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
Matriculation Number 90487
born on 21. August 1989 in Spremberg, Germany

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

Submission Date: 10. June 2016

Abstract

This is the abstract. That is very very hard to do!

TODO

Contents

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.

I spent the majority of the time of this thesis (October-December 2015 and February-May 2016) at *Scholars' Lab* of *University of Virginia*. I was provided with an excellent workspace, very cooperative people and delicious oatmeal raisin cookies – all of which 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, especially 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.

Since this thesis was an advancement of HistoGlobe, the project I founded in 2010, I want to thank everybody that has ever been involved in this project. In the previous 6 years I learned how to (not) make a project successful that almost everybody likes. Special thanks goes to Frej with 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 did to HistoGlobe and me. Their implementation of the current state of HistoGlobe was the basis of this work. I also want to thank the other developers that 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. I am also thankful for Carola and Katrin who 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 learned from you how valuable good design is.

Thank you, *Bauhaus-Universität Weimar* and all the associated professors and academic staff at this great institution, for the support throughout the previous seven years. The history and the values of the Bauhaus always inspired me and are a big part of how I think, design and create.

Finally, I want to thank my family that has always been there for me thourghout the first 26 years of my life. I studied for seven years, had three longer stays abroad and spent some of my time with entrepreneurial ideas instead of my studies. But nonetheless, my parents, my brother, my sister and the rest of my family always supported me. This feeling is invaluable and I am thankful for it.

Last but not least: Thank you, John Oliver and *Last Week's Tonight*. It was the best possible distraction for the work on my thesis.

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)*

John Lennon's song is an anthem for peace on Earth, not only for brotherhood of people, the end of materialism, but also for the end of countries. He connects the concept of a country to the nationalism that encourages people to fight and die for it. John Lennon wrote "Imagine" in the 1970s, in the midst of the Cold War between the capitalist and the socialist blocs, only 30 years after World War II and 50 years after World War I. Especially in Europe, this time would have probably not been described as "peaceful". And many, 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 largely shared economies. 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 did not exist 200 years ago. While traveling in those days was probably not as pleasant as it is today, Goethe at least did not need a passport when he traveled to Italy and back to Weimar. He also did not travel from the country "Germany" via "Austria" to "Italy", he instead crossed several duchies and principalities that do not exist any more.

1.1 Motivation

What we might call "our country" today has changed a lot over time. Hardly any of the current 193 member states of the United Nations have the same borders as they did 100 years ago. Countries have evolved in time and space. Would it not be desireable to see this development? Would it not be interesting to have a map that shows the state of the world at any point in history? With this map we could see what our country looked like 100, 200 or even 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 they 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*? These is the question at heart of this thesis: How can the historical development of countries be visualized, for the benefit of a better understanding of how we became what we are today.

This is a very ambitious undertaking, given that countries have changed frequently. There are serious conceptual problems: How do we know what a country looked like in 1600? If we find a historical map of this time, can we trust it? How certain can we be that the countries and their borders are correct? The next problem is that we are faced with contradictory histories of countries. There is not always *one story* which is supported by all sides. There are contested territories, even today, whose ownership is unclear. There are "places", even today, which are not clearly a "country" because some might disagree. There is a great deal of uncertainty and disagreement in the body of history which this thesis addresses.

Finally, the state of the world cannot simply be visualized, 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 acquired some data, we would not want to manually write it into a database. The third goal for this thesis is to develop a well-designed user interface to edit the history of countries directly on the map.

TODO: figure new globe vs. old globe

1.2 Problem Domain

> *All human action 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 time stamp including a date, 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. Today, 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 cannot provide an answer to the question of *when* a country was found 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 exist 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: *snapshots* are taken at certain points in time. They are maps that completely show the current state at this moment. Snapshots can immediately answer the question of what the world looked like at certain dates. However, they fail to answer the next question: what has changed since the last snapshot, when and why? Given two historical maps of Germany, one from 1871 after the formation of the German Empire and one from 1919 after the Treaty of Versailles – how did it look like at the beginning of World War I in 1914? As shown in figure 1.1, this is impossible to say, because there is no information about an arbitrary point in time 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 any country on Earth changes.

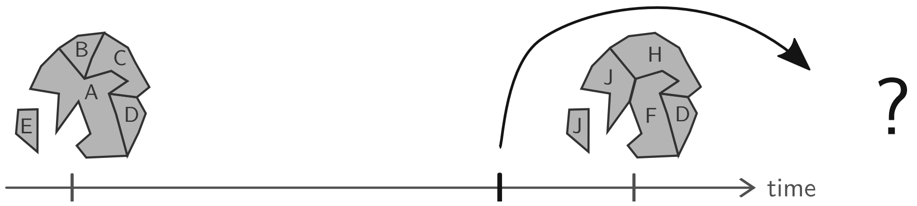


Figure 1.1: The snapshot approach for modeling time and space

The key problem is that snapshots cannot express 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 point in time and trigger changes on the map relative to the last event.

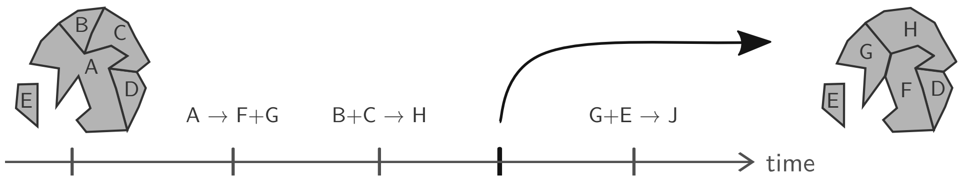


Figure 1.2: The event-based approach for modeling time and space

Figure 1.2 shows such an event-based approach: three changes are consecutively happening to the snapshot on the left. The question about the status of the world at an arbitrary point in time can be answered like this: it is the state of the reference map and each change of all events since that moment accumulated. In case of figure 1.2, country A splits up into F and G, and B and C unify to H. This approach is suitable for modeling a country's 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 a HGIS that deals with the development of countries in time and space. The domain is limited to countries, their names and borders as well as historical events that change them. In addition, the model is to be developed in HistoGlobe, an 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. 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* in section 2.3. Finally, 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 are able to model all possible changes to the development of countries in time and space. The next section 3.2 presents approaches to modify 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 3 are evaluated in section 4.1. Some shortcomings that result from the analysis are addressed in extensions to the original model. They handle some aspects of uncertainty regarding the historical development of countries. The designed extensions are presented in section 4.2.

In the final chapter, the thesis summarizes the work by answering the research questions. The work finishes with an overview about remaining questions, identified problems and possible approaches to solutions that are subject to future work on this topic.

Chapter 2

Basics

According to the title of this Master's Thesis:

Visualizing[1] and _Editing_[2] the _History_[3] of _Countries_[4] in _Time and Space_[5] with _HistoGlobe_[6]

this chapter will form the theoretical foundation and present related work:

[1] The purpose of a *visualization* is to visually 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.

[2] An information system can allow the user to modify, correct or generally *edit* the information in the system.

[3] 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.

[4] A *country* is commonly referred to as 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.

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

[6] The chapter closes in 2.4 with a presentation of *HistoGlobe*, a web-based HGIS that was used to implement the data model developed in this thesis.

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 English 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." [Oxf]. 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 organization that was founded in October 1945. It promotes international peace keeping, security and 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 an 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 by far the smallest sovereign state in the world (0.44 km^2), an enclave inside the city of Rome with a population of only 800 people [Vat].

The *State of Palestine* with a population of 4.8 million people as of 2016 [Pal] 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 [Sel]. Moreover, while 114 states officially recognize the Palestinian state, almost all current main economic powers do not, including Germany and the United States. Unlike the Holy See, Palestine is not a fully sovereign and recognized country.

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 Kosovo's membership. Therefore, Kosovo is not even a UN observer state, although it has 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 of 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: mainland China and the island of Taiway. Since 1971 the People's Republic of China is the only representative of whole China in the United Nations, including 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 their own jurisdiction, issue their own passports and have unofficial diplomatic relations to most countries in the world. Only 22 UN members officially uphold diplomatic relations to Taiwan [Rep]. To all of these states the People's Republic of China does not kepp 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) [BBCb], the Turkish Republic of Northern Cyprus (1) [Had15], Nagorno-Karabakh Republic (0) [BBCa], Transnistria (0) [Gut14] and Somaliland (0) [BBCc].

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 [Sco] 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 with the 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]

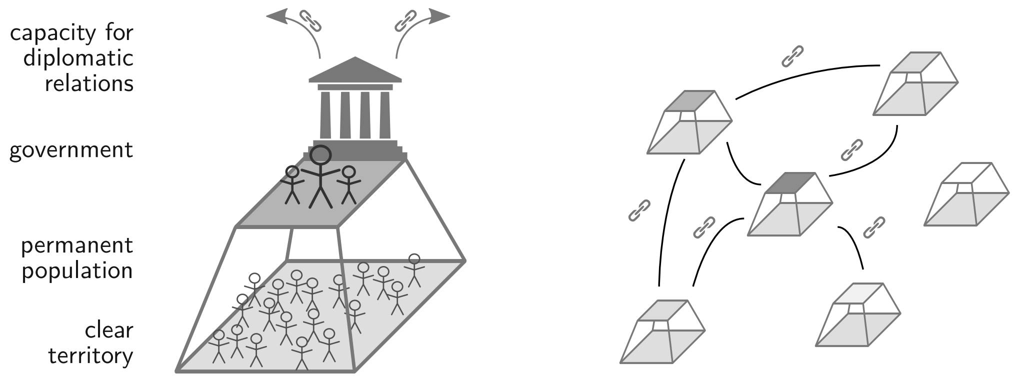


Figure 2.1: Declaratory Theory (left) vs. Constitutive Theory (right) of statehood (based on [Law])

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 one did, there were people who disagree. This is a big problem for the intended information system, because it is impossible to objectively classify a place as a country or not. Moreover, this section only covers the current countries. 100 years ago these two theories did not exist. It is thus impossible to make a conflict-free decision of what constitutes a country at any point before the 20th century. Moreover, it is 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 will disagree with. Its data model cannot perfectly deal with self-classifying data and cannot rely on an objective data source. The system has to contain approaches that deal with these problems.

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 on the territory, since it can be easily presented on a 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 mapping 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 country *borders*. A border can be modeled by 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. According to the United Nations, the territory of a country extends in a range of 5 to 20 kilometers into international waters [Uni82]. This forms the *sea territory* of a county. An interior border can be either *natural* if it is based on geographical features, e.g. a river or a mountain ridge, or *artificial*. 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 separated 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 country, is not an enclave. Enclaves can be nested and form a hierarchy, e.g. a *second-order enclave* is an enclave within an enclave [Gal68].

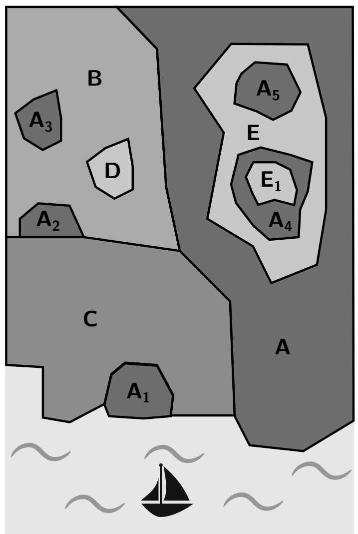


Figure 2.2: Enclaves and exclaves based on [Gaz]

In the example in figure 2.2 A has three exclaves: A_1, A_2 and A_3. Only A_3 is also an enclave, within B. E is an enclave of A. Within E, there are two second-order enclaves for A, A_4 and A_5. Inside A_4, there is E_1, a third-order enclave from the perspective of A and a second-order enclave for E. D is the only fully enclaved territory.

Second-order enclaves infrequently appear in the real world, as in the example of Baarle-Nassau and Baarle-Hertog at the border between the Netherlands and Belgium [Kau12]. There are ONLY three countries in the world that are complete enclaves: San Marino, Vatican City and Lesotho.

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 from the outside [Bus]. An *information system* is an application that acquires, manages, analyses and presents 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].

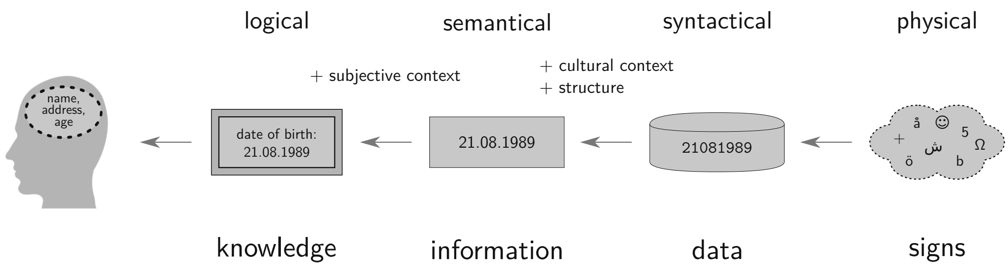


Figure 2.3: Signs, data, information and knowledge

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 the set of signs is uncountably infinite. *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 same data might just be a random string of numbers – it does not have any information. When a human understands the visualization of information he or she can integrate it into a 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 called a *Geographic Information System* (GIS). The data objects in the system are called *geo-objects* [Bol08]. If the information has an additional temporal dimension, e.g. via time stamps or time spans, which allow to trace developments of geo-objects, it becomes a *Historical Geographic Information System* [GG14] or alternatively *Spatio-Temporal Information System* (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].

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 Like many other fields in the humanities, *history* is "an ideal field for thinking long and hard about important questions" [AB]. It originates from the Greek word *Ιστορια / historia*: "finding out, learning through research, narration of what is learned" [Ran15] 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" [AB]:

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. Therefor 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 historical events or processs based on evidence. The problem is that history cannot run experiments, because the same conditions cannot be repeated. Therefor 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. This problem is comparable to Heisenbergs uncertainty principle in physics: Physical movements on the macro-level are a direct cause of a set of preconditions, e.g. speed, fraction, wind or weight. They are therefore predictable. However, the smallest of all particles are not traceable, their movements are not predictable and therefore their processes not entirely explainable.

Geography (Greek $\gamma\varepsilon\omega\gamma\rho\alpha\varphi\iota\alpha$/ *geographia*) literally means "describing the earth" [Ran15]. It is a science that studies the interplay between the landscapes and environments of the Earth (*physical geography*) on the one hand and people, their cultures, societies and economies (*human geography*) on the other. It is an interdisciplinary field between natural and social sciences [Roy]. 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" [Tob70].

Geographers use different techniques to answer their research questions. One important tool is a *map*: a two-dimensional representation of physical, environmental, political, economical or social properties of the Earth. The "art and science of making maps" is the field of *cartography* [Abe08].

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].

Geography	*Aspect*	*History*
Where?	Question	When?
Space	Dimension	Time
Exact, statistical	Character	Fuzzy, complex
Mainly quantitative	Research	Mainly qualitative
Spatial proximity of conditions	Causal explanation	Temporal sequence of events
Clustering	Organization principle	Periodization
Mostly visual (maps)	Form of expression	Mostly verbal (texts)
High (GIS)	Digitalization potential	Low (digital humanities)

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

2.2.2 Geospatial Data

A HGIS deals with the temporal development of geo-objects. The geo-objects in this case are the territories of countries. They are usually presented on a *map*. The problem is that the Earth in the real world is three-dimensional and infinitely in detail, but the map is two-dimensional and can only show discrete features. In order to show a country on a map, there are mainly three steps involved:

1. Choose a three-dimensional reference surface, an *ellipsoid* or a *spheroid*, that approximates the real shape of the Earth reasonably well.

2. Define the *geographic coordinate system* to locate each geo-object in using a *geodetic datum*.
3. Use a *map projection* to show the curved surface of the Earth on a flat two-dimensional map.

Reference surface The shape of the Earth is very complex. In the Babylonian Empire (≈ 2000-539 B.C.E) the theory of the Earth being a flat disc surrounded by an infinite body of water evolved. The Greek scientists Pythagoras and Aristotle rejected this theory around 340 B.C.E and proved the Earth to be a three-dimensional *spherical* object. It took almost 2000 years until Sir Isaac Newton reasoned in 1687 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*. However, the model disregards that the surface of the Earth is not smooth 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 uniform sea level of the ellipsoid model. These discoveries in the 20th century led to the *geoid*, a physical model of the Earth. The latest and most accurate measurements are the result of the GOCE satellite launched in March 2009 [Uot, Fra]. However, the geoid is too complex to work with and hence, usually a reference *ellipsoid* is used to approximate the geoid. The ellipsoid is a reference surface with two radii: the polar radius (r_p) and the slightly larger equatorial radius (r_e) [Bol08, pp. 69-77].

