victory Day in Moscow is celebrated on 9. May 1945. While the `Hivent.date` field in the data model works with timezones, it does not support different calendar systems or multiple dates associated with one `Hivent` which limits its precision.

The event location is represented by the `Hivent.location` name of a place, which can e.g. be a city, a battlefield or a region. The model is not very precise, because the actual geospatial location or region in which an historical event happened is not stored in the system. Additionally, it does not support names in different languages.

The `Hivent` model – like any model – is incomplete. There are relevant aspects about an historical event that are interesting for the purpose of the HGIS that are omitted in the model. In some cases the actors of an event are important, e.g. in a peace treaty. The model however does not answer the question “Who?”. Even more severe is that there is no support of the most important question: “Why” something has happened. The motivation behind the actors or the coherences with other events are not supported in the model.

**Areas** Also the model of an abstract Area, consisting of a territory and a name, is problematic in terms of accuracy and precision. As it has been discussed in subsection 2.1 in detail, it is impossible to objectively model a country without running into conflicts. In the preconditions of the model it is fixed that there can not be overlapping territories at the same time – although the nature of a contested territory is that it is claimed by more than one party. Whereas the data model allows to represent it as an own Area, it does not support any evidence of uncertainty about the status of an Area, e.g. being contested or international recognition. The model does also not support hierarchies of Areas, e.g. countries within countries, provinces or overseas territories. Some parts of countries have varying degrees of autonomy which is also not supported by Areas in the `Hivent` Model.

The `AreaName` has the same problem than the `Hivent.name`, but it is more precise, because it contains both the formal and the short name of a country. However, uncertainty about the name, different versions or languages are not supported. More problematic is the `AreaTerritory`: The two assumptions in the preconditions of the `Hivent` Model in section 3.1.2 – constant coastline and no sea territory – are inaccurate. The primary source for territories of countries are historical maps.

The process of extracting a boundary from an historical map introduced in section 2.2.5 is error-prone and yields a loss of accuracy in each step. The level of inaccuracy depends on the resolution, map projection or the colors used in the map. In the data model it is not possible to provide information about the expected accuracy of a territory. Another problem is that the territory is stored as a whole polypolygon. Different parts of the border can have a different type or level of certainty status, e.g. one part a well-established and demarcated border to neighboring country  $X$ , but the border to neighbour  $Y$  is contested. The `AreaTerritory` data model does not account for these differences.

**Hivent Operations** Another aspect to evaluate is if the Hivent Operations as the basis of the spatio-temporal Hivent Model are collectively exhaustive, i.e. each possible case in the real world can be expressed with a combination of operations, and mutually exclusive, i.e. no operation can be replaced by a combination of two others [n.A].

It is easy to see that a NCH operation is mutually exclusive to the other operations, because it is the only one that manipulates the name of an Area by preserving its identity. `UNI` and `SEP` are inverse, just like `INC` and `SEC` meaning that they are by the nature of an inverse exclusive. `INC` requires one update Area, multiple old Areas and no new Area, whereas `SEP` does not have an Area that preserves its identity and has several new Areas. Therefore `INC` and `SEP` are mutually exclusive, just like their inverse pair `UNI` and `SEC`. Therefore, the only possible combinations left are `UNI` and `INC` and `SEP` and `SEC`. While both pairs are geographically equivalent, because they unite respectively separate the territory in the same way, they are historically distinct: `INC` and `SEC` hand model one update Area whose identity is preserved and territory updated while the others are incorporated into respectively secedes from this Area. In `UNI` and `SEP` on the other hand, there is no Area that is preserved. These two cases can not be mapped onto each other. Therefore it has been shown that all operations are mutually exclusive.

To show that the collection of Hivent operation is exhaustive, all possible changes in the history of a country regarding its identity (formal name), short name and territory have to be examined. In reality, a historical country was once created, possibly underwent several changes of its short name and territory and ceased one day. The creation of an Area on previously unclaimed land can be expressed by a `SEC` from  $\Omega$ . If it originated from one other Area, it is represented as the resulting Area of a `SEP`. If it had several predecessors, it is expressed by a `UNI`. However, the model does not support reincarnation, i.e. an Area continues the existence of another historical Area that has previously ceased. The data model is only geared to direct historical relationship. In the lifetime of the country, it can change its short name and its territory. A NCH operation enables the first case, an `INC` or `SEC` the second case. Since operations can be combined at the same point in time, all combinations of identity-preserving operations are possible. If a country ceases, it can either have none, one or multiple direct historical successors. The first case is represented by an `INC` into  $\Omega$ , the second case by a `UNI` and the third case by a `SEP`. Therefore, with the exception of reincarnation, all possible historical changes regarding the name and the territory of a country can be expressed with the Hivent Model.

Overall, the Hivent Operations are a big strength of the Hivent Model, since it requires only five different operations. The Hivent model also reasonably represents countries as Areas with a name and a territory. Countries with broad international recognition and a clear territory can be expressed without a problem. However, the model very poorly accounts for different levels of uncertainty in historical and geographic information: imprecise and inaccurate sources, different viewpoints and interpretations, contested territories, changing coastlines or different languages are not supported

#### 4.1.2 Analysis of the Application

Additionally to the data model, also the implementation of HistoGlobe can partially be evaluated. The interface resulted from a Human Centered Design process illustrated in chapter 3. It underwent constant changes in multiple iterations with several user tests. Although there is a lack of quantitative data, the qualitative studies showed that both the Edit Operations and the interface as a whole were broadly understood. The only problematic operation was a border change: drawing the new territory of one Area and automatically adapting the other territory was not perceived as intuitive. As the name suggests, the users would have wished to actually change the border by editing the border points and moving it. Many users also complained that there is no feature to import existing geometries from external sources or from an historical map. While this is a highly desirable feature, it was out of the scope of this thesis but has a high priority in the future work.