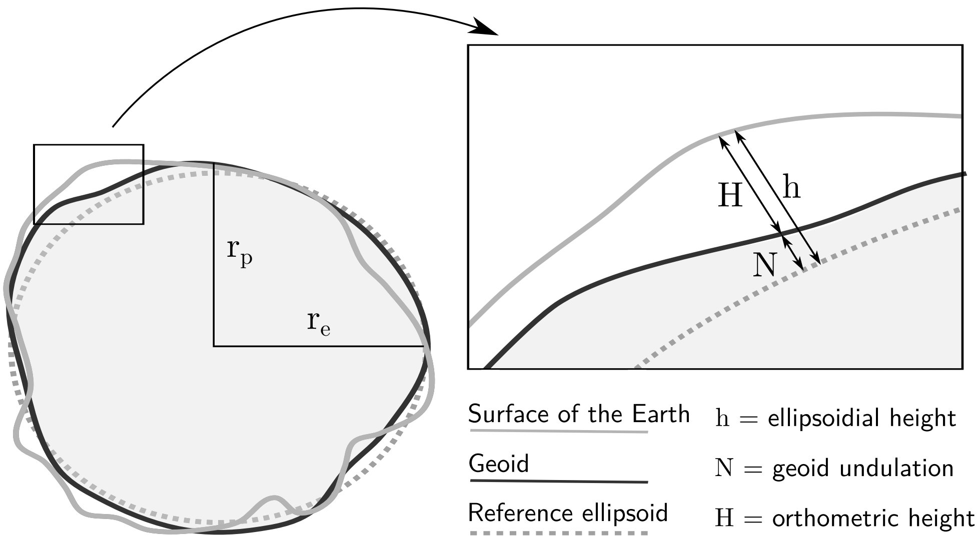


Figure 2.4: Models of the Earth's figure, differences are exaggerated [Bol08, Fig. 3-6, p. 75]

Geographic coordinate system The reference ellipsoid is expressed in a three-dimensional 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: $\gamma \in [-180° ... +180°]$
2. The rotation angle along the Prime Meridian, defining its latitude: $\phi \in =[-90° ... +90°]$
3. The distance to the origin: $r > 0$

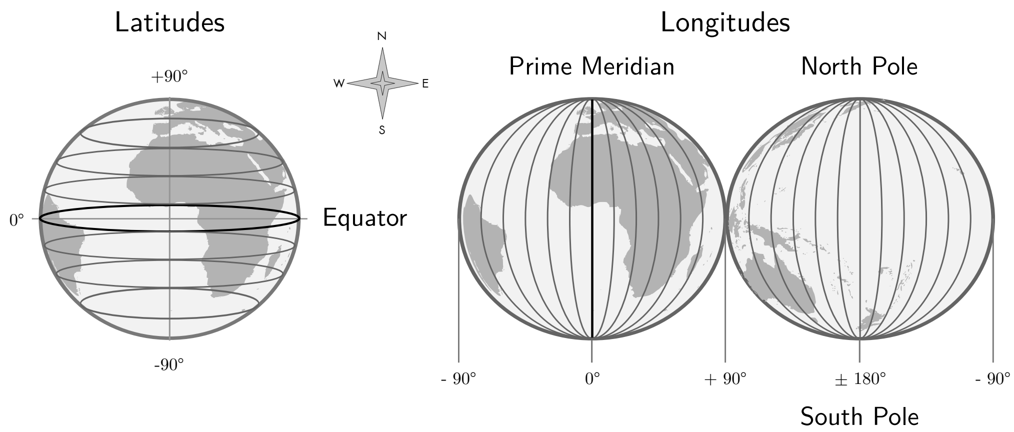
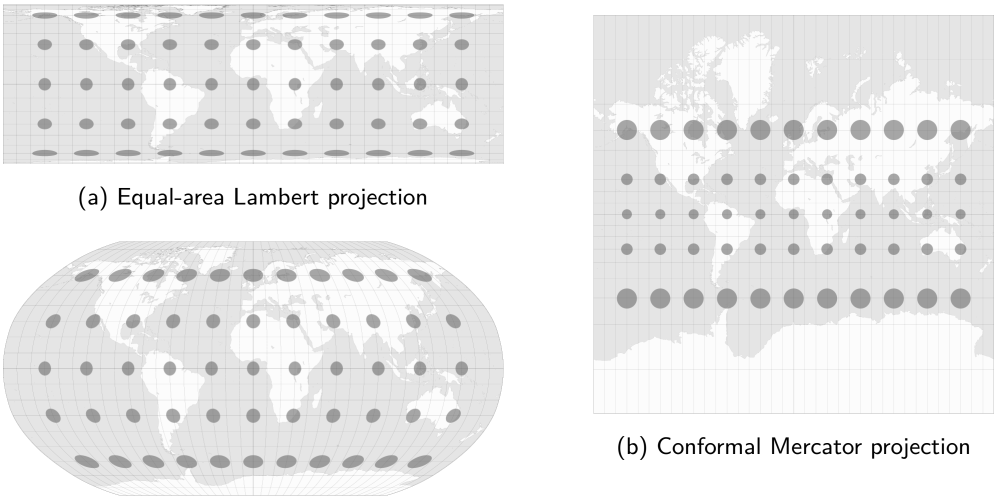


Figure 2.5: Geographic coordinates using latitude and longitude

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° latitude). All meridians have the same length. Geographic coordinates are usually recorded either in degree-minutes-second (`DMS`, e.g. `50° 58′ 22′′`) or in decimal degree (`DD`, e.g. `50.973`) notation [Bol08, pp. 30, 79].

The geometric ellipsoid aims to fit the physical geoid as good as possible. The *geodetic datum* is a geographic coordinate system, usually based on an ellipsoid, that defined a set of reference points that relate to the geoid. There are a lot of different geodetic datums, because they can be very accurate in one region of the world, but inaccurate in another. The same geographic coordinates in two different geodetic datums define two different points on Earth. Therefore, it 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.

Map projections In the last step, the surface of the three-dimensional ellipsoid has to be projected onto a two-dimensional map that can be printed on paper or visualized on a computer screen. Because the surface of the ellipsoid is curved, some features of the Earth will be distorted on the map: An *equal-area 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 presents all features of the real world correctly [Geo]. A compromise between preserving areas and shapes is the *Robinson projection*: it is neither conformal, nor equal-area, but provides a reasonable trade-off between both properties.



(a) Equal-area Lambert projection

(b) Conformal Mercator projection

(c) Compromise Robinson projection

Figure 2.6: Comparison between different map projections, based on [Gab08]

Digital maps A map in a HGIS is structured according to the *layer* principle: Each layer is a transparent film showing one specific aspect, e.g. a physical layer for landmasses or water and a political layer for international borders and names of countries. The layers are interchangeable and can be shown and hidden. A *legend* including the scale bar and north arrow should explain all symbols used on the map and give orientation. Interactive maps may have additional 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 interactive map services is *Leaflet.js*, “an open-source JavaScript library for mobile-friendly interactive maps” [1]. It is used in the implementation of this this thesis. A leaflet map is embedded in a HTML document on the client-side of a web-based information system. Additionally, the user can include own map layers, symbols and markers on that map. Leaflet uses a conformal Mercator projection, comparable to figure 2.6b, also known as *Web Mercator*.

[1] *Leaflet - JavaScript library for interactive maps*, URL: `http://leafletjs.com/`, accessed on: 02.11.2015

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 should be 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.

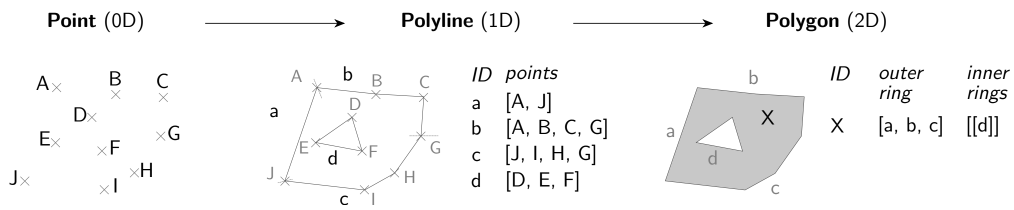


Figure 2.7: The basic geometric primitives point, polyline and polygon

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 an ordered set of at least two points. The outermost ones are the start and 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*. If it is even self-intersecting, as shown in figure 2.8, the polygon is *complex*. *Polypolygons* represent multiple separate polygons belonging to one logical entity [Esr].

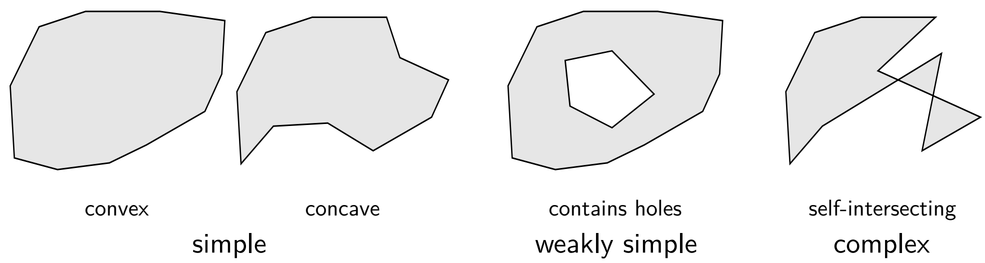


Figure 2.8: Types of polygons

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 2.9. If multiple polygons are unified at once, it is a *cascaded union* [Bol08].

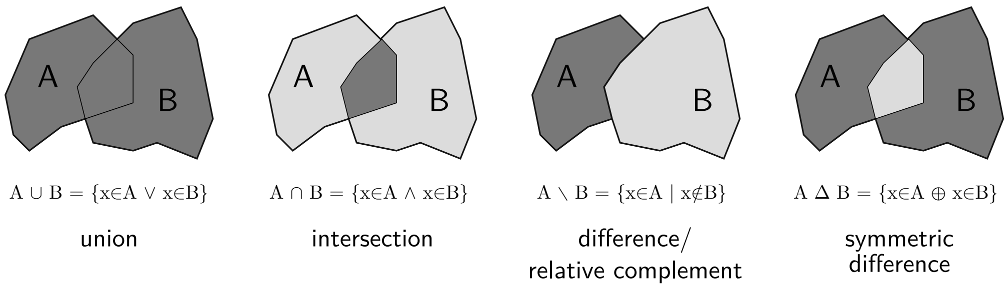


Figure 2.9: Boolean set operations on polygons, the dark gray area is the result of the operation

Common file types for geospatial vector data are the open format GeoJSON [2] or ESRI Shapefiles [3].

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 a 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. Relative to one point, time has two directions: historically *forward* into the future and *backward* into the past. The topological relationship between two points in time t_1 and t_2 is straightforward, because there are only three different order relations: $t_1 < t_2$, $t_1 > t_2$ and $t_1 = t_2$.

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

Timelines In contrast to space, time does not have an intrinsic representation. However, the most common form of visualizing 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 points in time or processes as time intervals in chronological order. A timeline shows additional time markers on a certain date

[2] *GeoJSON*, IETF GeoJSON Working Group, URL: `http://geojson.org/`, accessed on: 30.10.2015

[3] *ESRI Shapefile Technical Description*, ESRI White Paper, July 1998, URL: `http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf`, accessed on: 30.10.2015

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 points in time 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 center. Similar to linear timelines, the farther away a point is from the reference point, the farther away it is positioned on the timeline. However, the distance between this point and the reference point does not increase linearly – events that are further away appear less far. This time scale accounts for logarithmic human perception: Given two events A and B, whereas A happened 20 years ago and B ten years ago. The perceived temporal distance between A and B is smaller than the distance between B and today, although the absolute distance is the same [DISP08].
- A the scale of a timeline can also be *irregular*, e.g. to get the same distance between events on the timeline. This is particularly useful if the events are not distributed homogeneously.

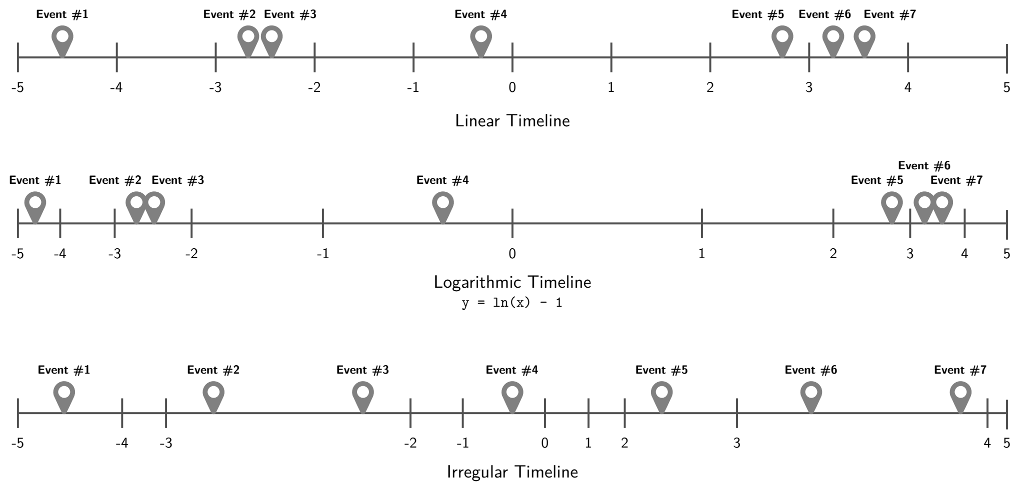


Figure 2.10: Comparison between a linear, a logarithmic and an irregular timeline

2.2.4 Existing Applications

"Today, operational temporal GIS does not exist" [Raz12, p. 5]. 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 can be found in [KH08] and [GG14]. One example related to the topic of this thesis is the *Great Britain*

Historical GIS Project (GBHGIS) [4]. It maps statistical data on historical territorial units of the United Kingdom (UK) using *aerial interpolation* [Arc]. One purpose is for example to analyze net migration in the districts of 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 [Bri]. The *National Historical Geographic Information System* (NHGIS) [5] 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 cumbersome 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 requires historical data about countries, their names, borders, historical events and their introduced 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 the public domain is hosted by Natural Earth [6]. There is physical data, e.g. coastlines and rivers, and cultural data, e.g. political borders and cities available. 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 data is *Wikipedia* and their article categories, e.g. `armistices` or `treaties` [7]. All sorts of historical events can be found. Some information is structured in information boxes, e.g. some historical treaties contain a name, location, a signature and an effect date. Particularly interesting for this thesis are articles about historical countries [8], because they contain the name and important meta information, e.g. their historical successors and predecessors.

The creation of an open-source HGIS 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

[4] *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/`, accessed on: 02.11.2015

[5] *Welcome to NHGIS*, Minnesota Population Center, University of Minnesota, since 2007, URL: `https://www.nhgis.org/`, accessed on: 02.11.2015

[6] *Natural Earth*, URL: `http://www.naturalearthdata.com/downloads/`, accessed on: 30.10.2015

[7] *Category:Treaties*, Wikipedia, the free encyclopedia,
URL: `https://en.wikipedia.org/wiki/Category:Treaties`, accessed on: 13.05.2016

[8] *List of former sovereign states*, Wikipedia, the free encyclopedia, URL: `https://en.wikipedia.org/wiki/List_of_former_sovereign_states`, accessed on: 13.05.2016

and events that are required 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. Further consideration must be given to accuracy and quality of information in Wikipedia, due to their open source nature. Overall, using Wikipedia as a data source for this thesis is not feasible, but this 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 the extraction of a border from a historical map. They can be found on Wikipedia as well, or in historical map collections, e.g. *OldMapsOnline* [9]. The project is developed "out of a love of history and heritage of old maps" and stores about 400 000 historical maps. There are five steps to retrieve the points of a border in geographic coordinates from a historical map. This process was developed in a preceding *HiBo – Historical Borders* project [WDSK15].

1. **Digitization**: If the map is on paper, it has to be scanned in the best possible quality. The result is a raster graphic.

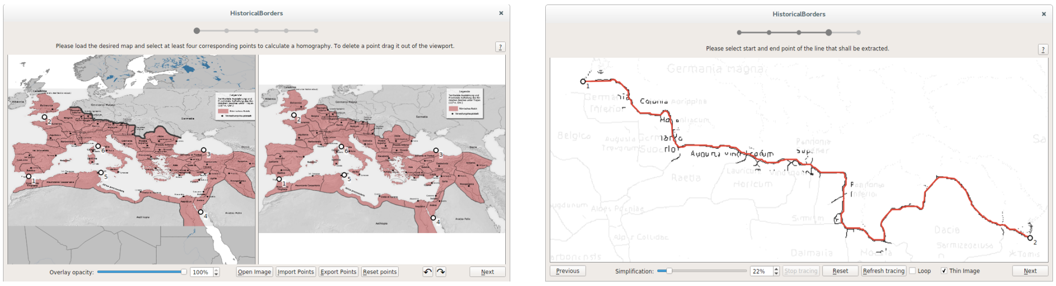
2. **Georeferencing**: The historical map has to fit as good as possible on the reference map. This requires a manual definition of a set of reference points which are used to transform the map into the geographic coordinate system. This process is error-prone, especially if the projection of the historical map is not known and the map itself is not accurate [Kno02]. The outcome is a rectified and resampled 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 Canny Edge Detector to find edges in a raster image [Can86]. 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. This can further be used as a border of a historic country.

[9] *Old Maps Online*, URL: `http://www.oldmapsonline.org/`, accessed on: 13.05.2016



(a) Georeferencing

(b) Semi-automatic digitizing

Figure 2.11: Semi-automatic extraction of a border from a map of the Roman Empire [WDSK15]

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). Each 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 in time, can undergo several changes and can disappear at some other 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 the 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 each of them is the concept of *Time Geography* [Häg70]: There is an orthogonal relationship between time and space. At each point in time 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. In contrast, *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 already introduced in section 1.2, the *Snapshot model* stores the full state of all geo-objects at certain points in time 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 [Lan88]

For all other 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 cannot be solved. Therefore, it is 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, improvements were made to tackle this problem, e.g. [Arm92].

2.3.2 Simple Time-Stamping

The simplest approach to trace the history of a geo-object is to assign a period of existence 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 is deleted. If an object still exists it gets a special value, e.g. `NOW`, as its end date [HW90].

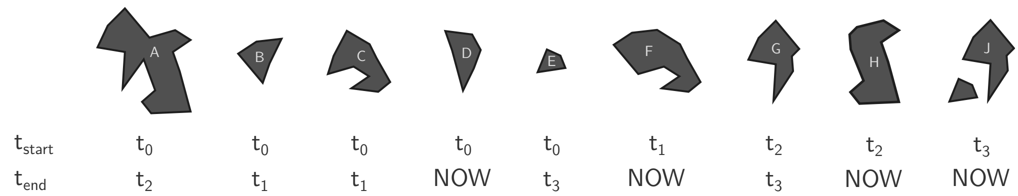


Figure 2.13: The Simple Time-Stamping method [HW90]

The *Simple Time-Stamping* method is also location-based and tracks discrete changes of objects. Given full and consistent information, the state of the system at an arbitrary point t_i can be retrieved: each geo-object for which $t_{start} \leq t_i < t_{end}$ is 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 t_1 B and C end and F starts. Visually, F is a successor of B and C, but this historical relationship cannot 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 point in time. Given the example above, it is unclear if two objects unified to a new one ($B + C \rightarrow F$) or if two are successors ($B \rightarrow F$) and one just stops to exist ($C \rightarrow \emptyset$). The model is also redundant: if a geo-object replaces another one ($B \rightarrow F$), then t_{end} of B is the same as the t_{start} of F [Sol14, p. 46-47].

2.3.3 Event-Based Spatio-Temporal Data Model

A time-based approach addresses exactly these 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 [PD95]. At one defined point in time 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 stamp 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 and not to the base map, change tracking is efficient.

The concept of the ESTDM suits the problem domain really well. A 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].

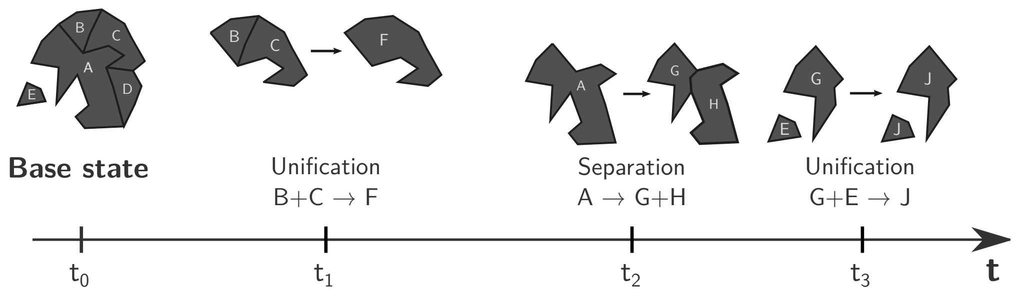


Figure 2.14: An example for an event-based spatio-temporal data model for vector data

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 the geospatial object in vector format representing the country.
- The *temporal domain* stores all temporal objects, e.g. time stamps of a historical event.

The model is not specific, but rather 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 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 non-spatial 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 behavior 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 behavior 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 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 also has an associated interval with a duration of zero for a sudden change. Additionally, a transition links the relevant objects to each other to establish a historical predecessor-successor-relationship.
3. An object that is currently not active, is *ceased* and not visible on the map.

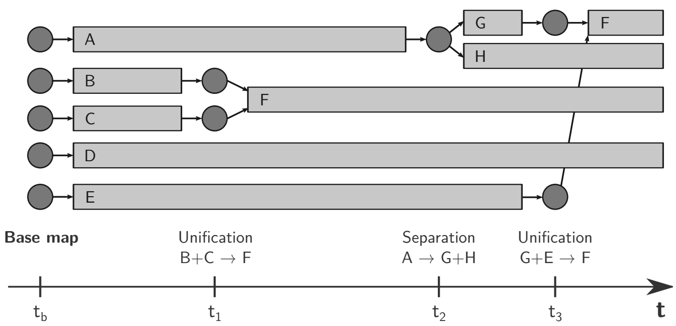


Figure 2.15: The History Graph model

The history of a geo-object is a chronologically ordered set of versions and transitions, that can be visualized in a graph like the one in figure 2.15. The model defines six basic types of temporal changes that can happen. They can be seen in figure 2.16:

- **Creation**: A new object is created.
- **Alteration**: A property of an object, e.g. its geometry, changes.
- **Cessation**: An object ceases to exist.
- **Reincarnation**: An object that has previously been deleted is recreated.
- **Split/Deduction**: An object is divided into multiple new objects, or at least one new object is deducted from an existing one.
- **Merge/Annexation**: Multiple objects are unified to a new object, or at least one object is annexed to another object.

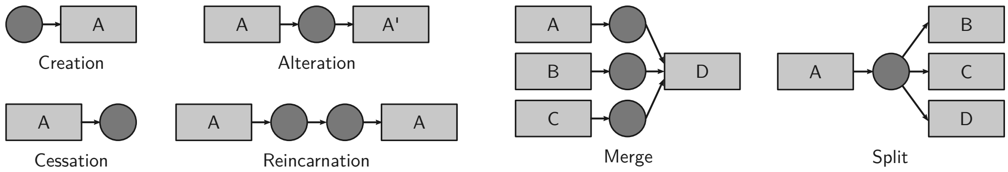


Figure 2.16: Types of changes in the History Graph model

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 be directly 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, including 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*, usually 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 capital and vice versa.
- `1:n` One-to-many relation, e.g. one country can have many cities, but each of those cities 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" [10].

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 [Oba]. The following and many more object-oriented principles are supported:

- *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. a country with its territory. This relates to a tuple in RDBMS.
  - The *attributes* of an object cannot 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 cannot 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`, but it does not need a `head_of_state`.

Object-relational DBMS ORDBMS combine the advantages of both worlds. Internally, the established and efficient relational database is used 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

[10] *MySQL :: About MySQL*, URL: `https://www.mysql.com/about/`, accessed on: 31.10.2015

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" [11].

Spatio-temporal database models The spatio-temporal data models discussed in section 2.3 need to be implemented in a relational, object-oriented or object-relational database management system. While the details depend 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 at which something happened 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 bi-temporal 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 bi-temporal 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 cannot 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 deletes 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, peoples migrate on the planet or kingdoms give way for democracies.*
>
> *This is HistoGlobe, a revolution of teaching and learning of history!*

[11] *PostgreSQL:*, The world's most advanced open source database, URL: `http://www.postgresql.org/`, accessed on: 31.10.2015

HistoGlobe is a web-based HGIS 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 evolved since then 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 on `http://histoglobe.com`.

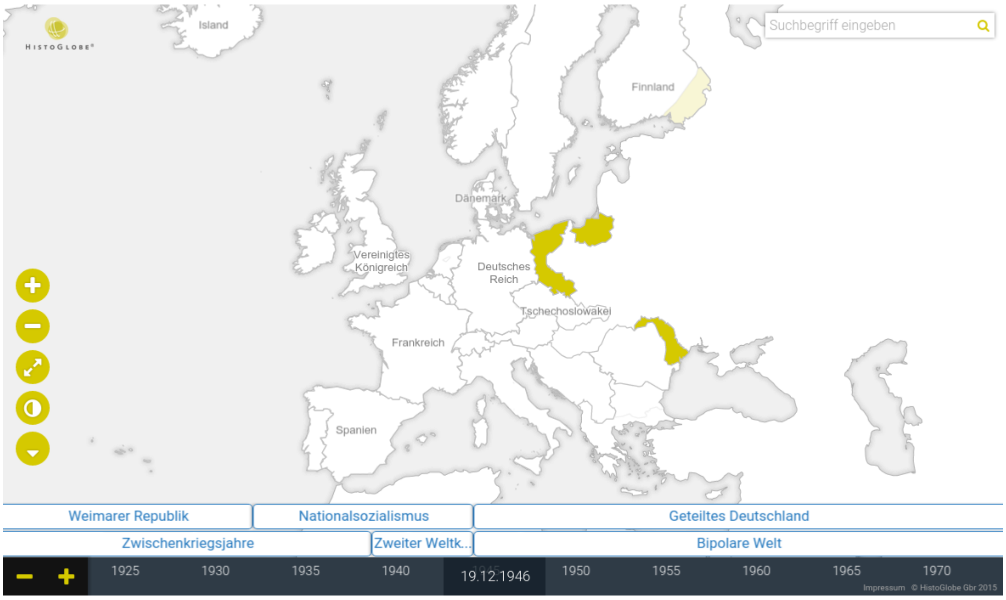


Figure 2.17: The HistoGlobe school project (April 2015)

Figure 2.17 shows the user interface of HistoGlobe: a 2D map, a timeline and control buttons for both. Addtionally, a topic bar extends 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: editing historical information about countries and events is very cumbersome due to a complex domain and a lack of a back-end. Therefore, there is very little data in the system.

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 discussed. 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 not a trivial undertaking, because both the reality and the human using the system are complex. For such 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 into 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 attempts to solve the problem. Each iteration has four steps: The requirements for the system are analyzed in the *planning* step (P). Afterwards, they are translated into an abstract *design* (D) which is realized in a specific *implementation* (I) of the prototype. Finally, this prototype is tested with humans to find out how well it works. Based on the results of this *testing* (T) 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.

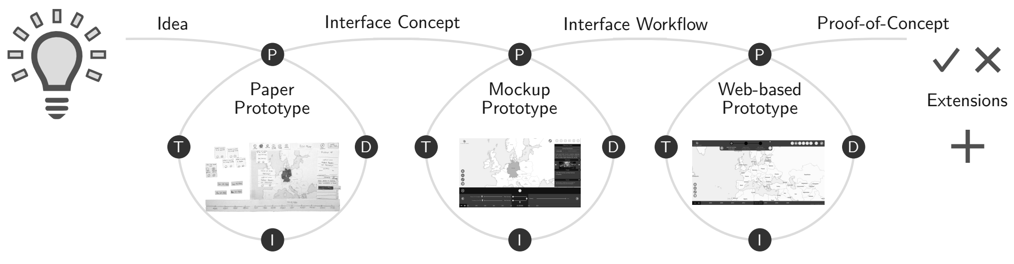


Figure 3.1: Human-centered design process with five project phases

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 specific 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. Every aspect in this chapter is based on the assumption of full certainty about the data: for each moment in history, there is a clear and undisputed state of the world. The data model and the application presented in this chapter is open for extension to tackle the actual problem of uncertainty in history. This is the topic of chapter 4 of this thesis.

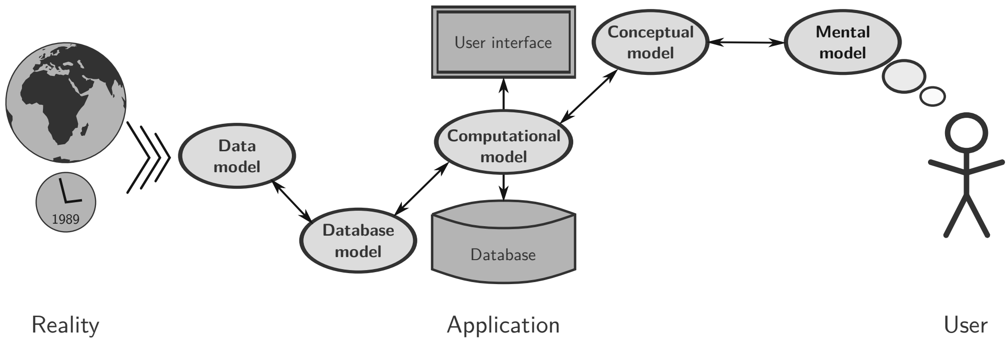


Figure 3.2: Relevant models for an information system

Several models are 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 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 iterative computer systems, the *mental model* is a representation of how the interface should work in the mind of the human – 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 section 3.4 of this chapter.

3.1 Hivent Model

This section proposes the spatio-temporal *Hivent model* to represent the history of countries in time and space. *Hivent* is an acronym for ***Hi***storical e***vent*** and is the main element of the data model. 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 required. For that purpose, the *ESTDM* was developed, but since it only works for raster data, it does not match 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. Section 3.1.1 introduces the main elements of the Hivent model. Afterwards, the basic axioms and assumptions are defined in section 3.1.2. A 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. Section 3.1.4 presents the *HistoGraph*, a non-spatial visualization of historical developments.

3.1.1 Elements

Hivents 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 abstract entities 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 (see section 2.2.2). 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 may be deleted. 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 point in time it is historically existing, otherwise it is *inactive*.

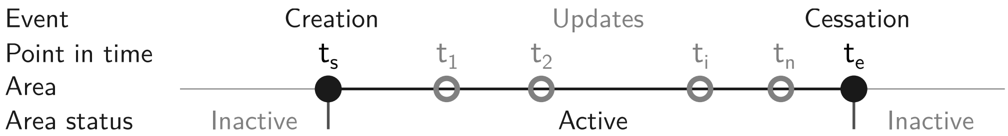


Figure 3.3: Three event types that change Areas, resulting in two different area states

Historical changes and Areas are mutually linked, i.e. an Area keeps references to the changes creating, updating and deleting it and a historical change stores a set old Areas that are deleted, 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 modeled independently from each other:

- The *semantic domain* holding the Area uniquely identifiable is represented by its formal name.
- The *spatial domain* is expressed by 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 with the Area.

The main difference between the spatial and the thematic domain is that there is no relation between the names of two Areas. While the intention is questionable, 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 onto 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 is initialized at t_0. At this initial state, exactly one Area exists. It is denoted by Ω and is referred to as the* universe *Area. It has no name and its territory covers the whole surface of the Earth.*

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

Axiom 5 *At each 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 defined in section 3.1.1, a historical change can create, manipulate and delete 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 secede it.

Formally, it can be said that each change consists of three sets of Areas: the old Areas A that are deleted, 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 must maintain the spatial integrity of axiom 5. Therefore, the total size of the territories before the change and after the change must be the same:

$$\left|\bigcup_{i=1}^{n} A_i^T\right| + \left|\bigcup_{i=1}^{n} C_i^{OT}\right| = \left|\bigcup_{i=1}^{n} C_i^{NT}\right| + \left|\bigcup_{i=1}^{n} B_i^T\right|$$

This behavior is based on to the law of conversation of energy in physics: energy in a closed system can only be transformed from one object to another, but the total energy in the system is preserved at any time. In the spatio-temporal system of this HGIS, the total size of the territory of the Earth is preserved, it is distributed among the active Areas in the system.

The first changes introduced in the system at 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 Ω. The result is that after t_0, the map is divided into water (W) and land (Ω). 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 *unclaimed*, i.e. belonging to Ω.

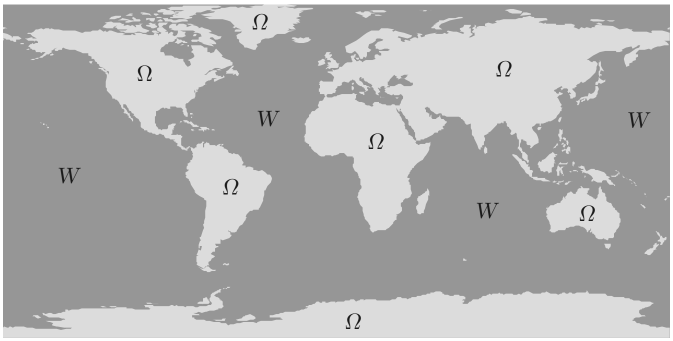


Figure 3.4: The initial state of the world map at t_0

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 geographical processes, e.g. the sea level rise influencing the coastlines, or according to a historical event, e.g. a treaty shifting a border between two countries. 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 coastlines have not changed over time.*

Both assumptions are 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 the coastlines, mainly the continental drifts of tectonic plates. 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 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 are deleted, 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 [Mü].

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 05.05.2016: 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.

UNI	INC	SEP	SEC	NCH
Unification	Incorporation	Separation	Secession	Name Change

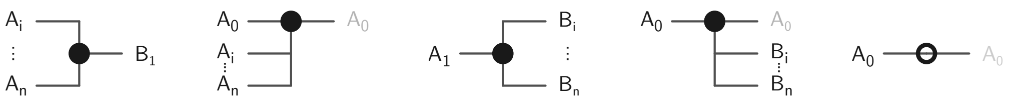


Table 3.1: The five Hivent operations

3.1.4 HistoGraph

Based on the idea of the History Graph model, introduced in section 2.15, 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. As previously explained in section 3.1.1, an Area keeps references to the historical changes creating, updating and ceasing it. These changes themselves are linked to their old and new Areas. Therefore, each Area indirectly knows their successors and predecessors.

The two-dimensional HistoGraph has a horizontal orientation. The x-axis refers to one point in time, 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 exactly to one Area. If a 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 a duration of zero. 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 only the predecessors on the graph, but not the successors of the predecessors. 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.

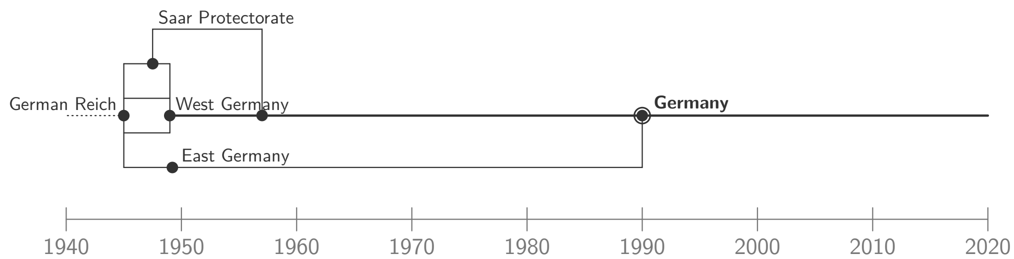


Figure 3.5: The concept of the HistoGraph at the example of Germany's history since 1945

This 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` cannot 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 and becomes the only historical successor of the old Area.

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 federal state "Saarland" in West Germany.	`INC`
03.10.1990	In the German Reunification, East joins West Germany. The Federal Republic of Germany is now 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, which are visualized in the next step. Each of the four occupation zones originated themselves from the German Reich. However, the German 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 cannot be shown in the graph, due to lack of space. One more important aspect can be seen in the foundation 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 had the same identity as 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 beyond the scope of this thesis and subject to future work in the field of Information Visualization.

3.2 Edit Hivent Data

The previous section proposed the abstract Hivent model, a set of Hivent operations and the HistoGraph 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 beneficial, 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. A main goal of this thesis is to develop a user interface that humans can understand and hence, a second set of six high-level *edit operations* is proposed:

	`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.
	`SPL`	Split	. . . 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.
A B	`REN`	Rename	. . . an Area and set a new formal name, short name or both.
	`DEL`	Delete	. . . an Area 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 the correction of information in an event-based system works. Given an Area A at 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 that they can be internally expressed by a set of Hivent operations. The creation of an edit operation happens in four steps:

1. Select the Areas that will be changed in the operation.
2. Create a territory for each new Area resulting from the operation.
3. Create a name for each new Area.
4. Add the edit operation to a Hivent to inherit the time stamp.

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.

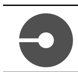
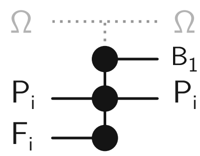
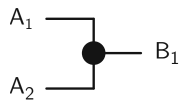
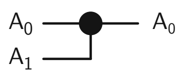
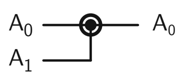
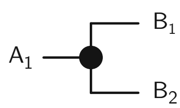
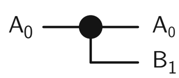
	Select old Areas	*Create new territories*	*Create new names*
	–	Create a territory of the new country	Create a name for the new country
	Select the countries to be merged	– *Territories of selected countries are automatically unified*	Create a name for the new country
	Select a country to be split	Create a territory for each new country	Create a name for each new country
	Select two neighboring countries to change their border	Create a new border between both countries *The territories of both countries are created automatically*	–
	Select a country to rename it	–	Create a new name of the country
	Select a country to delete it	–	–

Table 3.4: The requirements of each step for the edit operations

There are different possibilities to express an edit operation by a set of Hivent operations, depending on the user input in the workflow. All possibilities are introduced in table 3.5. 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`).