In terms of performance system performs well with the current content of the database: It takes about 3.5 seconds to initially load all the data from the server to the client. As explained in section 3.4.2 all the data from the database is loaded at once. While this behaviour is not scalable, for the initial dataset it was sufficient: It took 3.5 seconds to load it. As soon as the amount of data in the database increases significantly, especially because of the territories, the approach to load all at once has to be changed to a gradual loading process: Initially the client sends the current date and the center of the viewport to the server. The server sends the currently active Areas close to the viewport center first. The goal is to minimize the time until the user sees a significant portion of the relevant data on the screen. This can also be achieved by storing different levels of detail of the vector data. Initially the server would send only the coarsest level to the client, then gradually increasing the granularity until the whole data is loaded on the client.

Given HistoGlobe would additionally to the current countries store a significant amount of historical countries since the beginning of the common era 2000 years ago. An estimation of the total memory consumption of the territories in the database is valuable. Currently there are about 200 countries on Earth. Wikipedia lists another 700 modern states in the “List of former sovereign states”<sup>1</sup> and additionally about 1000 in lists of states of the Classical Age, Iron Age, Late Antiquity and the Middle Ages. However, some of them appear on several lists if the empires have existed throughout

---

<sup>1</sup> *List of former sovereign states*, [https://en.wikipedia.org/wiki/List\\_of\\_former\\_sovereign\\_states](https://en.wikipedia.org/wiki/List_of_former_sovereign_states), last access: 02.06.2016

the Ages. As a broad estimate, there were probably not more than 1500 historical “countries” since the beginning of the common era. Each of them probably underwent multiple territorial changes. The “List of national border changes from 1815 to 1914”<sup>2</sup> collects about 75 border changes, about 100 border changes are listed since World War I. Therefore the estimation of 1 international border change per year seems accurate. Given this rate of change stayed constant, there were about 2000 border changes since the beginning of the common era. They are especially storage-intensive, especially because of the precondition with  $\Omega$ , each time the territory of one Area changes, at least one other will do so as well. As a conservative estimation, the total number of territories in the system will not exceed  $2000 \times 2 + 1500 \approx 5000$ . The current dataset of Natural Earth Data has a decompressed size of 8.8 MB for about 200 countries, that is about 0.05 MB per country. Given that there would be accurate and precise measurements of all historical countries in the previous 2000 years, the total memory consumption for all coordinate data would not exceed  $5000 * 0.05 \approx 250$  MB. That is an insignificant storage requirement and fits into the main memory of each modern computer system.

## 4.2 Modelling Uncertainty

The first part of this chapter presented a list of shortcomings of the current Hivent Model and the implementation, most of them focussing on the problem of uncertainty and disagreement. The purpose of this section is to develop approaches to increase the accuracy of the model with respect to uncertainty. A higher accuracy usually leads to a higher complexity of the data model. This trade-off has to be thoroughly taken into consideration when supporting a new feature. For that reasons some problems are be ignored in the rest of this chapter:

- The preconditions of the Hivent Model will not change, i.e. countries will stop on their coastlines and the distribution of water on the planet has not changed in history – otherwise the model would need to support not just sudden events but also gradual processes. This out of the scope of the thesis.
- The support for different historical perspectives on the same Hivent, e.g. different names and descriptions or even different historical changes would create a research tool with great potential. It would enable the possibility for different versions of history based on alternative scenarios: “What if X would have (not) happened?”. However, this would significantly increase the complexity of the system and would also be very subjective.
- The introduction of different calendar systems would not increase the accuracy of the model significantly. The dates in the system must all stick to the Julian calendar, which is a reasonable requirement to avoid unnecessary complexity.

---

<sup>2</sup> *List of national border changes from 1815 to 1914*, [https://en.wikipedia.org/wiki/List\\_of\\_national\\_border\\_changes\\_from\\_1815\\_to\\_1914](https://en.wikipedia.org/wiki/List_of_national_border_changes_from_1815_to_1914), last access: 02.06.2016

In order to tackle the remaining shortcomings of the current concept, both the user interface and the data model have to be extended. The main problem when dealing with uncertainty is that from the perspective of a computer, there is no such thing as an uncertain value of a variable. That is why all sorts of uncertainty must ironically be quantified: The computer needs to know exactly how uncertain the user is about the data.

#### 4.2.1 International Recognition



Figure 4.2: Newly designed and extended buttons for Edit Operations.

Two new Edit Operations shown in figure 4.2 are introduced: CHS changes the status of an Area and REC declares a new recognition, i.e. one country internationally recognizes another one. This is simply performed by selecting two Areas on the map, whereas the first Area recognized the second.



Figure 4.3: New Edit Operation: Recognition – sets up the recognition of one Area to another.