Edit operation	Old Areas	Update Areas	New Areas	Expression by Hivent operations [1]	Visualization
`CRE` (1)	Area B_1 is created with territory T. The part of T that is on previously unclaimed land (T_Ω) is seceded as B_1 from Ω. If T_Ω is empty, then B_1 is initialized with an empty territory. The rest of T covers some Areas $P_i \in P$ partially and some Areas $F_i \in F$ fully. $\forall P_i$: The covered territory T_i is seceded and incorporated into B_1. $\forall F_i : F_i$ is completely incorporated into B_1.				
	$\forall F_i$	$\forall P_i$	B_1	`SEC` of B_1 from Ω `SEC` of T_i from P_i, `INC` of T_i into B_1 `INC` of F_i into B_1	
`MRG` (1)	Multiple Areas $A_i \in A, \|A\| \geq 2$ are unified to B_1. The new Area receives a name distinct from all names of A_i.				
	$\forall A_i$	–	B_1	`UNI` of $\forall A_i$ to B_1	
`MRG` (2)	Multiple Areas $A_i \in A, \|A\| \geq 2$ are unified. The resulting Area reuses the short and formal name of one of the old Areas (A_0) and therefore preserves it. The remaining Areas A_i are incorporated into A_0.				
	$\forall A_i$	A_0	–	`INC` of $\forall A_i$ into A_0	
`MRG` (3)	The same as the previous case, but A_0 receives a new short name and therefore an additional name change is required.				
	$\forall A_i$	A_o	–	`INC` of $\forall A_i$ into A_0 `NCH` of A_0	
`SPL` (1)	Multiple Areas B_i are separated from one initial Area A_1. Each B_i receives a part of the territory of A_1 and a name. Each name is distinct from the name of A_1.				
	A_1	–	$\forall B_i$	`SEP` of A_1 into $\forall B_i$	
`SPL` (2)	Multiple Areas B_i are separated from one initial Area A_0. Each B_i receives a part of the territory of A_0 and a name. One of the separated Areas has the same short and formal name as A_0, so it preserves its identity. The remaining new Areas secede from A_0.				
	–	A_0	$\forall B_i$	`SEC` of $\forall B_i$ from A_0	

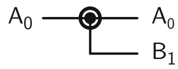
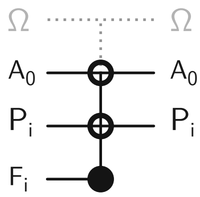
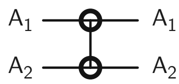
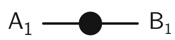
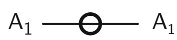
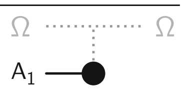
Edit operation	Old Areas	Update Areas	New Areas	Expression by Hivent operations [1]	Visualization
SPL (3)	The same as the previous case, but A_0 receives a new short name and therefore an additional name change is required.				
	–	A_0	$\forall B_i$	SEC of $\forall B_i$ from A_0 NCH of A_0	
CHB (1)	One 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 withdraw (T_w). The part of T_e that expands into unclaimed land ($T_\Omega : T_\Omega \in T_e$) is seceded from Ω and incorporated into A_0. The Areas F_i which are 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 incorporated into Ω, resulting in unclaimed land.				
	$\forall F_i$	$A_0, \forall P_i$	–	SEC of T_Ω from Ω, INC of T_Ω into A_0 SEC of T_i from P_i, INC of T_i into A_0 INC of F_i into A_0 SEC of T_w from A_0, INC of T_w into Ω	
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	–	SEC of T_2 from A_1, INC of T_2 into A_2 SEC of T_1 from A_2, INC of T_1 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	–	B_1	UNI of A_1 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	–	NCH of A_1	
DEL (1)	One Area A_1 is selected and deleted by incorporating into the universe.				
	A_1	–	–	INC of A_1 into Ω	

Table 3.5: Translation from edit operations to Hivent operations

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 are the initial start point t_0, a current 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}. In order 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 this 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 changes in the past. Therefore, the system needs to support retrospective insertion of Hivent operations in between existing operations and therefore update the set of Hivent operations in the system. Each Hivent operation that is not added 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.

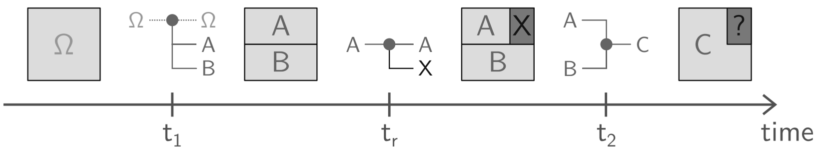


Figure 3.6: Example for a simple conflict due to a retrospective update

The example in figure 3.6 illustrates the problem: Given t_1 with two Areas A and B and a UNI 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 any more, 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 ***a**utomatically* without the interference of the user.
S) The conflict requires the user to choose between two alternatives (***s**emi-automatic* resolution).
M) The conflict is complex and the user needs to resolve it ***m**anually*.

[1] Multiple Hivent operations in one row happen exactly at the same moment and thus are combined.

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: The system has 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 a 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 a Hivent operation. The question is: What happens to 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 there are five possible Hivent operations for both H_o and H_r.

Retrospective Name Change The first five cases are straightforward: Assuming `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` deletes 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. However, there might be a conflict in the next operation that manipulates A_1.

Figure 3.7 illustrates the conflicts that occur in the remaining possibilities for H_o. In 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 influences H_o. In order to maintain spatial integrity, the system has to update the territory of incoming A_1. The territory of B_1 has to be updated as well, because it is enlarged in the same way as A_1. This

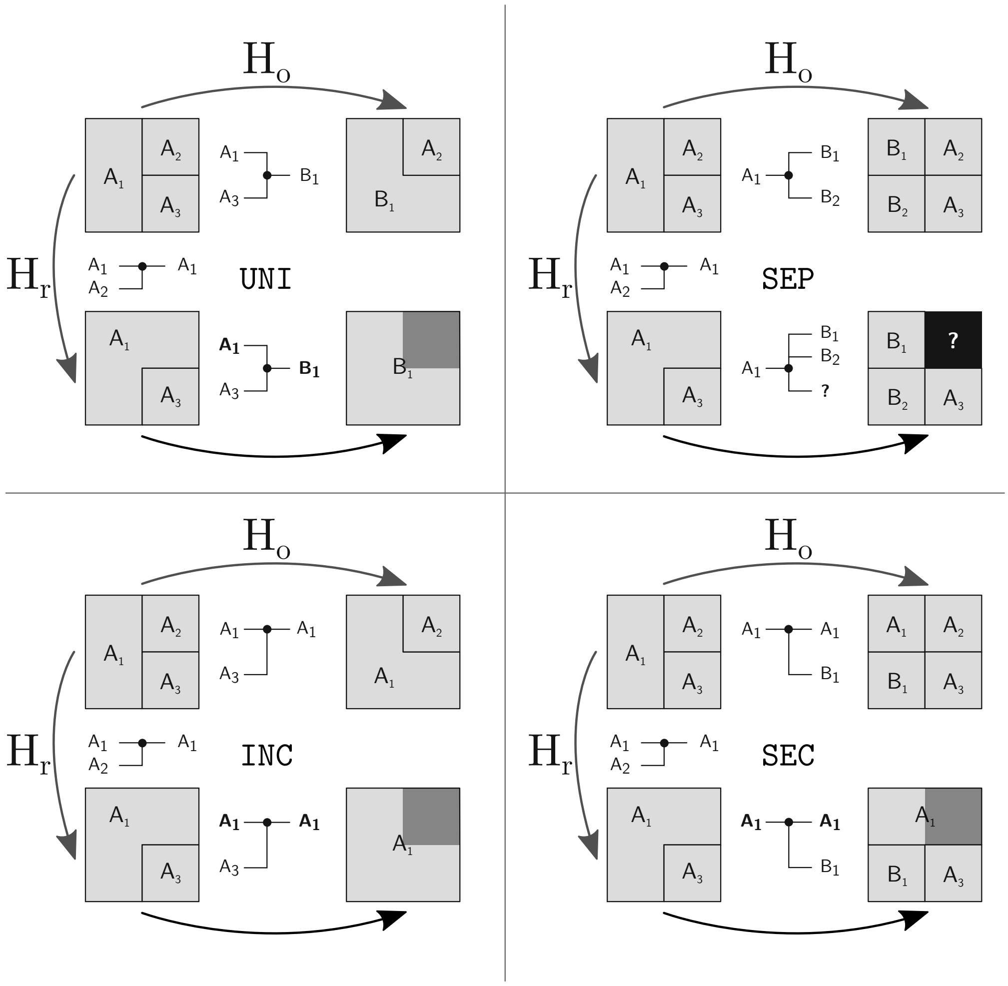


Figure 3.7: Conflicts after a retrospective incorporation

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. This update can cause new conflicts that either the system or the user have to solve, which can then lead to new conflicts again, etc. The system has to repeat this process into the future until all conflicts are resolved. The same is true if H_o is an `INC` operation: the system needs to update the old and new territory of A_1 in H_o and recursively update the territory of A_1 into the future.

If H_o is a `SEP` operation, there is a major conflict: Originally, A_1 split into B_1 and B_2. The territory of A_1 is larger due to H_r. H_o can still separate B_1 and B_2 from A_1 the same way as before, but it is uncertain 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 cannot derive a decision from any existing information. The remaining part could either become Ω, or it could be incorporated into B_1 or B_2, or stay A_1. However, the user has to decide. In case of an original SEC, the situation is slightly different: H_1 still exists as before, just with a larger territory. H_o can secede B_1 as in the first place, 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 is deleted 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 secedes 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 as for retrospective incorporation, there is no conflict if H_o is a NCH, but possibly later.

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 make a decision, because the user's intention when creating R_1 in H_r is not clear: should the newly created Area remain there 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 they prefer. 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 they want 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 conflicts 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 remaining 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 the choice for both conflicting territories if either R_1 should stay an Area or if B_1 or B_2 should incorporate R_1 into their territory. In the first case, the territory of R_1 has to be subtracted from both the incoming A_1 and recursively from the outgoing

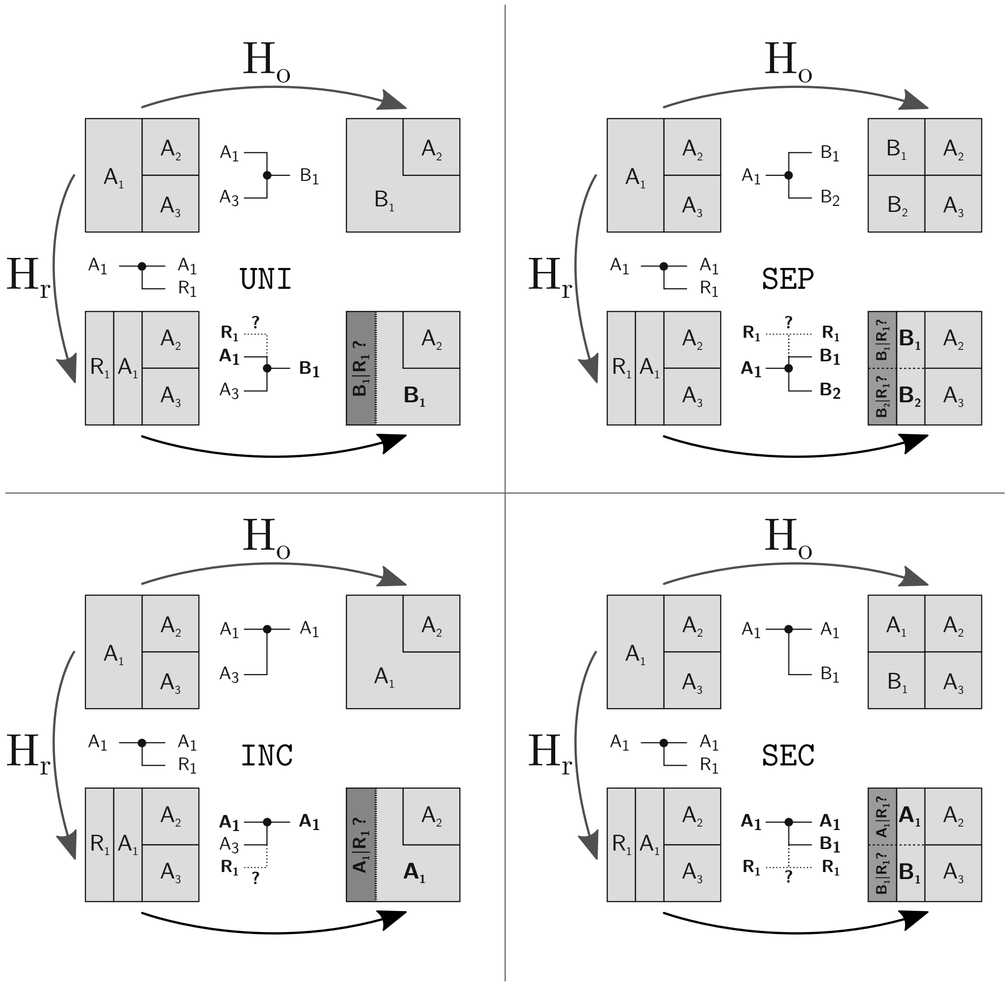


Figure 3.8: Conflicts after a retrospective secession

territory B_1 or B_2. The latter case needs at least one additional Hivent operation: The part of R_1 that shall be part of B_1 or B_2 needs to be seceded from R_1, and in the same moment incorporated into A_1. A_1 itself is separated into B_1 and B_2 at the same time: H_o = `SEC+INC+SEP`. If R_1 is entirely deleted in the operation, then it is completely incorporated into A_1, so it is only an `INC+SEP` at H_o. The case of an original `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 and hence, it cannot be seen as a conscious 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 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 that manipulates 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 any more, because A is already deleted. The only way to resolve this conflict is to automatically delete the NCH operation. The remaining four cases behave in exactly the same way regarding spatial integrity 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.

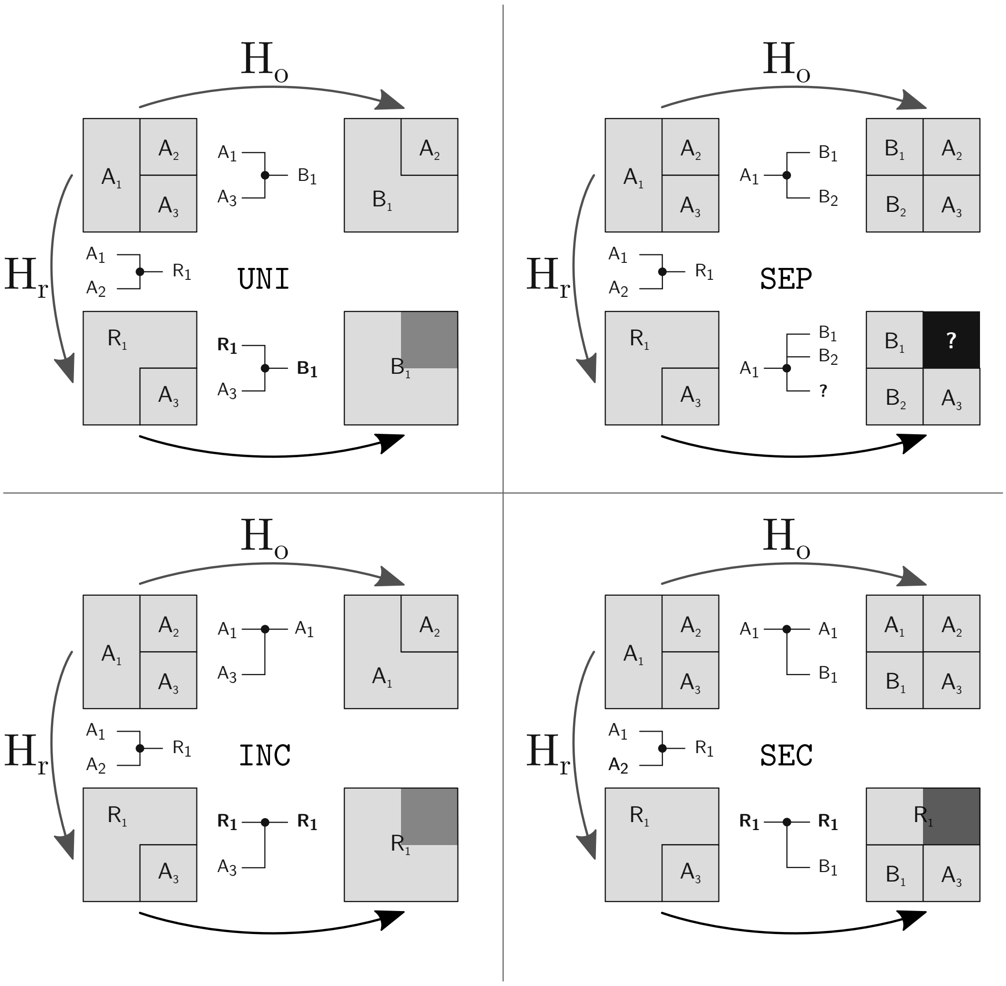


Figure 3.9: Conflicts after a retrospective unification

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 A_1 is deleted 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 any more. Figure 3.10 shows the conflicts that arise for the remaining four cases.

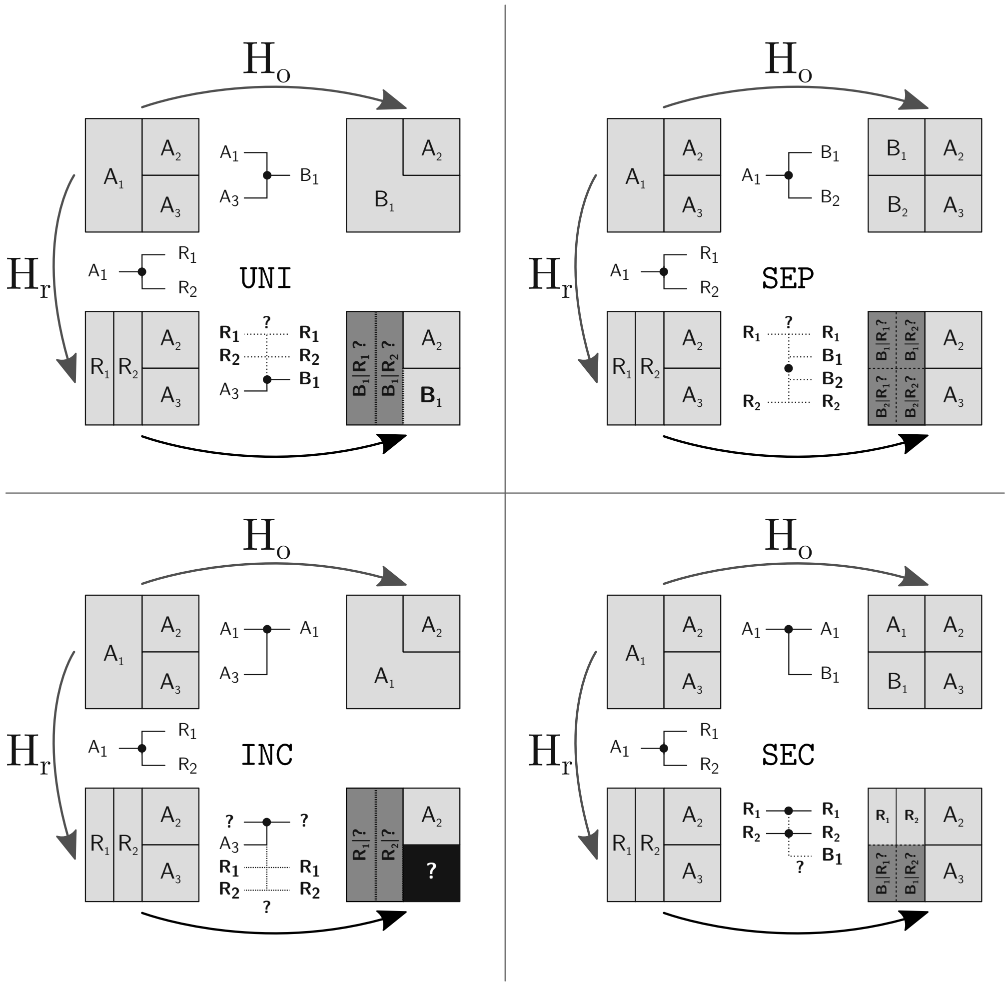


Figure 3.10: Conflicts after a retrospective separation

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 cannot 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 Area created in H_r – in this case only R_1 and R_2 – the system asks the user if they prefer 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. Otherwise, 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 any more. 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 be deleted in this operation. Since there is no information about what should be there instead, this conflict has to be solved manually. The situation is even more complex, because on the one hand, in H_o the user intentionally incorporated A_3 into A_1, which means that there is a clear intention to keep A_1. On the other hand, they separated A_1 in H_r to two new Areas R_1 and R_2, so they can also be seen as intentionally created. In order to avoid any unexpected behavior of the system, the best approach is to give him or her the choice to keep R_1 and R_2, let the user resolve the whole conflict manually in case of denial.

The case of an original SEP is likewise complex: A_1 does not exist any more 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. Otherwise, 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 it not necessary any more, because there is not one simple old Area any more.

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 as 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 been recursively subtracted from the related B_i.

		Original Hivent operation H_o				
		UNI	SEP	INC	SEC	NCH
Retrospective operation H_r	UNI	A	**M**	A	A	A
	SEP	S	S	**M**	S	A
	INC	A	**M**	A	A	X
	SEC	S	S	S	S	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

All possible cases and their resolvability are visualized in table 3.6. It became clear in the extensive examination of the conflicting cases that the insertion of 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, the user has to decide between two alternatives in almost all cases. In three cases there is even a manual resolution necessary. A lot of cases also require recursive updates which means that the retrospective update can lead to a potentially high number of semi-automatic or even manual resolutions by the user. This is likely to lead to frustration. 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 adaption that might simplify retrospective updates and make the data handling less cumbersome.

3.2.4 Backward Operations

From a relative point in time t_i there are two historical directions: forward into the future with a predefined end at the current 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 t_i into the future until t_{now} or until another operation changes it again.

As argued in the previous section 3.2.3 for retrospective updates, it is not sufficient for historical research to have only forward operations to the end of the timeline. A `backward operation` might be useful for historians to edit a state in the past. It is a Hivent operation that is inserted at t_i, but into the past: $t_0 < t_i < t_{now}$. As an example: Given the initial state on 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 not trivial, because the conceptual model, data model and computational model have to be adapted to this requirement.

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

One problem is in the conceptual model: the user interface has to provide a visual clue that the inversion of an operation is possible. Additionally, if a Hivent operation is inserted backwards, another problem occurs: each new Area 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 deletes it in backward direction or creates it in forward direction, otherwise the Area would be active all the way back to t_0. This is probably not desirable.

3.3 User Interface Design Process

The Hivent model presented in section 3.1 and the methods to edit the data in section 3.2 serve as the foundation of HistoGlobe. However, the development of the system bottom-up from the data model to the interface might not lead to a 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. Initially, four researchers of University of Virginia were interviewed and asked about their opinion on the interface concept of HistoGlobe. They confirmed that the combination of a map and a timeline are an appropriate way to interactively visualize the history of countries. Therefore, the concept 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 the *edit mode*, a second interface mode, was developed. It 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

In 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 as a paper prototype. It consists 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 of them had to solve four tasks that cover 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*)

Two paper prototype iterations were created. Each iteration took about three full working days including the creation of the prototype, the conducting of the study and the analysis of the results. Finally, the concept was revised. 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:

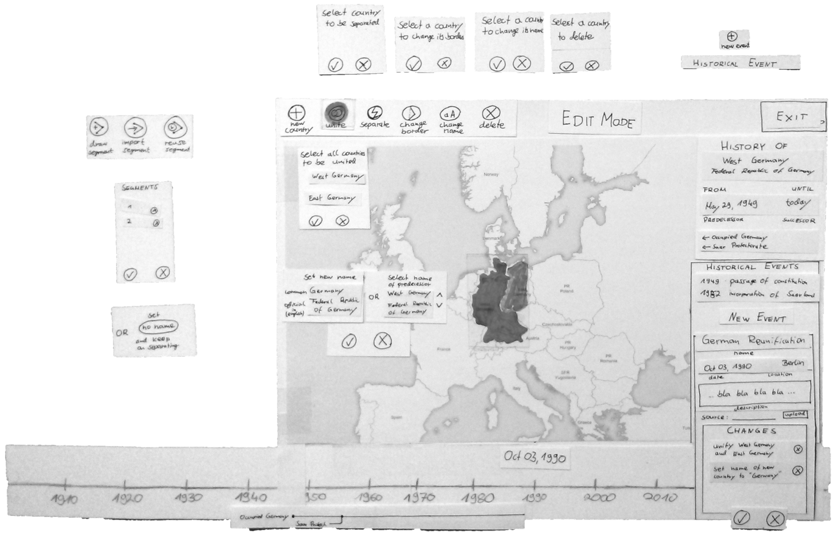


Figure 3.11: The second iteration of the paper prototype for the edit mode

1. The difference between Hivents, the history of a country and a 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 a Hivent-based and an Area-based mental model of the task. This became apparent in the edit operation for the German Reunification: 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 was concluded that the interface has to support both a 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 understood 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 modeled with geometric elements, i.e. 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 the prototype to simulate sudden changes in the interface.

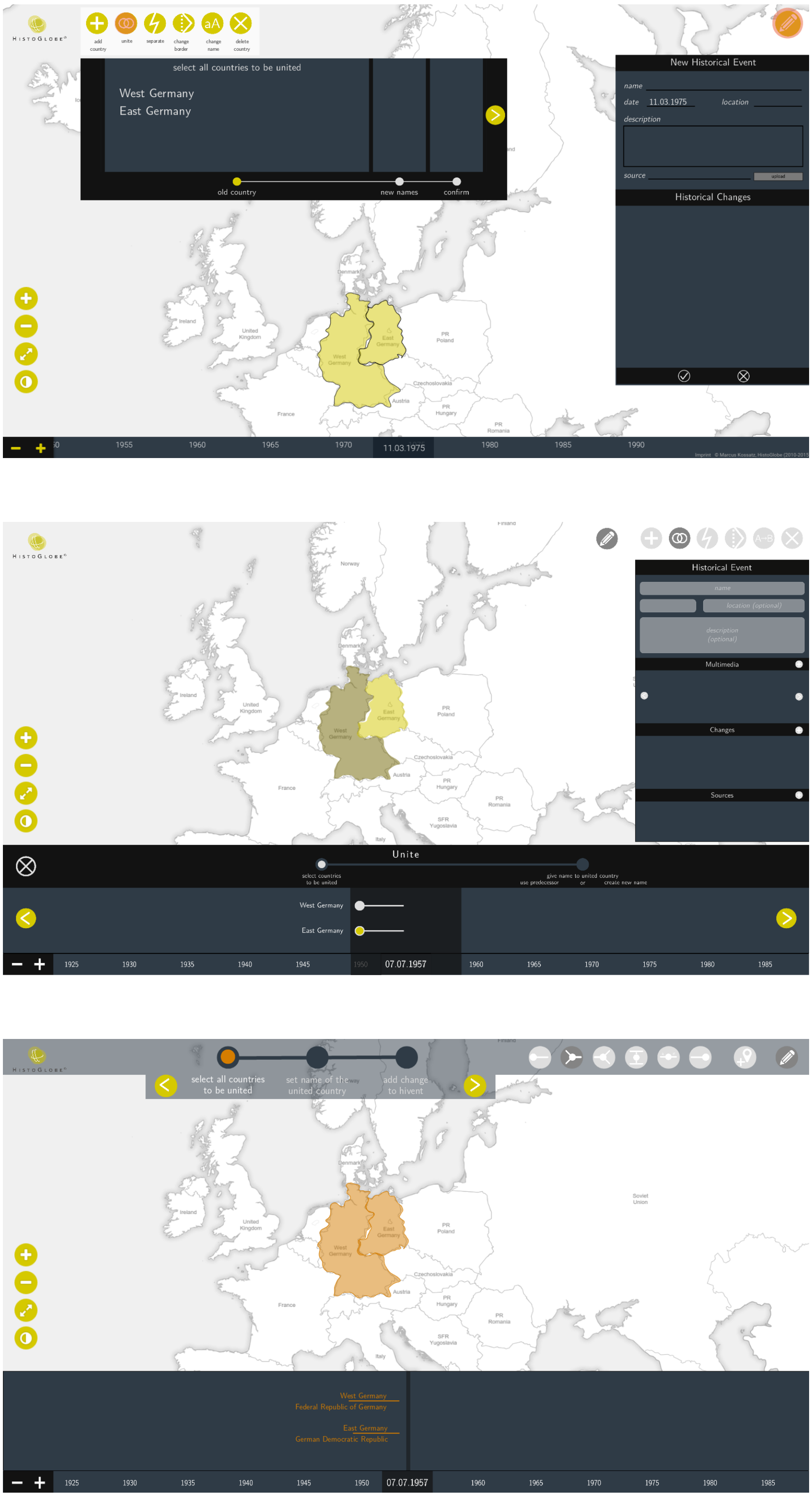


Figure 3.12: Three iteration stages of the mockup prototype for the edit mode

Each prototype iteration was tested with multiple subjects and similar tasks as for the paper prototype. The interface was changed from one test to the next one. Several 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"

"there is a training session needed"

"the interface is very clear and graphically pleasing"

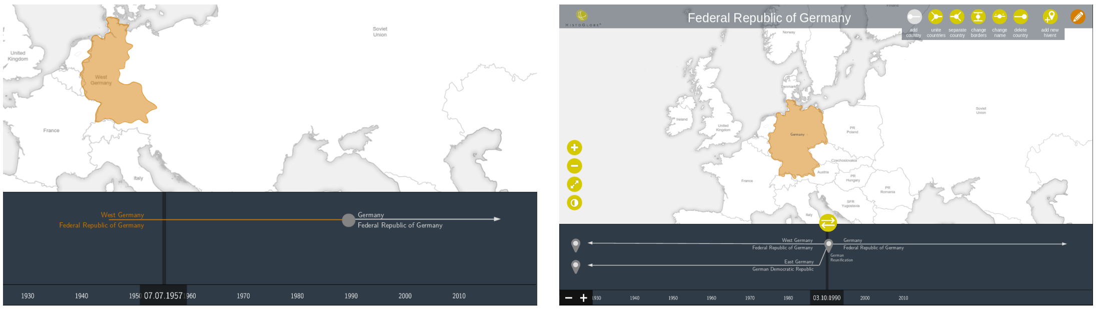
"the logic makes sense, it is just very complex"

"it's looking good"

"a nice tutorial and a good documentation are necessary"

– quotes from the users of the mockup prototype

The user tests revealed that especially introducing a retrospective update and a backward change are very challenging. For both problems a design solution was developed. 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. A button that flips an edit operation to initiate a backward operations 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 gray 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 that the users understood. Its creation took longer than the paper prototype, but it 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 time 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, the control buttons and the edit mode.

Not all desired features could be implemented: As mentioned in section 3.1.4, there were too many conceptual problems for the HistoGraph that had to be solved first. Backward operations and retrospective updates are not supported as well, due to their complex nature in the computational model. 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 using the example of the fictional secession of Scotland from the United Kingdom in 2018.

Figure 3.14: Initial state of the browsing mode

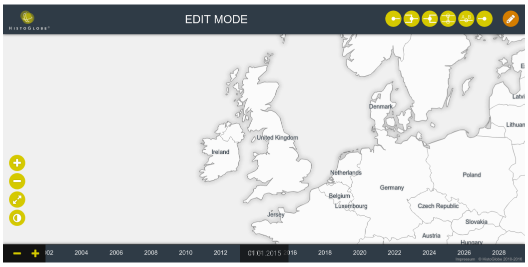


Figure 3.15: Initial state of the edit mode

The initial state of the user interface. Additional to the original elements, there is an edit button on the upper right corner. A click on that button enters the edit mode of the system.

In the edit mode, a title bar and six buttons for the edit operations appear. A click on one of these buttons starts the associated operation workflow as introduced in section 3.2.2.

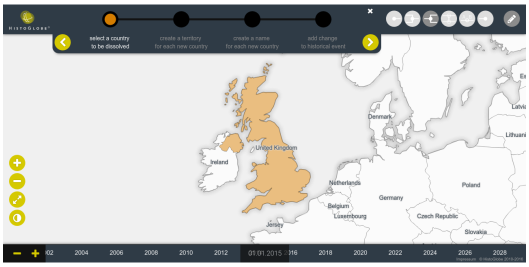


Figure 3.16: 1) Select old Areas

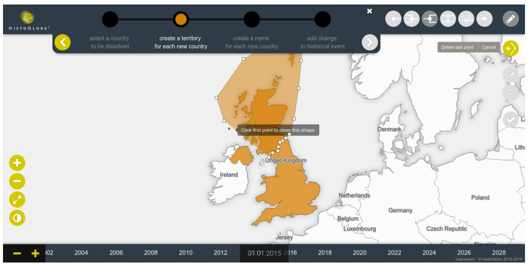


Figure 3.17: 2) Set a new territory

A *Workflow Window* guides the user step-by-step through the process of completing the edit operation. In case of `SPL`, the user has to select the country that he or she wants to be split via click on the map. After the step is completed, a click on the next button in the workflow window proceeds to the next step. At each point in the workflow, the back button reverts the previous action.

In step two, the user has to create a territory for each new Area. The *New Territory Tool* provides the option 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 has been created successfully, the remaining old territory can be selected on the map and finally be used as the last territory. If the entire old territory is distributed among the new Areas, the workflow proceeds to the next step.

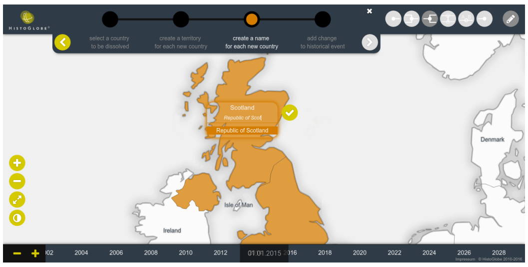


Figure 3.18: 3) Set a new name

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. The user can make use of instant search to select existing country names from the database to be applied. The short name is put directly on the map once the confirm button is pressed.

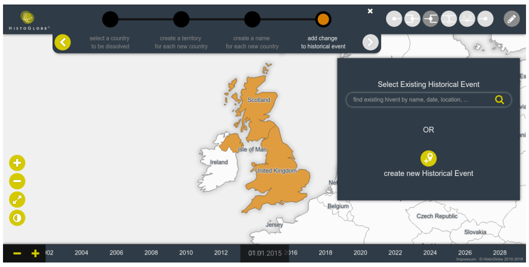


Figure 3.19: 4) Add the Operation to a Hivent

When all names are set, the edit operation is complete. It has to be added to an Hivent in the last step of the workflow. The *New Hivent Box* offers two options: either search for an existing Hivent add the edit operation to it or create a new Hivent.

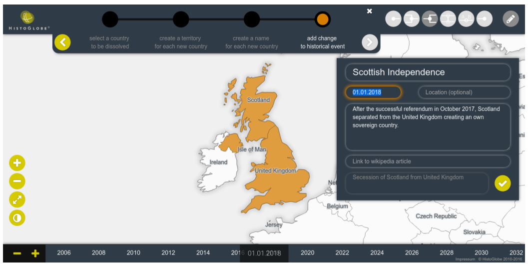


Figure 3.20: 4) Create a new Hivent

The new Hivent created for that change is the "Scottish Independence" on 01.01.2018 with a description of the Hivent and optionally a location and a link to a Wikipedia article. In the last line, the historical change "Secession of Scotland from the United Kingdom" is noted. A click on the confirm button finalizes the edit operation workflow.

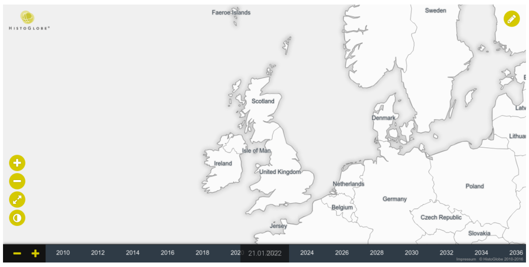


Figure 3.21: The final state with Scotland

A click on the edit button quits the Edit mode and returns to the browsing mode. Scotland and the United Kingdom are now separate Areas on the map after 2018. When the timeline is moved prior to 2018, Scotland is still part of the United Kingdom.

3.4 Implementation

HistoGlobe is a web-based Historical Geographic Information System. The data model and the conceptual interface model 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. The remaining part of the section presents the implementation of the server-side and the client-side application in HistoGlobe.

3.4.1 System Architecture

HistoGlobe uses a classic client-server architecture of a web-based information system that is shown in figure 3.22. The user opens the application in a Web browser, the *client* side of the system, and interacts with it through the user interface. 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 for new data to the web server. The middleware checks the request and sends it as a query to the database to obtain the required data. It then transforms the data and sends it back to the client. The interface shows the new information.

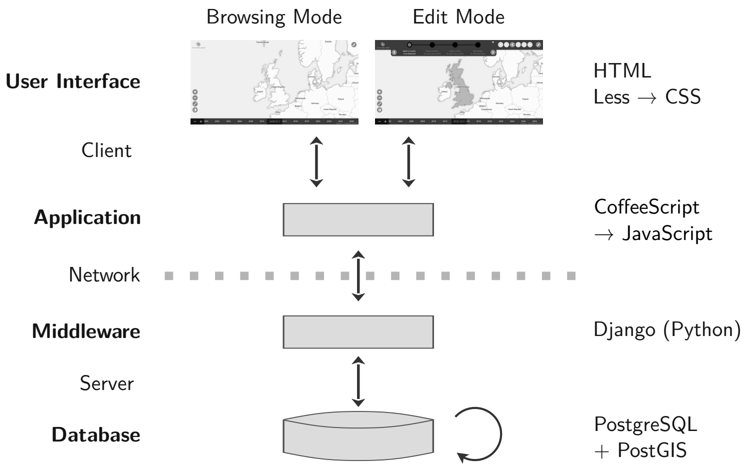


Figure 3.22: The system architecture of HistoGlobe

This clear separation between data, application and user interaction in the system follows directly from the *model-view-controller* pattern. One part can be changed independently from the others. 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* [2], a free and open-source web framework combined with *PostgreSQL* [3], which was 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* [4] is used as a spatial database extension for PostgreSQL. With these tools, the Hivent model from section 3.1 was implemented in a database model shown in figure 3.23. This is the final result of a highly iterative process that was subject to several improvements and adaptations to new requirements introduced in the human-centered design process. The model is structured in two parts:

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 Ω, 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 when 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 `EditOperation`s introduced by the user. They are an abstraction layer in the Hivent model between the `Hivent` and the `HiventOperation`s. The operations replace a set of `OldArea`s with a set of `NewArea`s and might update the name or the territory of one specific `UpdateArea`.