#### 4.2.2 Uncertain Borders

Also the Edit Operation workflow is altered. The second step SetNewTerritories defines the territory of the new Area(s). Instead of drawing the whole territory as a set of polygons, the user draws one borderline at a time, geometrically as a polyline. This has the main advantage that each part of the border is treated separately. The borderline is assigned a degree of certainty, in the interface controlled by a horizontal slider, in the model as a certainty value ( $\text{certainty} \in ]0..1]$ ). Absolute certainty (1.0) creates a sharp and crisp line on the map. In case of uncertainty ( $\text{certainty} \in ]0..1[$ ) three different visualization methods are introduced:

1. Blurred Border: The higher the uncertainty, the wider and more blurry the border.
2. Border Corridor: With increasing uncertainty, the offset around the actual border line extends. That creates a corridor in which the actual border is probably in.
3. Blurred Border Corridor: The combination of the first two approaches.

A simple model for the calculation of the blur factor, line width and offset distance is:

$$f(c) = -1 \cdot S \cdot \ln(c) + I$$

where  $c$  is the certainty factor,  $S > 0$  is a scaling factor and  $I$  is the initial value (for width: 1 px, for blur: 0, for offset: 0 px). In the example in figure 4.4 the scaling factor is  $S = 4$ . In the Blurred Border Corridor method, the scaling factor for line width and the blur factor was halved. Further analysis and user testing are required in order to decide for one approach to develop further.

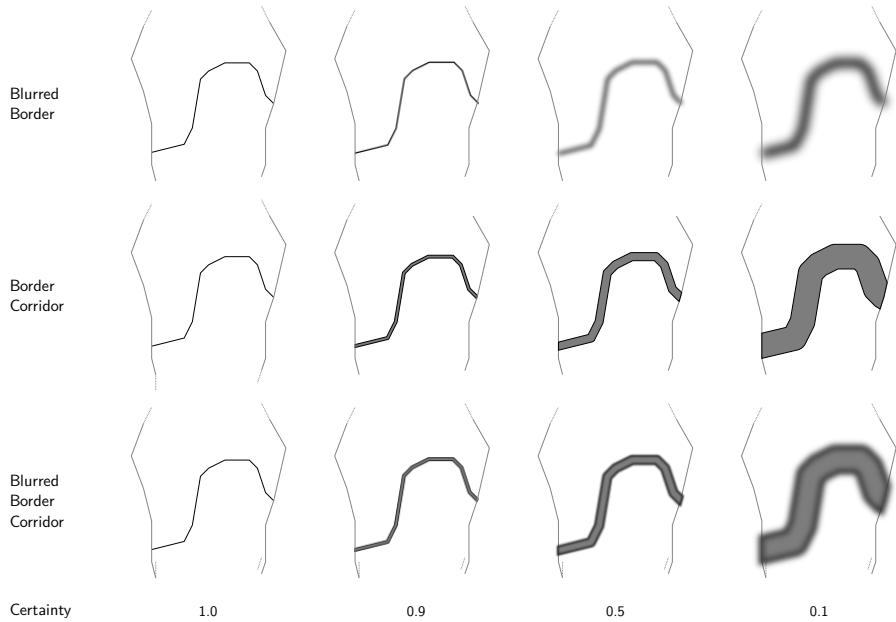


Figure 4.4: Three different methods to visualize uncertain courses of a border

Another advantage of the input of borderlines instead of territories is that once the model is further advanced, coastlines can be continuously changed according to an appropriate change model. This can be applied solely to the coastlines without affecting the interior borders.

A new border point automatically snaps to an existing border point, if the mouse position is close enough to it (an appropriate threshold might be 5 px). This allows for a smooth workflow and is

required to create closed polygons. When the user finished a territory by defining all surrounding polylines that create a closed ring, the polygon is assembled. If a borderline meets another borderline at an interior node, the polyline splits up into two parts so that each meeting point of borders is the start or end point of a polyline. This way the spatial integrity is maintained.

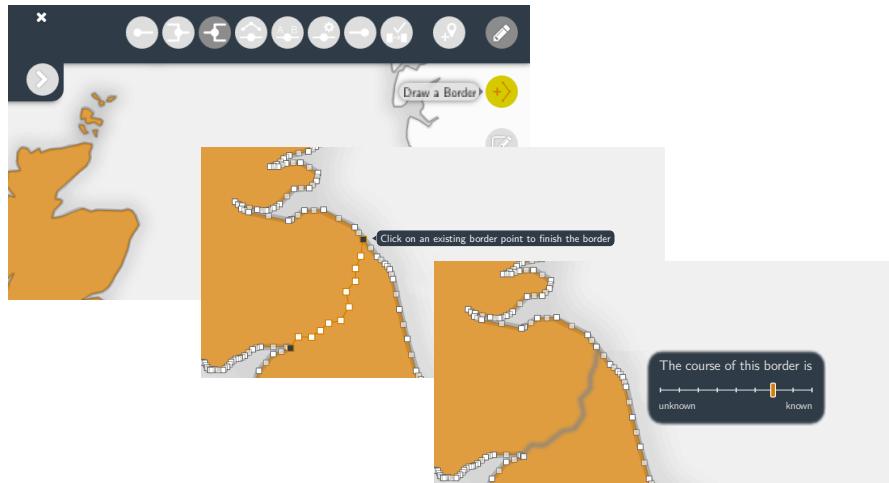


Figure 4.5: Drawing historical borders instead of full Areas and definining a level of certainty.

If the created territory overlaps with an existing territory, its intersection will create a separate territory. In the next step, this territory can then be defined as a contested Area or defined as a part of another Area. If the step yields an empty territory that was claimed before, it can later be defined as a neutral zone or unclaimed land.

#### 4.2.3 Special Areas

To treat special Areas differently, a new step in the workflow is defined. After the territory and the name of a new Area have been defined, a special status can be assigned to it.

1. A *fully sovereign country* is a political entity with full sovereignty over its territory and people and significant international recognition, e.g. Estonia.
2. An *unclaimed land* is a territory that is not claimed by any political entity, e.g. Antarctica.
3. A *neutral zone* is often a buffer zone between two conflicting parties, e.g. currently on Cyprus.
4. A *contested territory* is claimed by at least two different political entities of the same hierarchical level, e.g. the Kashmir region between India and Pakistan. It is also suitable for Areas that have claimed independence from a sovereign country but are not yet recognized as such, making their whole territory contested, e.g. Nagorno-Karabakh (see figure 4.6).

5. A territory can be a subordinate part of another country with a certain degree of autonomy ( $\in [0..1]$ ). Fully subordinate parts of a country, like a US State or a German Bundesland have no autonomy (0). Autonomous countries within another country, like England to the United Kingdom or Greenland to Denmark, receive have certain a degree of autonomy ( $\in]0..1[$ ). The value 1 is excluded, because full autonomy means the territory is a sovereign country.

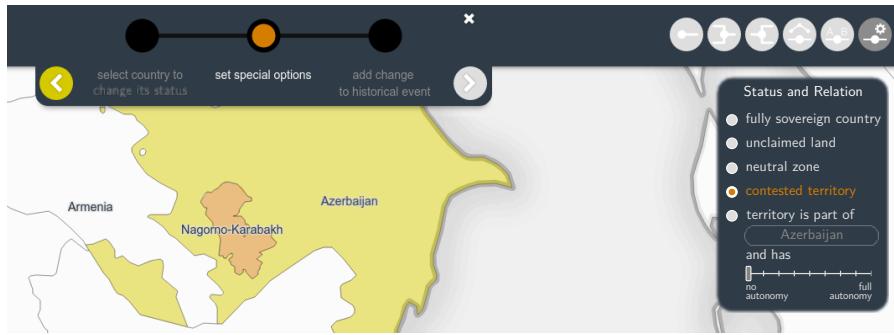


Figure 4.6: Defining a special status or relationship to a territory.

#### 4.2.4 Content from Wikipedia

As it has been envisioned in section 2.2.5, Wikipedia stores a lot relevant data for HistoGlobe which is not straightforward to reuse, because of the Hivent-based data model and the lack of broad standardization in Wikipedia. This section assumes that the data could be parsed and processed easily. When defining the name of an Area, the user will get actual name suggestions from a collection of current and historical countries in Wikipedia. That saves time for researching short and formal names of Areas.

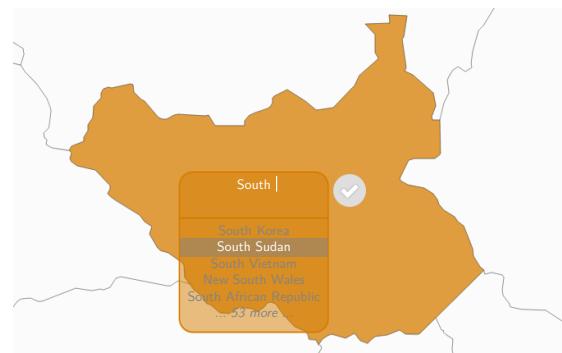


Figure 4.7: Getting suggestions for the name name from Wikipedia.

**New Hivent Box** The visualization of an Hivent is split up into three parts (see figure 4.8):

1. An information section storing important meta data of the event location, the dates (timespan in which the event happened), a description and the link to the wikipedia article (if given).
2. A section storing all Edit Operations associated with that Hivent. Each operation is visualized and is assigned a date at which this event came into effect.
3. A multimedia section stores images, videos, audio files and documents and their sources associated to the historical event.

Similar to the extension of the Area name step, also Hivent names can be chosen among a collection of Wikipedia articles describing historical events. Selecting a name from a wikipedia article automatically fills the information section and adds multimedia files from the wikipedia article. The Edit Operation is automatically entered in the section. With this separation, different Edit Operations at different dates can be associated with one Hivent, increasing the accuracy and precision of the Hivent Model.