Figure 3.24 shows the Hivent database model using the example of the German Reunification on 03.10.1990. Before 1990, there were the Areas `GDR` ("German Democratic Republic", East Germany) and `FRG` ("Federal Republic of Germany", West Germany). A user introduced a (`MRG`) operation in

[2] *Django*, The Web framework for perfectionists with deadlines, URL: `https://www.djangoproject.com/`, accessed on: 27.05.2016

[3] *PostgreSQL:*, The world's most advanced open source database, URL: `http://www.postgresql.org/`, accessed on: 31.10.2015

[4] *PostGIS*, Spatial and Geographic Objects for PostgreSQL, URL: `http://postgis.net/`, accessed on: 27.05.2016

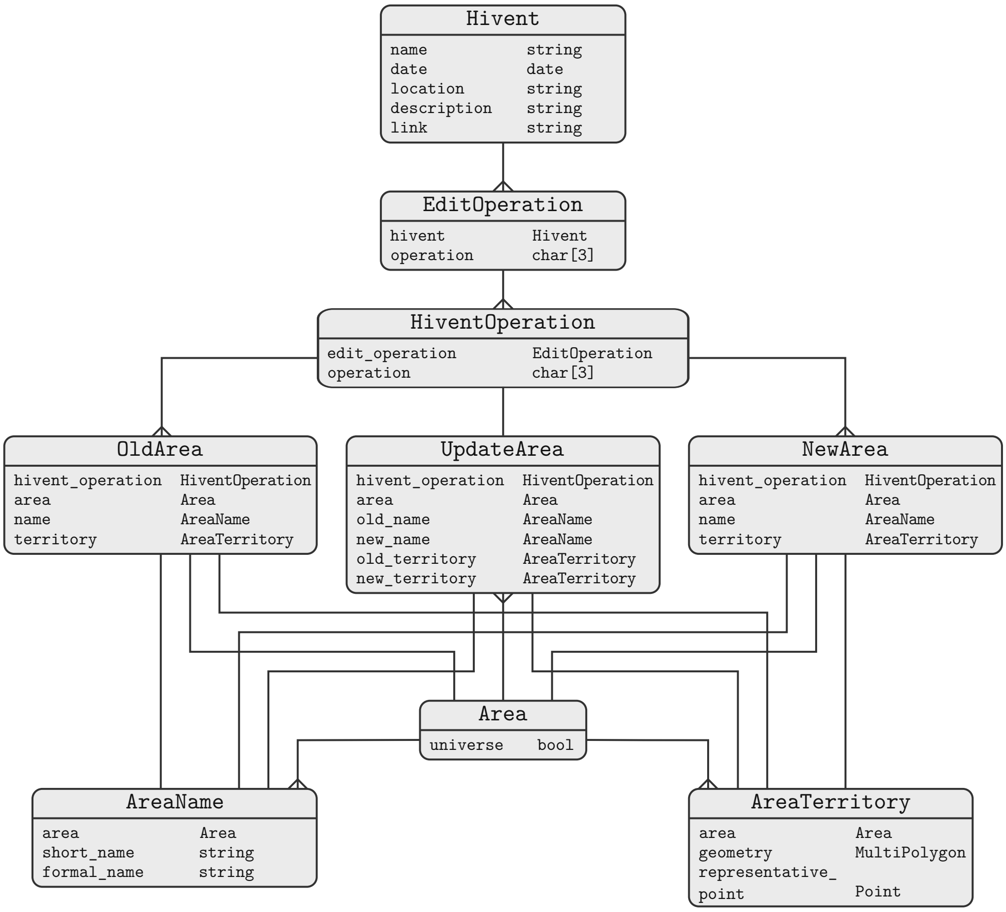


Figure 3.23: The Hivent Database Model
Each entity additionally has an `id` attribute, which is omitted for simplification purposes.
Simple arrows denote a `1:1`-relationship, branching arrows a `1:n`

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 is deleted, 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 beyond 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. The issue regarding international recognition is explained in section 2.1.1. Seven Areas were created by a `SPL` operations from their homeland on the day of their

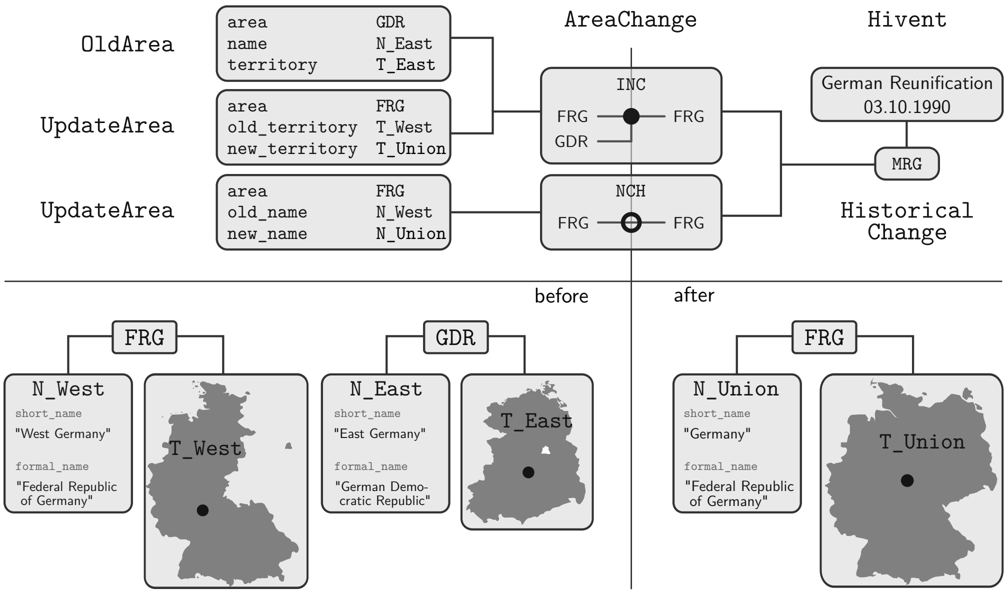


Figure 3.24: Visualization of the German Reunification in the Hivent Database Model

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 following two use cases:

1. **`get_all`** is called by the client upon initializing the web service. The server responds to this `HttpRequest` with all data from the database in a single *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 `HiventOperation`s, data about each `OldArea`, `UpdateArea` and `NewArea`. The view checks the data for integrity and stores it in the database. The method returns a confirmation to the client and a set of final `id`s 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 modular system. The modules used in this this implementation of HistoGlobe are emphasized in **bold** in the class diagram in figure 3.25. The remaining classes were instantiated at run time by one of the seven modules. The classes are structured by their functionality regarding the Model-View-Controller pattern.

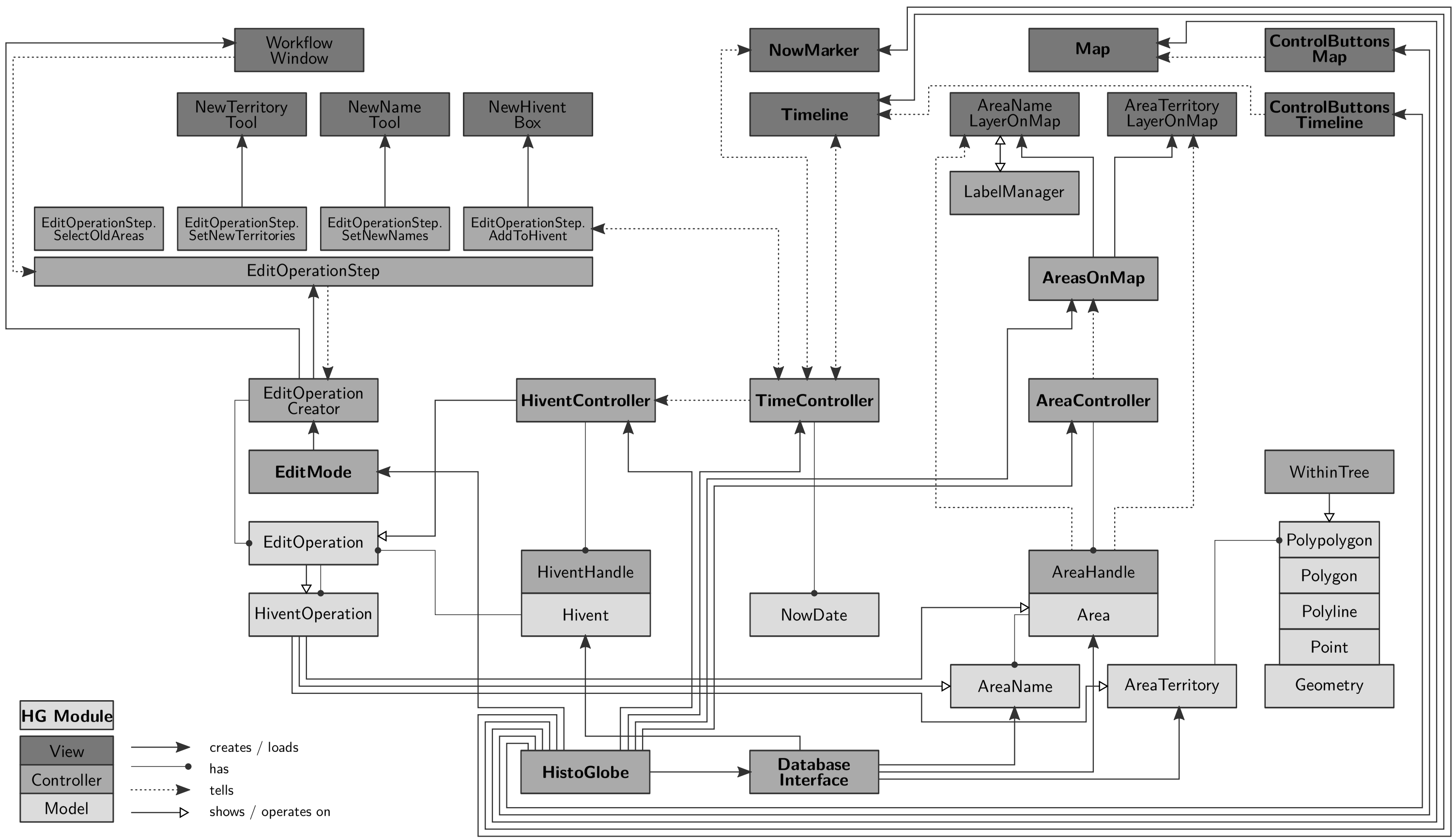


Figure 3.25: Modules and classes in HistoGlobe

Initialization The main HistoGlobe instance at the bottom initializes all modules, mainly the interface elements, the controllers and the `DatabaseInterface`. This class communicates with the middleware as seen in section 3.4.2. It loads data from the database and stores data in it. Initially, each `Hivent` and the related `EditOperation`s are created. Each `HiventOperation` is assembled by its associated set of `OldArea`s, `NewArea`s 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 `HiventOperation`s via the `startOperation`, `updateOperation` and `endOperation`.

Execution of 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.

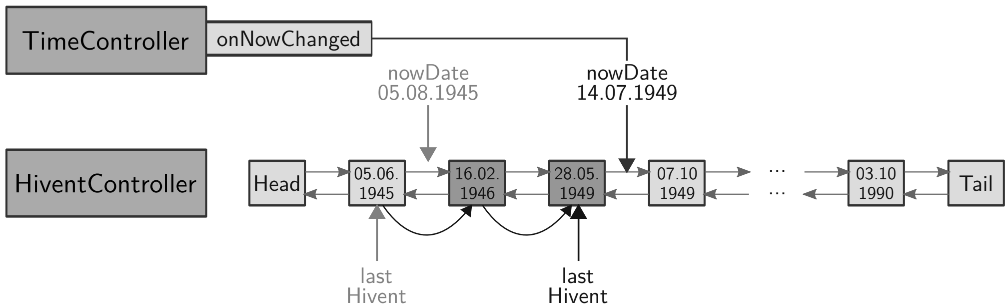


Figure 3.26: Detecting the next Hivent that happens in the HiventController

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` that has happened and a copy of the current date in the list. It listens to the `TimeController` – if the `NowDate` changes, the `HiventController` checks if the next `Hivent` has happened. It compares this new date with its current date and checks if `Hivent.date` is in between these two dates. If this is the case, the `Hivent` happens and each `EditOperations` associated to this `Hivent` is 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 the `Hivents` backwards through the list and executes all the `EditOperations` backwards. This simple data structure effectively and efficiently manages temporal changes of `Area`s on the map.

The initialization of the system works as follows: all `Hivent`s 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, then it happens in the following three steps:

1. For each `newArea`, the `AreaName` and the `AreaTerritory` that the `Area` has in the moment are historically created and 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 with the `newName` or `newTerritory`, respectively. The `update` method of the `AreaHandle` updates the respective layers on the map.

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

```
class HiventOperation
    #... initialization of the member variables

    @oldAreas   = []  # {area, name, territory}
    @newAreas   = []  # {area, name, territory}
    @updateArea = {}  # {area, oldName, newName, oldTerritory, newTerritory}

  execute: (direction) ->

    if direction is 1

      for newArea in @newAreas
        newArea.area.name =       newArea.name
        newArea.area.territory =  newArea.territory
        newArea.area.handle.show()

      for oldArea in @oldAreas
        oldArea.area.name =       null
        oldArea.area.territory =  null
        oldArea.handle.hide()

      if @updateArea
        if @updateArea.newName
          @updateArea.area.name =      @updateArea.newName
        if @updateArea.newTerritory
          @updateArea.area.territory = @updateArea.newTerritory
        @updateArea.area.handle.update()
```

Listing 3.1: Execution of an `HiventOperation` in forward direction

EditMode The Edit Mode is the main contribution of this thesis to the HistoGlobe project. Its interface was previously introduced in section 3.3.3 and its implementation is briefly explained here. When the user clicks an edit operation button in the edit mode interface, internally an `EditOperationCreator` sets up the `WorkflowWindow` in the interface. Initially, the first `EditOperationStep` for this operation is loaded. 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 requires the user to select a certain number of old Areas for the edit operation, i.e. one Area for `SPL`, `REN` and `DEL`, one or two Areas for `CHB` and any number of Areas for `MRG`.

2. `CreateNewTerritories`: The map is set into 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 the territory A^T of each existing Area A if the territories intersect ($A^T \cap B^T \neq \emptyset$). If they do, 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 is empty, i.e. $A^T = \emptyset$, it means that A is completely covered by B, so it is entirely incorporated into B and 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 user interaction. The new territory B^T is created by the union of the territories A_i^T of the selected Areas $A_i \in A$ from the first step: $B^T = \bigcup\limits_{i=1}^{n} A_i^T$. It is set as the territory of B.

   `SPL` Each desired new Area $B_i \in B$ needs a territory B_i^T. This is obtained by intersecting the temporary territory T_i that the user has created in the `NewTerritoryTool` with the territory of the selected Area: $B_i^T = A^T \cap T_i$. This is repeated until the entire 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, then territory T_1, created by the `NewTerritoryTool` in this step, 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 territory A_1^T. It is temporarily 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 from this temporary union with an intersection: $A_1^T = B^T \cap T_1$. A_2 receives the remaining part as its new territory: $A_2^T = B^T \setminus T_1$.

No new territories are created in `REN` and `DEL`.

3. `CreateNewNames`: For each Area that is created or updated in this operation, the user needs to create a new `AreaName` with a short name and formal name using the draggable `NewNameTool`. The user writes the new names or selects them from the suggestions. Not only the name will be taken from the `NewNameTool`, 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`: The user creates a new Hivent using the `NewHiventBox` 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 to the server along with the associated `Hivent` and finishes the operation.

WithinTree 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, they create holes in the polygon. A polygon consists 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 are 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* data structure is introduced. The idea is to set up a hierarchical structure of polygons that contain each other. An example WithinTree for a set of polygons is shown in figure 3.27.

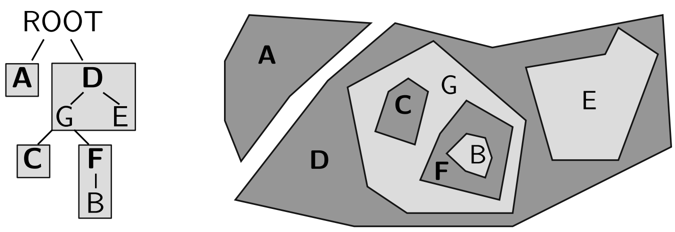


Figure 3.27: The WithinTree (left) for the set of polygons (right)

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 polygons have to be extracted in the correct structure with respect to their outer and inner rings. Therefor, the WithinTree has to be traveres 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.

```
insert: (newNode, parentNode) ->

  ## PREPARATION
  # case 1) newNode also in 1 child of parentNode  -> withinChild
  # case 2) 1+ children of parentNode in newNode   -> containChildren
  # case 3) no hierarchical relation between newNode and any other node

  withinChild =     null
  containChildren = []

  for childNode in parentNode.children

    if newNode.isWithin(childNode)                 # check if case 1)
      withinChild = childNode
      break # no other hierarchical relation to any other child possible

    else if childNode.isWithin(newNode)            # check if case 2)
      containChildren.push(childNode)

  ## EXECUTION

  if withinChild                                   # case 1)
    @insert(newNode, childNode)

  else                                             # cases 2 and 3)
    # newNode is not in any child of parentNode => place it underneath
    @_nodes.push(newNode)
    newNode.setParent(parentNode)
    parentNode.addChild(newNode)

    for containChild in containChildren            # case 2)
      # => detach from parent node and place them as children of newNode
      containChild.setParent(newNode)
      newNode.addChild(containChild)
      parentNode.removeChild(containChild)
```

Listing 3.2: Insertion of a polygon node into the WithinTree

LabelManager A major visualization problem that was sufficiently solved in the thesis is the label collision problem. Each active Area has both a territory and a name that should be shown on the map. Axiom 5 states that no territories can overlap, so 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
- `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 descending priority order. The algorithm is based on the following heuristic: *A label is shown unless it is covered by a higher-priority label*.

When a new label is introduced, it is inserted into the `LabelManager` in the following way: Each label has to be inserted in the correct position in the `labelList`. Therefor, it is compared with each existing label in descending priority if it overlaps 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 – otherwise it is shown. In the latter case all lower-priority labels are checked if they are covered by the new label – if they are, then they are hidden.

If an Area is also deleted, its name is also 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 any more. If no other label still covers it, the label can be shown.

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 hidden labels can be shown if they are not covered by any other label any more. Vice versa, if the user zooms out, the labels have less space to their neighbors. No additional label can be become visible, but a visible label needs to be hidden if it is covered by at least one other label.

Figure 3.28 shows the result of the `LabelManager` on the map of Europe in 2016. It is obvious that no two labels collide. This 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. If a short name has more than one word, automatic line breaks at could be inserted at reasonable positions, e.g. "Czech|Republic". With this possible extension, more labels could be shown. However, this is subject to future work. The `LabelManager` sufficiently serves the purpose of this thesis.