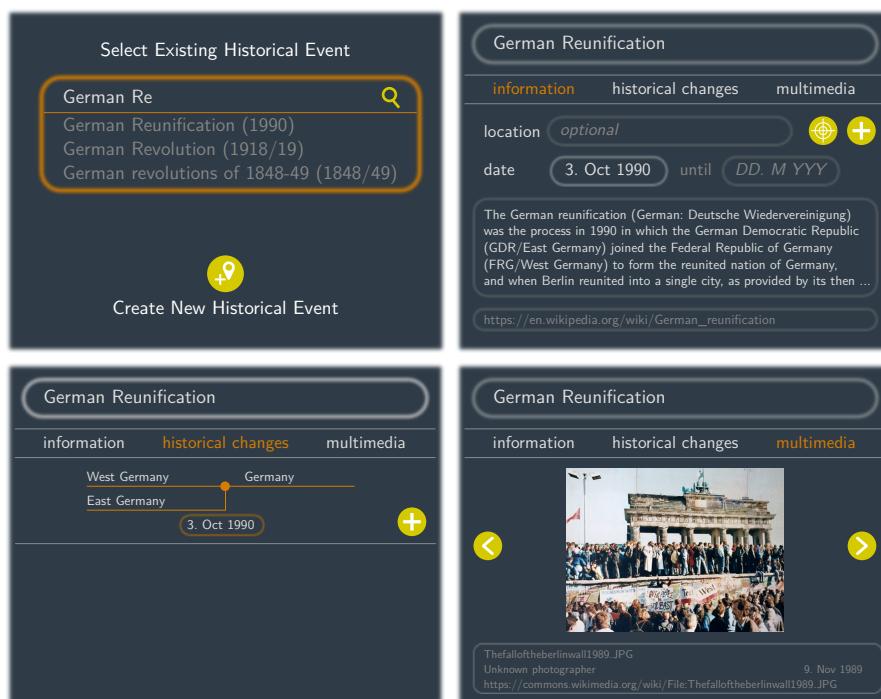


Figure 4.8: Creating a new Hivent and adding the newly created Edit Operation.

In the long run, HistoGlobe can be synchronized with Wikipedia or even be designed as an extension for Wikipedia articles about current or historical countries.

#### 4.2.5 Extended Hivent Model

To account for the extensions in the interface, also the data model has to be adapted. Mainly the thematic domain of the Hivent Model changed, since the status or an Area, its recognition or relation to other countries is a non-spatial attribute of the data. The new Edit Operations that were introduced need to be internally expressed by Hivent Operations. For that purpose the name change operation (NCH) is abstracted to a thematic change operation (TCH). Instead of changing only the `old_name` to the `new_area` of its associated update Area, it can also change its `old_status` to a `new_status`. It can be visualized in exactly the same way, since it is just as well an identity-preserving change.

More thoughts have to be spent on the question how to express a new recognition of an Area in an Hivent Operation. Additionally, the introduction of hierarchical layers of Areas adds new requirements to the Hivent Model: All Hivent Operations can potentially be performed on the same level of hierarchy as before, but also downwards or upwards. The example of the fictional secession of Scotland from the United Kingdom in 2018 would be simple: The user could simply select Scotland as a second-level Area inside the first-level United Kingdom and secede Scotland from the Union by updating its status to a fully sovereign first-level Area. The design and implementation of the data model requires further analysis.

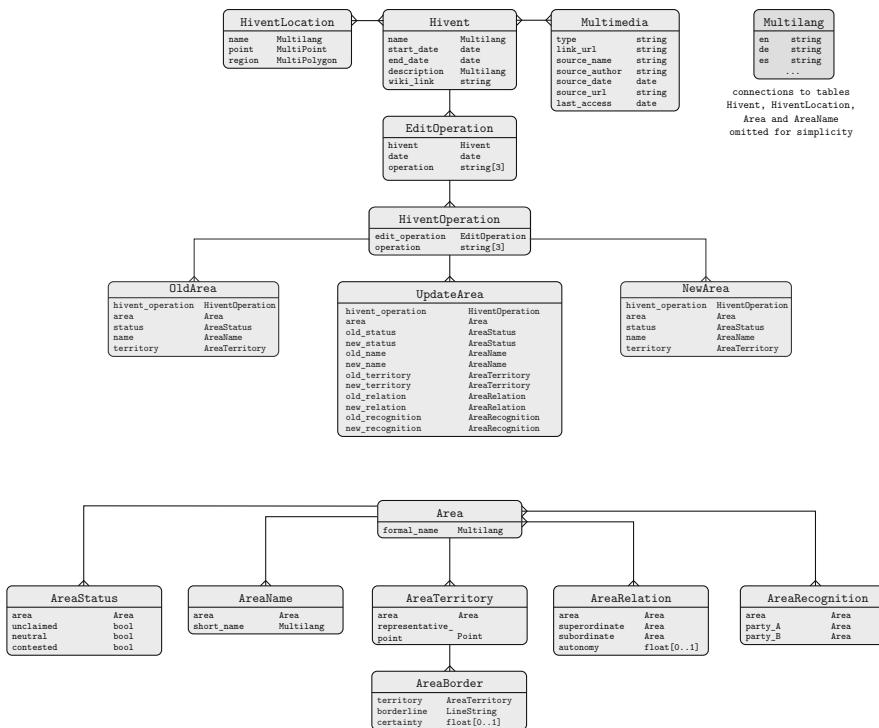


Figure 4.9: The new data model to support the developed approaches regarding uncertainty

The Hivent database model from section 3.4.2 is extended by the following entities and relations:

1. Creation of a `Multilang` entity to store a name of an Hivent, its location or an Area name in different languages.
2. Outsourcing of the `HiventLocation` into an own entity to identify a location with a name and a geospatial reference.
3. Creation of a `Multimedia` entity to manage multimedia files associated to an Hivent.
4. Attachment of a date to an `HistoricalChange`.
5. Inclusion of the `formal_name` into the `Area` to emphasize it as the identifier.
6. Creation of an `AreaBorder` with a borderline. A set of `AreaBorders` create one `AreaTerritory` which is associated to the Area. Each change of an `AreaBorder` creates one or two new `AreaTerritory`/ies.
7. Creation of an `AreaStatus` and an `AreaRelation` to account for special status of an Area alone or in relation to another Area with a certain level of autonomy.
8. Creation of an `AreaRecognition` to account for international recognition of one Area to another one.
9. Adaption of the `AreaChange` entity to model a change of each possible property of an Area.

This chapter provided an analysis of the data model and the implementation developed in the previous chapter. The main problem with the current state of the application is that it does not support any kind of ways to cope the inherent problem of uncertainty and disagreement in history. For of the shortcomings from the evaluation design solutions for the user interface of HistoGlobe were developed and integrated into the Hivent Model. The final chapter of this thesis summarizes the achievements of this and the previous chapter, embeds the work into the current state-of-the-art and concludes the thesis as a whole.

# Chapter 5

## Summary

This Master's thesis started with the motivation to lay the foundation for an Historical Geographic Information System that shows the history of countries on Earth. While there are many interesting visualizations about historical topics, wars or events, there is no such thing as an interactive historical world atlas. This may be due to the fact that there is no comprehensive collection of historical data and that the whole nature of history is that everything we know is potentially uncertain. Even the commonly accepted concept of a "country" is impossible to define without running into conflicts. To create an information system in such a complex domain is very challenging.

### 5.1 Results

To summarize the most important results and contributions of this Master's thesis, the four research questions are finally answered.

**1) What type of historical changes can happen in the development of countries in time and space?**

The history of countries is very complex. The problem space of this work was limited to the territories and the names of countries. Except for the coastlines, the more interesting interior borders have always changed due to sudden events. The same is true for the name of countries. If a universe  $\Omega$  is defined as an ever-existing territory that initially covered the whole surface of the Earth, then this thesis has shown an interesting result: With the exception of the rare case of reincarnation, everything that has ever happened to names and territories of countries can be expressed by five basic historical changes: Unification, separation, incorporation, secession and name change.

### **2a) How can these changes be modeled in an information system?**

These five changes are modelled as *Hivent Operations*. Countries are represented by abstract *Areas* with a formal name, a short name and a territory. Areas can change due to *Hivents* – *Historical events* – that introduce the Hivent Operations to cease or create a set of Areas or to update the properties of an Area over time.

### **2b) How can these changes be edited by humans in a user interface?**

While the Hivent Operations are

edit operations well understood by a human 1:n hivent operatoins wlel understood by the machine

The Hivent Operations are the main contribution of this Master's thesis.

### **3) How can the model handle uncertainty and disagreement in history?**

## **5.2 Problems**

## **5.3 Prospect**

implementation of extensions + other shortcomings identified in evaluation + retrospective updates and backward changes + import of historical data from Wikipedia + integration of tool for semi-automatic extraction of historical maps

HistoGlobe would have the potential to be a good Historical Geographic Information System showing the history of countries in time and space.

If scholars in humanities use the Edit Mode of HistoGlobe to continuously improve the historical data in the system, animate historical countries and discuss historical events, then HistoGlobe would be a comprehensive historical world atlas as envisioned by its inventor. It would help teachers to explain history to their teachers and students to understand the seemingly complex course of history

An implementation of these would yield a Historical Geographic Information System that is well suitable for visualizing and editing the history of countries on Earth.

has a great potential to teach, learn and understand processes in the past. A system that is able to tell *what* happened and *where* the historical event has influences on and *when* the event happened happened might be tool to answer the most important of all questions: *Why* it happened?

finally to reason about if wars actually make sense. If John Lennon is right, then this HGIS has come to its ultimate end: all Areas have unified. All the people living life in peace. You may say I am dreamer, but I am not the only. I hope someday you'll join us. And the world will be as one.

# Bibliography

- [AHA] AHA - American Historical Association (Andrews, Thomas and Burke, Flannery). *What Does It Mean to Think Historically?* URL: <http://historians.org/publications-and-directories/perspectives-on-history/january-2007/what-does-it-mean-to-think-historically> [cited 23 10 2015].
- [Amn] Amnon Sella. *Custodians and Redeemers: Israeli Leaders' Perceptions of Peace, 1967-79.* URL: <http://www.jstor.org/stable/4283114> [cited 26 04 2016].
- [Arc] ArcGIS Resources. *ArcGIS Help 10.1.* URL: <http://resources.arcgis.com/EN/HELP/MAIN/10.1/index.html#/0031000000q8000000> [cited 03 11 2015].
- [Arm92] C. Armenakis. Estimation and organization of spatio-temporal data. *Proceedings of the Canadian Conference on GIS92*, 1992.
- [BBCa] BBC News. *Nagorno-Karabakh profile.* URL: <http://www.bbc.com/news/world-europe-18270325> [cited 26 04 2016].
- [BBCb] BBC News. *Somaliland profile.* URL: <http://www.bbc.com/news/world-africa-14115069> [cited 26 04 2016].
- [BBCc] BBC World News. *Russia recognises Georgian rebels.* URL: <http://news.bbc.co.uk/2/hi/7582181.stm> [cited 26 04 2016].
- [Bol08] P. Bolstad. *GIS Fundamentals: A First Text on Geographic Information Systems.* Eider Press, 2008.
- [Bus] BusinessDictionary. *What is a system? definition and meaning.* URL: <http://www.businessdictionary.com/definition/system.html> [cited 04 11 2015].
- [CGP] CGP Grey on YouTube. *How Many Countries Are There?* URL: <https://www.youtube.com/watch?v=4AivEQmfPpk> [cited 26 04 2016].
- [CT95] Christophe Claramunt and Marius Thériault. Managing time in gis: An event-oriented approach. In *Proceedings of the International Workshop on Temporal Databases: Recent Advances in Temporal Databases*, pages 23–42, London, UK, UK, 1995. Springer-Verlag. URL: <http://dl.acm.org/citation.cfm?id=647207.719698>.
- [Dar] Dare Obasanjo. *An Exploration of Object Oriented Database Management Systems.* URL: <https://www.cs.umd.edu/class/spring2002/cmsc434-0101/MUIseum/applications/odms.html> [cited 14 May 2016].
- [Dav] David Scofield, AsiaTimes. *Seoul's double-talk on reunification.* URL: <http://www.atimes.com/atimes/Korea/GA04Dg01.html> [cited 26 04 2016].