This chapter presented the iterative design approach to the core part of the thesis and its main results. The Hivent model with the five low-level Hivent operations to describe a historical change is introduced. Additionally, methods are proposed to edit these changes using six edit operations in

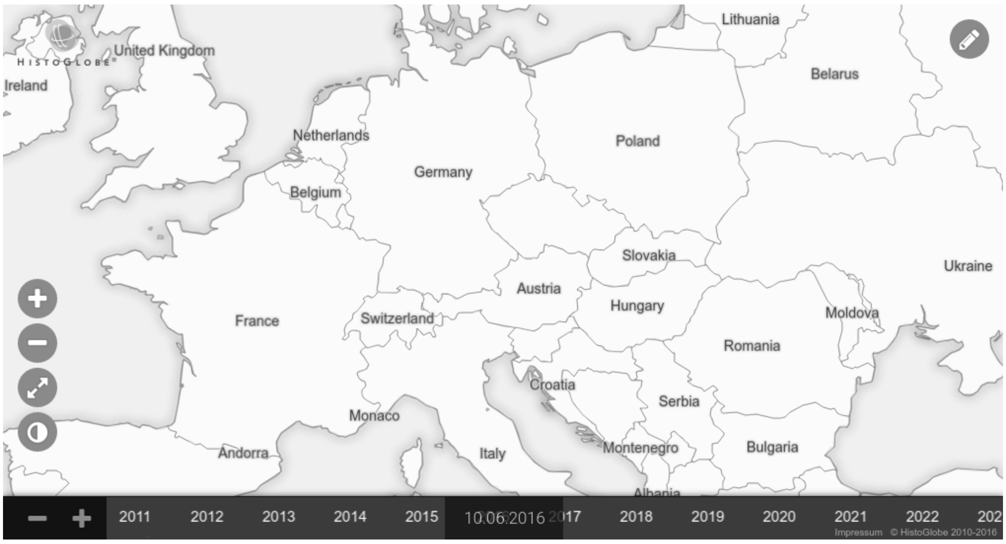


Figure 3.28: The resulting labels on the map of Europe in 2016

a workflow. These operations form the basis of the edit mode in the user interface of HistoGlobe. Finally, the system architecture and some selected implementation issues are presented. 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. It focuses on the important aspects of the uncertain nature of history in the first section. The second part of this chapter develops extensions to the Hivent model and the edit mode for HistoGlobe that deals with problems of uncertainty and disagreement.

4.1 Evaluation

In order to evaluate a model, it is important to understand the concepts of *uncertainty* and *disagreement*. The model in an information system tries to resemble the real world as much as possible and necessary. Both concepts represent two different kinds of problems in two different domains. *Disagreement* describes a conflict in the real world. On the 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 a 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 modeled 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 already exists in the real world. The conflicting versions of the story can be modeled very precisely, but in order for the model to be accurate as well, it needs to support the declaration of an Area as a contested territory.

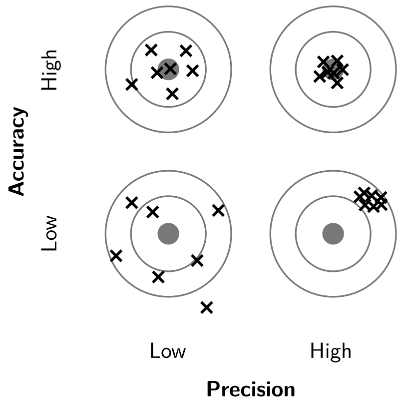


Figure 4.1: The difference between accuracy and precision

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 that historians have 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 information in the HGIS 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 a Hivent. In reality, the name can have different versions: an official version and a usually shorter 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` intends to to represent the temporal dimension of a historical event. While a historical change itself is discrete and happens at exactly one point in time, the historical event yielding this change might not. The decisions of the "Congress of Vienna" came into effect on 09.06.1815, 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" of 1898 which had a predefined length of 99 years. Therefore, the treaty has two dates with historical changes: 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 1917 October Revolution in Russia happened in November according to the Gregorian Calendar system used in Russia at this point, but in October in the Julian Calendar. Time zones can also play a crucial role: The German Instrument of Surrender that ended World War II in Europe came into effect on 08.05.1945 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 one day later. While the `Hivent.date` field in the data model works with time zones, 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 be a city, a battlefield or a region. The model is not very precise, because the actual geospatial location or region in which a 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 a 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 the lack of answer to 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 that consists of a territory and a name is problematic in terms of accuracy and precision. As discussed in section 2.1 in detail, it is impossible to objectively model a country without running into conflicts. The axioms of the model state that there cannot be overlapping territories at the same time – although the nature of a contested territory is that it is claimed by more than one party. While the data model is able to to represent it as an own Area, it does not support any evidence of uncertainty about the status of an Area, e.g. if it is contested or has limited international recognition. The model does not support hierarchies of Areas as well, 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 shortcomings as 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.

The `AreaTerritory` is more problematic, because the two assumptions 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 a 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. It is not possible to provide information about the expected accuracy of a territory in the data model. Another problem is that the territory is stored as a whole polypolygon. Different parts of the border can have different types or levels of certainty, e.g. one part has a well-established and demarcated border to neighboring country X, but the border to neighbor 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 are mutually exclusive, i.e. no operation can be replaced by a combination of two others [For].

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, just like `INC` and `SEC`, inverse operations, so they are exclusive. `INC` requires one update Area, multiple old Areas and no new Area, whereas `SEP` has several new Areas and does not have an Area that preserves its identity. Therefore, `INC` and `SEP` are mutually exclusive, just like their inverse pair `UNI` and `SEC`. 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: On the one hand, `INC` and `SEC` model one update Area whose identity is preserved. In `UNI` and `SEP` on the other hand, there is no Area that is preserved. These two cases cannot be mapped onto each other. This shows that all operations are mutually exclusive.

In order to show that the collection of Hivent operations is exhaustive, all possible changes in the history of a country have to be examined. This includes its identity (formal name), short name and territory. In reality, a historical country was once created, possibly underwent several changes of its short name and territory and was deleted one day. The creation of an Area on previously unclaimed land can be expressed by a `SEC` from Ω. 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 was previously deleted. 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 is deleted, it can either have none, one or multiple direct historical successors. The first case is represented by an `INC` into Ω, 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 modeled 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

In addition to the data model, the implementation of HistoGlobe can partially be evaluated as well. 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 about the missing feature to import existing geometries from external sources or from historical maps. While this is a highly desirable feature, it was beyond the scope of this thesis but is subject to future work.

Time and memory estimation The system performs well with the current content of the database: it takes about 3.5 seconds to initially load the data from the server to the client. As explained in section 3.4.2, the client receives the data from the database all at once. This behavior is not scalable, but sufficient for the initial dataset. However, the size of the databases may increase significantly due to the territories.

This storage analysis assumes that HistoGlobe manages a significant amount of historical countries since the beginning of the common era 2000 years ago. There are currently about 200 countries on Earth. Wikipedia lists another 700 modern states in the "List of former sovereign states" [1] 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 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" [2] collects about 75 border changes, about 100 border changes are listed since World War I. Therefore, the estimate of one international border change per

[1] *List of former sovereign states*, `https://en.wikipedia.org/wiki/List_of_former_sovereign_states`, accessed on: 02.06.2016

[2] *List of national border changes from 1815 to 1914*, `https://en.wikipedia.org/wiki/List_of_national_border_changes_from_1815_to_1914`, accessed on: 02.06.2016

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 Ω, each time the territory of one Area changes, at least one other will do so as well. As a conservative estimate, 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 for both the database and the client-side application, because it fits into the main memory of each modern computer system.

However, for practical purposes speed is more important than storage. The goal is to minimize the time in which the user sees a blank map on the screen. A standard DSL connection has a bandwidth of $16.000\ \frac{KBit}{s} = 2\ \frac{MB}{s}$. Loading all data at once would take $\frac{250\ MB \cdot s}{2\ MB} = 125\ s \approx 2\ min$. This motivates to develop a sophisticated algorithm to gradually load the data from the server to the client. One idea is that the client initially sends the current date and the center of the viewport to the server. The server responds with the currently active Areas close to the viewport center first and then sends the remaining data to the client subsequently. Another idea is to store and load different levels of detail of the vector data. Initially the server would send only the coarsest level to the client and gradually increase the granularity until the whole data is sent ~~to the client~~.

4.2 Modeling Uncertainty

The first part of this chapter presented a list of shortcomings of the current Hivent model and the implementation, most of them focusing 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 ignored in the rest of this chapter:

- The assumptions 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.

In order to tackle the remaining shortcomings of the current concept, both the user interface and the data model have to be extended. The interface shall prevent the illusion of certainty and provide tools to express uncertainty. 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. This is why all sorts of uncertainty must be quantified. The computer needs to know exactly how uncertain the user is about the data, although it seems counterintuitive to the idea of uncertainty.

4.2.1 International Recognition

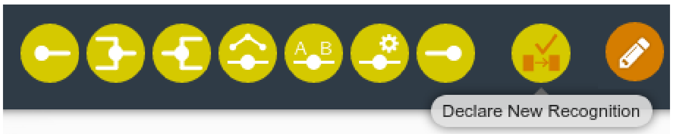


Figure 4.2: Newly designed and extended buttons for edit operations.

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.

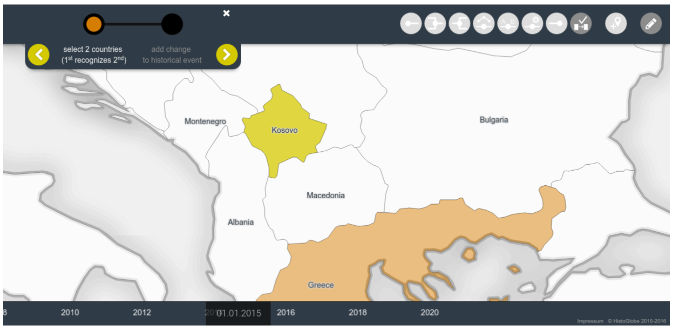


Figure 4.3: New edit operation: `REC` sets up the recognition of one Area to another.

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.

A contested border that is product of a disagreement between two and more parties, can be visualized by a dashed line. If the border is uncertain, i.e. it has a low expected accuracy, three different visualization methods are introduced:

1. Blurred Border: The higher the uncertainty, the wider and blurrier 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.

The borderline is assigned a degree of certainty. In the interface this is controlled by a horizontal slider, in the data model as a certainty value ($\texttt{certainty} \in \;]0..1]$). Absolute certainty ($1.0$) creates a sharp and crisp line on the map. 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.

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. With this extension, the underlying map tiles as raster data must be eliminated. A body of water is modeled by an Area as well, and this this is subject to change as well, the distribution of land and water can not be used as a static layer any more.

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.

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.

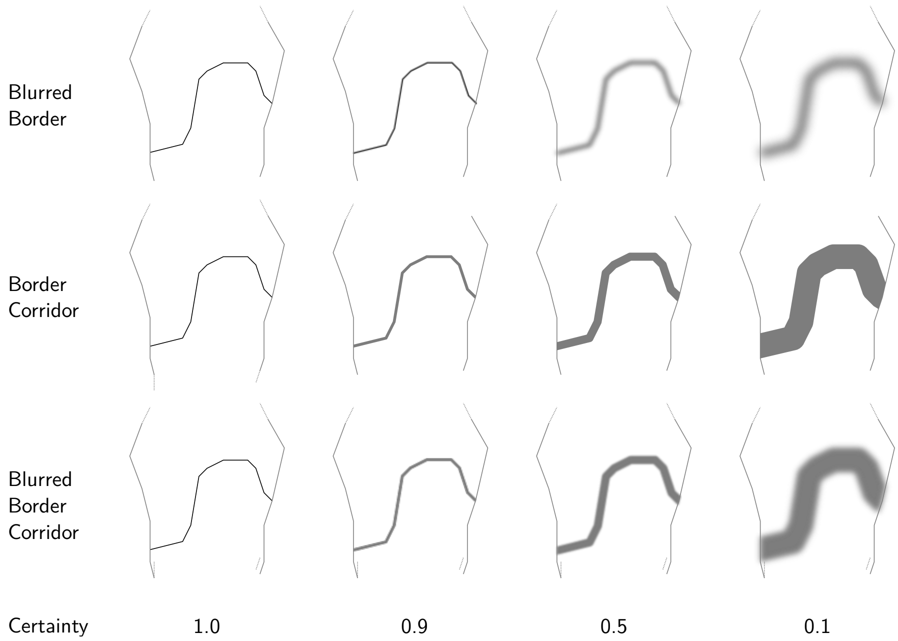


Figure 4.4: Three different methods to visualize uncertain courses of a border

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 that can govern its territory and people independently from other countries and that has a significant international recognition, e.g. Germany.
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 German federal state have no external autonomy (0). Autonomous countries within another country, like England to the United Kingdom or Greenland to Denmark, have a certain degree of autonomy ($\in]0..1[$). The value 1 is excluded, because full autonomy means the territory is a sovereign country.

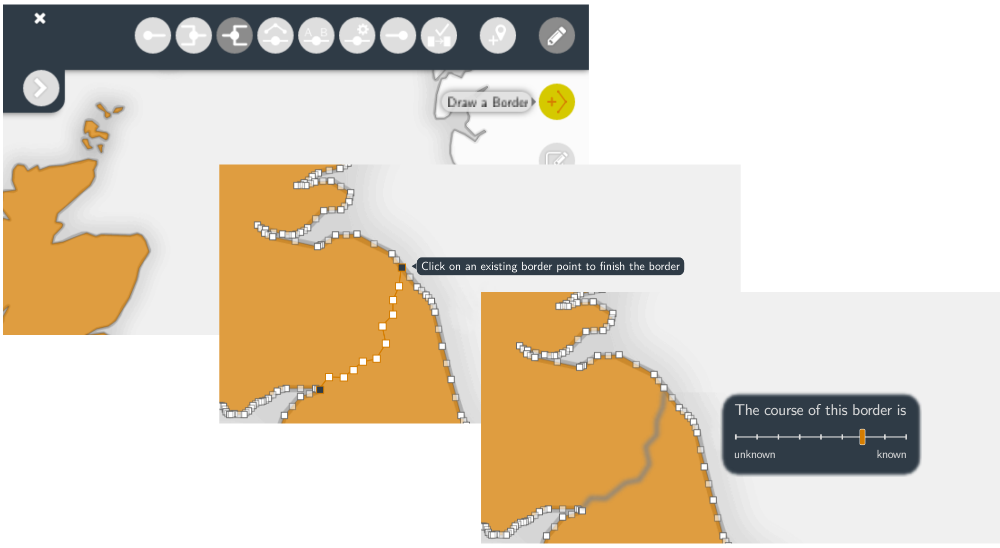


Figure 4.5: Drawing historical borders instead of full Areas and defining a level of certainty.

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.

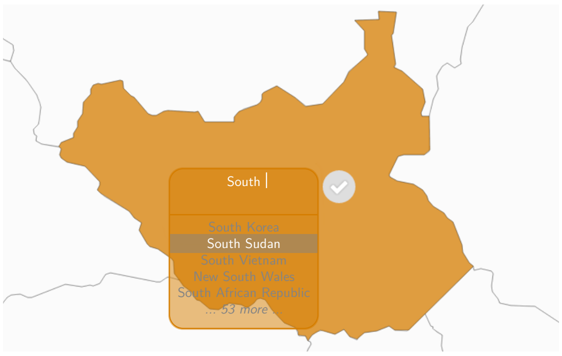


Figure 4.7: Getting suggestions for the name from Wikipedia.

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

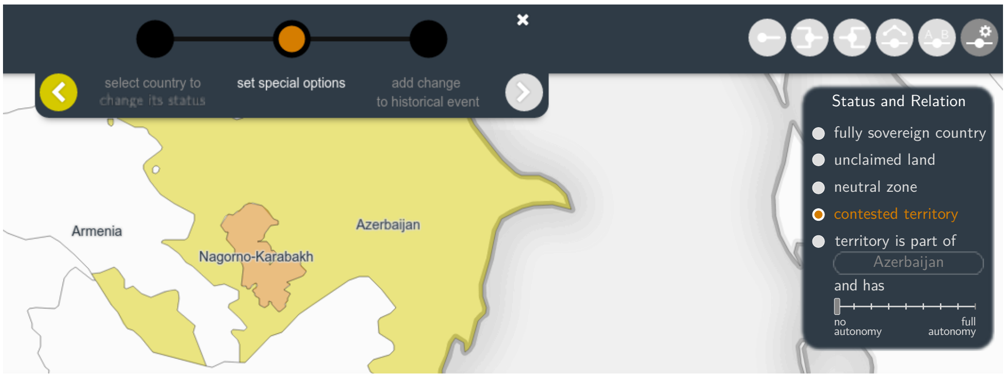


Figure 4.6: Defining a special status or relationship to a territory.

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, Hivent names can be chosen from a collection of Wikipedia articles as well. 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.

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, the data model has to be adapted. Mainly the thematic domain of the Hivent model changed, since the status of an Area, its recognition or relation to other countries are non-spatial attributes 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 an identity-preserving change as well. The remaining Hivent operations do not have to be fundamentally changed, but only extended: the old

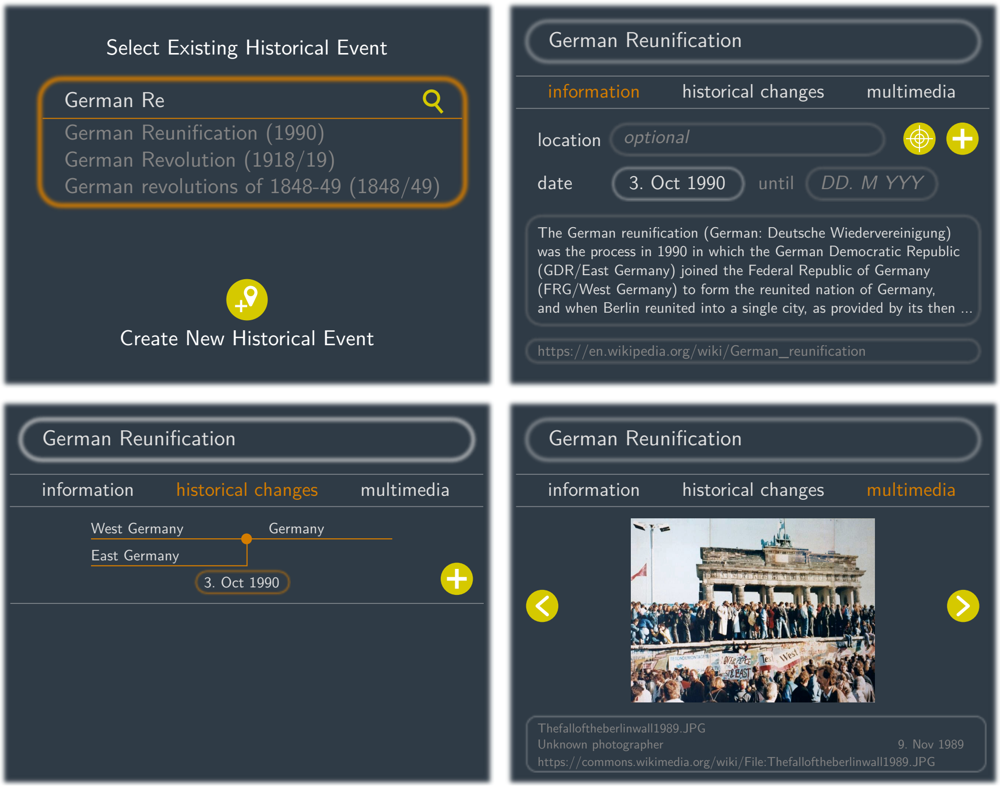


Figure 4.8: Creating a new Hivent and adding the newly created edit operation.

and new Areas receive an additional `status` attribute to create or delete the current `AreaStatus` associated to the `Area`, respectively. This minor extension shows the robustness of the Hivent operations.