- [DISP08] Stanislas Dehaene, Véronique Izard, Elizabeth Spelke, and Pierre Pica. Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigenous Cultures. *Science (New York)*, pages 1217–1220, 2008.
- [Dra] Nicolae Dragulanescu. Knowledge Map of Information Science: Data, Information, Knowledge. URL: <http://www.success.co.il/is/dik.html> [cited 05 11 2015].
- [Esr] Esri. *Polygon*. URL: <http://esri.github.io/geometry-api-java/doc/Polygon.html> [cited 30 Oct 2015].
- [Eth07] Philip J. Ethington. Placing the Past: ‘Groundwork’ for a Spatial Theory of History. *Rethinking History*, 11:465–493, Dec 2007.
- [Eur] European Parliament. *Turkey and the problem of the recognition of Cyprus*. URL: [http://www.europarl.europa.eu/meetdocs/2004\\_2009/documents/nt/553/553930/553930en.pdf](http://www.europarl.europa.eu/meetdocs/2004_2009/documents/nt/553/553930/553930en.pdf) [cited 26 04 2016].
- [Fra] Fraczek, Witold (ESRI). *Mean Sea Level, GPS, and the Geoid*. URL: <http://www.esri.com/news/arcuser/0703/geoid1of3.html> [cited 27 10 2015].
- [Fra98] Andrew U. Frank. Different Types of Times in GIS. *Spatial and Temporal Reasoning in Geographic Information Systems*, pages 40–62, 1998.
- [Geo] Geokov. *Map projection - types and distortion*. URL: <http://geokov.com/education/map-projection.aspx> [cited 27 10 2015].
- [GG14] I.N. Gregory and A. Geddes. *Toward Spatial Humanities: Historical GIS and Spatial History*. The Spatial Humanities. Indiana University Press, 2014.
- [Glo] Global Security. *Georgia mocks Nauru's recognition of Abkhazia*. URL: <http://www.globalsecurity.org/military/library/news/2009/12/mil-091215-rianovost11.htm> [cited 26 04 2016].
- [Gut14] Perparim Gutaj. The foreign policy of counter secession: Preventing the recognition of contested states. *Political Studies Review*, 12(2):278–278, 2014. URL: <http://EconPapers.repec.org/RePEc:bla:pstrev:v:12:y:2014:i:2:p:278-278>.
- [Häg70] Torsten Hägerstrand. What about people in regional science? *Papers of the Regional Science Association*, 24(1):6–21, 1970. URL: <http://dx.doi.org/10.1007/BF01936872>.
- [HW90] Gary J. Hunter and Ian P. Williamson. The development of a historical digital cadastral database. *International journal of geographical information systems*, 4(2):169–179, 1990.
- [Isr] Israel Ministry of Foreign Affairs. *Israel's Diplomatic Missions Abroad*. URL: <http://www.mfa.gov.il/mfa/abouttheminstiry/pages/israel-s%20diplomatic%20missions%20abroad.aspx#note> [cited 26 04 2016].
- [Jan] Jan Werner Müller. *East Germany: Incorporation, Tainted Truth, and the Double Division*. URL: <http://www.oxfordscholarship.com/view/10.1093/0199240906.001.0001/acprof-9780199240906-chapter-9> [cited 18 05 2016].
- [KH08] A.K. Knowles and A. Hillier. *Placing History: How Maps, Spatial Data, and GIS are Changing Historical Scholarship*. ESRI Press, 2008.
- [Kno02] A.K. Knowles. *Past Time, Past Place: GIS for History*. ESRI Press, 2002.
- [Lan88] G. E. Langran. A framework for temporal geographic information systems. *Cartographica*, 25, 1988.

- [Lan89] G. E. Langran. Time in geographic information systems. 1989.
- [Law] Law Teacher. *Declaratory And Constitutive Theories Of State*. URL: <http://www.lawteacher.net/free-law-essays/constitutional-law/declaratory-and-constitutive-theories-of-state.php> [cited 26 04 2016].
- [Leo15] Leon Hadar, AntiWar.com. *In Praise of 'Virtual States'*, 2015. URL: <http://antiwar.com/hadar/?articleid=8042> [cited 26 04 2016].
- [n.A] n.A. *What Does It Mean to be "MECE" at McKinsey?* URL: <http://workingwithmckinsey.blogspot.de/2014/02/MECE-at-McKinsey.html> [cited 18 05 2016].
- [Nak] Frieder Nake. Data, Information and Knowledge. URL: <http://viola.informatik.uni-bremen.de/typo/fileadmin/media/DataInformKnowledge.pdf> [cited 05 11 2015].
- [Ora] Oracle. *1970s — Defying Conventional Wisdom*. URL: <http://www.oracle.com/us/corporate/profit/p27anniv-timeline-151918.pdf> [cited 14 May 2016].
- [OS01] T. Ott and F. Swiaczny. *Time-Integrative Geographic Information Systems: Management and Analysis of Spatio-Temporal Data*. Number v. 1 in Time-integrative Geographic Information Systems: Management and Analysis of Spatio-temporal Data. Springer Berlin Heidelberg, 2001.
- [Pal] Palestinian Central Bureau of Statistics. *Estimated Population in the Palestinian Territory*. URL: [http://www.pcbs.gov.ps/Portals/\\_Rainbow/Documents/gover\\_e.htm](http://www.pcbs.gov.ps/Portals/_Rainbow/Documents/gover_e.htm) [cited 26 04 2016].
- [PD95] Donna Peuquet and Niu Duan. An event-based spatiotemporal data model for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 9(1):7–24, 1995. URL: <http://dblp.uni-trier.de/db/journals/gis/gis9.html#PeuquetD95>.
- [Peo] People of Kosova. *Who Recognizes Kosova?* URL: <http://www.kosovothanksyou.com> [cited 26 04 2016].
- [Peu99] D. J. Peuquet. Time in GIS and geographical databases. *Geographical information systems*, pages 91–102, 1999.
- [PTKT04] Nikos Pelekis, Babis Theodoulidis, Ioannis Kopanakis, and Yannis Theodoridis. Literature review of spatio-temporal database models. *Knowl. Eng. Rev.*, 19(3):235–274, September 2004. URL: <http://dx.doi.org/10.1017/S02698890400013X>, doi:10.1017/S02698890400013X.
- [Raz12] Ale Raza. Working with spatio-temporal data type. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B2:5–10, 2012.
- [Ren96] A Renolen. History Graphs: Conceptual Modeling of Spatio-temporal Data. *GIS Frontiers in Business and Science*, 2, 1996.
- [Rep] Republic of China. *Ministry of Foreign Affairs*. URL: <http://www.mofa.gov.tw/en/default.html> [cited 26 04 2016].
- [RGS] RGS - Royal Geographical Society. *What is geography?* URL: <http://www.rgs.org/geographytoday/what+is+geography.htm> [cited 23 10 2015].
- [Sol14] A.C. Solana. *Spatio-Temporal Narratives: Historical GIS and the Study of Global Trading Networks (1500-1800)*. Cambridge Scholars Publisher, 2014.

- [Tho] Thomson Gale, International Encyclopedia of the Social Sciences (1968). *Enclaves and Ex-claves*. URL: <http://www.encyclopedia.com/doc/1G2-3045000363.html> [cited 02 Jun 2016].
- [Tod] Today.az. *For Azerbaijan Pakistan does not recognize Armenia as a country*. URL: <http://www.today.az/news/politics/30102.html> [cited 26 04 2016].
- [Uni] United Nations. *Member States*. URL: <http://www.un.org/en/member-states/index.html> [cited 26 04 2016].
- [Uni82] United Nations. *United Nations Convention on the Law of the Sea*, 1982. URL: [http://www.un.org/depts/los/convention\\_agreements/texts/unclos/unclos\\_e.pdf](http://www.un.org/depts/los/convention_agreements/texts/unclos/unclos_e.pdf) [cited 28 04 2016].
- [Uot] Uotila, Urho A. (Britannica.com). *geoid — geology*. URL: <http://www.britannica.com/science/geoid> [cited 27 10 2015].
- [Vat] Vatican City State. *Population*. URL: <http://www.vaticanstate.va/content/vaticanstate/en/stato-e-governo/note-generali/popolazione.html> [cited 26 04 2016].
- [Wes] Western Sahara Online. *Proclamation Of The Saharawi Arab Democratic Republic*. URL: <https://web.archive.org/web/20080502035045/http://www.wsahara.net/sadr.html> [cited 26 04 2016].
- [WHM90] MICHAEL F. WORBOYS, HILARY M. HEARNSHAW, and DAVID J. MAGUIRE. Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, 4(4):369–383, 1990.
- [Won] WonderWhy on YouTube. *How many countries are there in the world?* URL: [https://www.youtube.com/watch?v=4vsPB\\_lbo94](https://www.youtube.com/watch?v=4vsPB_lbo94) [cited 26 04 2016].
- [Yal] Yale Law School. *Convention on Rights and Duties of States*. URL: [http://avalon.law.yale.edu/20th\\_century/intam03.asp](http://avalon.law.yale.edu/20th_century/intam03.asp) [cited 26 04 2016].
- [Yua96a] May Yuan. Modeling Semantical, Temporal, and Spatial Information in Geographic information Systems. *Geographic Information Research: Bridging the Atlantic*, pages 334–347, 1996.
- [Yua96b] May Yuan. Temporal gis and spatio-temporal modeling. In *Proceedings of Third International Conference Workshop on Integrating GIS and Environment Modeling, Santa Fe, NM*, 1996.
- [Zha11] Ziliang Zhao. Design and Implementation of an Object-Oriented Space-Time GIS Data Model. *Master's Thesis, University of Tennessee*, 2011. URL: [http://trace.tennessee.edu/utk\\_gradthes/1043/](http://trace.tennessee.edu/utk_gradthes/1043/) [cited 07 11 2015].
- [Zwa] Vladimir Zwass. *information system*. URL: <http://www.britannica.com/topic/information-system> [cited 05 11 2015].