More thought must be given to the question of how to express a new recognition of an Area. 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, because 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.

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 a 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.

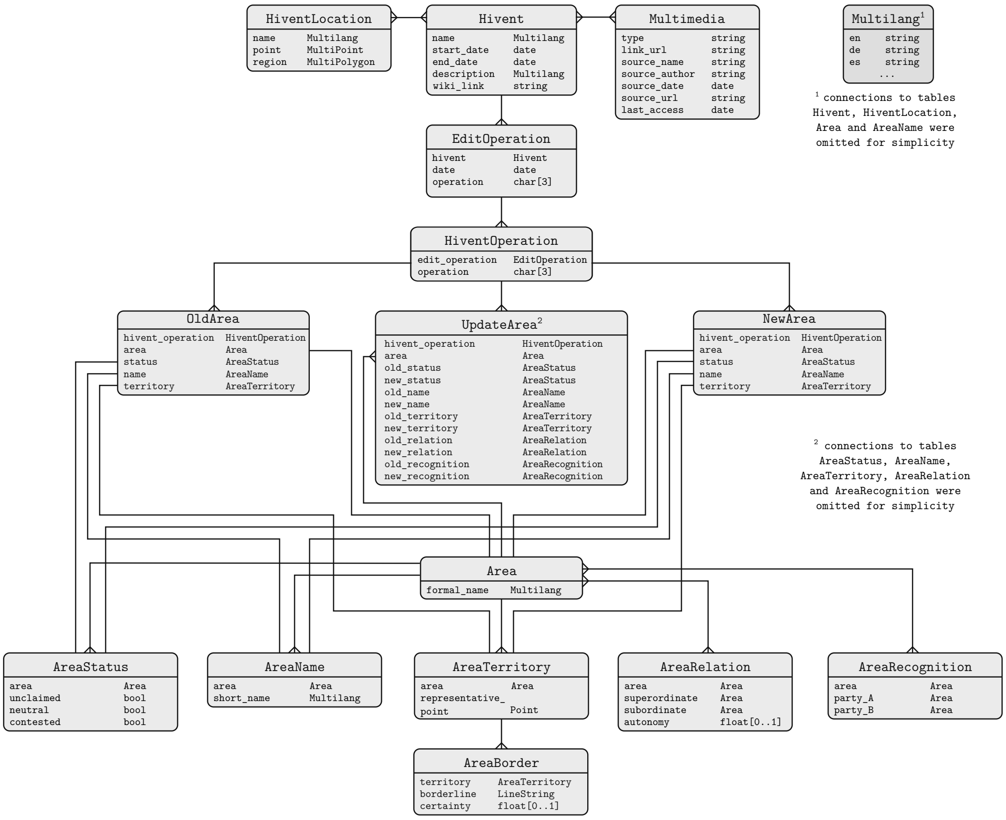


Figure 4.9: The updated database model to support several kinds of uncertainty

3. Creation of a Multimedia entity to manage multimedia files associated to a 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 an 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 way to cope with the inherent problem of uncertainty and disagreement in history. For these shortcomings, design solutions for the user interface of HistoGlobe were developed. They were integrated into the abstract Hivent model and the database model. The final chapter of this thesis summarizes the achievements of the current 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 a 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 territory and name of a country. 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 Ω 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, i.e. Unification, separation, incorporation, secession and name change.

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

These five changes are modeled in the *Hivent model*, an event-based spatio-temporal data model for vector data, organized by a four-domain model and visualized in a graph. *Hivents* – ***Hi**storical* `events` - are historically significant happenings in time. Countries are represented by abstract *Areas* with a formal name, a short name and a territory. They can change due to *Hivent operations* that represent exactly the five historical changes. Each operation deletes or creates a set of Areas or updates the properties of one Area. The history of Areas can be visualized spatially on a map and non-spatially on the *HistoGraph*.

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

While the Hivent operations are very well understood by a machine, they are not suitable to be used by humans to manually edit the course of history. For that matter, six *edit operations* are developed, i.e. create, merge, split, change border, rename and delete. An edit operation can be directly performed on the map using a workflow of four steps:

1. Select the Areas that will be changed in the operation.
2. Create a territory for each new Area resulting from the operation.
3. Create a name for each new Area.
4. Add the edit operation to a Hivent to inherit the time stamp.

Internally, the edit operations are expressed by a combination of the five Hivent operations and visualized on the HistoGraph. The Hivent model including the Hivent operations and the edit operations are the main contributions of this Master's thesis to the research field of spatio-temporal data models.

The model and the operations were implemented in HistoGlobe, a web-based Historical Geographic Information System that aims to visualize the history of the world on a map and a timeline. The user interface was extended by the edit mode to edit the historical data about Areas and Hivents in the system. In several user studies of the human-centered design process in this work the interface proved to be understandable and usable.

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

While the initially developed Hivent model works very well given the absolute certainty about the data in the system, it has no support for any kind of imperfection, known lack of accuracy and precision or disagreement. Therefore, extensions to the original Hivent model were developed.

Areas can be assigned a special status, e.g. being a contested territory, an autonomous country within a sovereign country or a neutral zone. This will visualize the Areas differently and serve as a visual clue to signal disagreement or uncertainty. The concept of international recognition is introduced to solve the problem how to deal with questionable "maybe" countries. If a country has no or limited recognition by other countries, it will not be visualized the same way. The borders of Areas get an additional property that signifies the *certainty* about their course. The lower the certainty, the blurrier or wider the border will appear on the map to signify limited expected accuracy.

These approaches reveal the actual problem of information systems dealing with imperfection: the user who is uncertain about the property of an object, needs to tell the information system *exactly how uncertain* they are. This is ironic and difficult.

5.2 Problems and Improvements

As it has already been examined in the evaluation in section 4.1, there are several problems with the Hivent model. It has currently no support for actors – to answer the question "who?" – and for actual locations – to say "where?". Both can easily be integrated and would allow new perspectives on historical events. The current implementation is only available in English. Internationalization is desired, but problematic, because in different cultures there are not just syntactically different translations, but semantically different concepts of the same historical event. This issue comes along with the desired support for different perspectives, based on different historical evidence. There is a lot of interdisciplinary research to be done in digital humanities how to support multiple perspectives on data.

The two basic assumptions about the territory of a country – constant coastlines and a lack of sea territory – are wrong. The Hivent model supports only sudden spatio-temporal changes due to events, but not due to gradual processes. The data model for HistoGlobe would need to be extended with a concept of *Geoprocesses* that model the long-term geographical processes that lead to changing coastlines. This is an entirely different research field that would need to be entered.

Given the current status of the implementation, there is plenty of room for improvement. First of all, the design approaches developed in the previous section have to be further developed and implemented into the final system to support various degrees of uncertainty. Additionally, the information visualization problems regarding the HistoGraph have to be solved to support it in the browsing and the edit mode of HistoGlobe. The tool for semi-automatic extraction of historical maps needs to be integrated into the New Name Tool to simplify the process of creating the territory of a historical country. It also needs to support importing existing geodata in various formats. To cope with potentially more data in the database, a more sophisticated client-server interaction for initializing and maintaining the state of the system needs to be developed.

However, the most crucial part is the implementation of the complicated concepts of retrospective updates and backward changes. Without them, HistoGlobe is still useless for editing the historical developments of countries in time and space. For that matter, the problem of retrospective updates needs to be further analyzed to identify possibilities for simplification. If the insertion of an update leads to more than 10 recursive updates that the user has to perform, this might be very frustrating. One idea would be to always give priority to the latest correction in case of a conflict that can be solved in two ways.

New visualization techniques can also be introduced. Animated transitions between two states of an Area would significantly increase the usability of the browsing mode. The attention of the user would be drawn to the territories that currently change. To properly account for the name of the software, a three-dimensional globe can be implemented to replace the map. This requires the use of the graphics card via *WebGL*, because rendering of and interaction with a globe is much more computationally intensive than with a map. A logarithmic version of the timeline would also be interesting. A user study could analyze whether it really suits the logarithmic nature of human perception better than a normal linear timeline.

Finally, to tackle a problem of digital humanities, the current model allows to trace *changes over time* and provide *context*. These are two of the foundations of history introduced in section 2.2.1. However, historians do not necessarily need a tool to better visualize existing knowledge, because digital historical maps, audio or video sources are sometimes sufficient to show their results. For their research purposes it is crucial to generate new knowledge to establish *causality*, another foundation for history. Therefor historians need to analyze spatio-temporal coherences or distributions in historical data by sorting, selecting or classifying the data in the system and running spatio-temporal queries. Spatio-temporal reasoning is still not easily possible with existing HGIS and yields many interesting research questions to be solved.

5.3 Prospect

Imagine the problems that were mentioned are solved and the room for improvement is filled with nicely designed user interface elements, then HistoGlobe would have the potential to be a well usable Historical Geographic Information System for the history of countries in time and space.

Imagine scholars in humanities would use the edit mode of HistoGlobe to continuously improve the historical data in the system, animate historical countries and discuss historical events. Imagine HistoGlobe could be used not just to answer *what* has happened, *where* and *when* did it happen and *who* participated – but to answer ultimate question *why* something is the way it is? What are the real causes for the current Syrian civil war? Despite the conflicts in the Middle East, do we live in a peaceful time? Why did the Roman Empire collapse?

It would help teachers to explain history to their students and learners to finally understand the seemingly complex course of history. If that would work, HistoGlobe would be a comprehensive historical world atlas with a great potential to teach, learn and understand processes in the past. We could learn from our mistakes we have done in the past and reason about the things that actually matter: To provide a world without the need for greed and hunger, for brotherhood of men. If John Lennon is right, then this information system would at some point come to its ultimate end: all countries unify to one world, the common universe – without borders. 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

[AB] Thomas Andrews and Flannery Burke. *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 [accessed 23.10.2015].

[Abe08] James S. Aber. *History of maps and cartography*, 2008. URL: http://academic.emporia.edu/aberjame/map/h_map/h_map.htm [accessed 24.10.2015].

[Arc] ArcGIS. *ArcGIS Help 10.1*. URL: http://resources.arcgis.com/EN/HELP/MAIN/10.1/index.html#//0031000000q8000000 [accessed 03.11.2015].

[Arm92] C. Armenakis. *Estimation and Organization of Spatio-Temporal Data*. *Proceedings of the Canadian Conference on GIS92*, 1992.

[BBCa] BBC World News. *Nagorno-Karabakh profile*. URL: http://www.bbc.com/news/world-europe-18270325 [accessed 26.04.2016].

[BBCb] BBC World News. *Russia recognises Georgian rebels*. URL: http://news.bbc.co.uk/2/hi/7582181.stm [accessed 26.04.2016].

[BBCc] BBC World News. *Somaliland profile*. URL: http://www.bbc.com/news/world-africa-14115069 [accessed 26.04.2016].

[Bol08] Paul Bolstad. GIS Fundamentals: A First Text on Geographic Information Systems. Eider Press, 3 edition, 2008.

[Bri] British Ordnance Survey. *Automatic change detection*. URL: https://www.ordnancesurvey.co.uk/education-research/research/automatic-change-detection.html [accessed 02.11.2015].

[Bus] Business Dictionary. URL: http://www.businessdictionary.com [accessed 04.11.2015].

[Can86] John F. Canny. *A Computational Approach to Edge Detection*. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[CGP] CGP Grey. *How Many Countries Are There?* URL: https://www.youtube.com/watch?v=4AivEQmfPpk [accessed 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:///citation.cfm?id=647207.719698.

[DISP08] Stanislas Dehaene, Véronique Izard, Elizabeth Spelke, and Pierre Pica. *Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigene 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 [accessed 05.11.2015].

[Esr] Esri. *Polygon*. URL: http://esri.github.io/geometry-api-java/doc/Polygon.html [accessed 30.10.2015].

[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 [accessed 26.04.2016].

[For] Former McKinsey Consultant. *What Does It Mean to be "MECE" at McKinsey?* URL: http://workingwithmckinsey.blogspot.de/2014/02/MECE-at-McKinsey.html [accessed 18.05.2016].

[Fra] Witold Fraczek. *Mean Sea Level, GPS, and the Geoid*. URL: http://www.esri.com/news/arcuser/0703/geoid1of3.html [accessed 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.

[Gab08] Eric Gaba. *Map projections*, 2008. URL: https://commons.wikimedia.org/wiki/User:Sting/Gallery:_Maps_of_the_world,_of_seas_and_about_history [accessed 28.10.2015].

[Gal68] Thomson Gale. *Enclaves and Exclaves*, 1968. URL: http://www.encyclopedia.com/doc/1G2-3045000363.html [accessed 02.06.2016].

[Gaz] Gazilion. *Explicative diagram of territorial discontinuities: Enclaves and exclaves*. URL: https://commons.wikimedia.org/wiki/File:Diagrama_enclave_exclave.svg [accessed 02.06.2016].

[Geo] Geokov. *Map projection - types and distortion*. URL: http://geokov.com/education/map-projection.aspx [accessed 27.10.2015].

[GG14] Ian N. Gregory and Alistar 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-rianovosti11.htm [accessed 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.

[Had15] Leon Hadar. *In Praise of 'Virtual States'*, 2015. URL: http://antiwar.com/hadar/?articleid=8042 [accessed 26.04.2016].

[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/abouttheministry/pages/israel-s%20diplomatic%20missions%20abroad.aspx#note [accessed 26.04.2016].

[Kau12] Kaushik. *The Curious Case of Baarle-Nassau and Baarle-Hertog*, 2012. URL: http://www.amusingplanet.com/2012/11/the-curious-case-of-baarle-nassau-and.html [accessed 02.06.206].

[KH08] Anne K. Knowles and Amy Hillier. Placing History: How Maps, Spatial Data, and GIS are Changing Historical Scholarship. ESRI Press, 2008.

[Kno02] Anne K. Knowles. Past Time, Past Place: GIS for History. ESRI Press, 2002.

[Lan88] Gail E. Langran. *A Framework for Temporal Geographic Information Systems*. *Cartographica*, 25, 1988.

[Lan89] Gail 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 [accessed 26.04.2016].

[Mü] 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 [accessed 18.05.2016].

[Nak] Frieder Nake. *Data, Information and Knowledge*. URL: http://viola.informatik.uni-bremen.de/typo/fileadmin/media/DataInformKnowledge.pdf [accessed 05.11.2015].

[Oba] Dare Obasanjo. *An Exploration of Object Oriented Database Management Systems*. URL: https://www.cs.umd.edu/class/spring2002/cmsc434-0101/MUIseum/applications/odms.html [accessed 14.05.2016].

[Ora] Oracle. *1970s — Defying Conventional Wisdom*. URL: http://www.oracle.com/us/corporate/profit/p27anniv-timeline-151918.pdf [accessed 14.05.2016].

[OS01] Thomas Ott and Frank 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.

[Oxf] Oxford English Dictionary. URL: http://www.oed.com/ [accessed 25.04.2016].

[Pal] Palestinian Central Bureau of Statistics. *Estimated Population in the Palestinian Territory*. URL: http://www.pcbs.gov.ps/Portals/_Rainbow/Documents/gover_e.htm [accessed 26.04.2016].

[PD95] Donna J. 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 [accessed 26.04.2016].

[Peu99] Donna 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, 2004. URL: http://dx.doi.org/10.1017/S026988890400013X.

[Ran15] Random House Dictionary, 2015. URL: http://dictionary.reference.com/ [accessed 23.10.2015].

[Raz12] Ale Raza. *Working with Spatio-Temporal Data Type*. *International Archives of the Photogrammetry, Remote Sensing and Sptial Information Sciences*, XXXiX-B2:5–10, 2012.

[Ren96] Agnar 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 [accessed 26.04.2016].

[Roy] Royal Geographical Society. *What is geography?* URL: http://www.rgs.org/geographytoday/what+is+geography.htm [accessed 23.10.2015].

[Sco] David Scofield. *Seoul's double-talk on reunification*. URL: http://www.atimes.com/atimes/Korea/GA04Dg01.html [accessed 26.04.2016].

[Sel] Amnon Sella. *Custodians and Redeemers: Israeli Leaders' Perceptions of Peace, 1967-79*. URL: http://www.jstor.org/stable/4283114 [accessed 26.04.2016].

[Sol14] Ana C. Solana. Spatio-Temporal Narratives: Historical GIS and the Study of Global Trading Networks (1500-1800). Cambridge Scholars Publisher, 2014.

[Tob70] Waldo Tobler. *A computer movie simulating urban growth in the Detroit region*. *Economic Geography*, 46 (2):234–240, 1970. URL: http://academic.emporia.edu/aberjame/map/h_map/h_map.htm.

[Tod] Today.az. *For Azerbaijan Pakistan does not recognize Armenia as a country*. URL: http://www.today.az/news/politics/30102.html [accessed 26.04.2016].

[Uni] United Nations. *Member States*. URL: http://www.un.org/en/member-states/index.html [accessed 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 [accessed 28.04.2016].

[Uot] Urho A. Uotila. *geoid — geology*. URL: http://www.britannica.com/science/geoid [accessed 27.10.2015].

[Vat] Vatican City State. *Population*. URL: http://www.vaticanstate.va/content/vaticanstate/en/stato-e-governo/note-generali/popolazione.html [accessed 26.04.2016].

[WDSK15] Bastian Weber, Nana Kesewaa Dankwa, Kiti Singh, and Tejeswini Kashyappan. *HiBo - semi-automatic extraction of borders from historical maps*, 2015. URL: https://bitbucket.org/bastian_weber/hibo [accessed 29.10.2015].

[Wes] Western Sahara Online. *Proclamation Of The Saharawi Arab Democratic Republic*. URL: https://wayback.archive.org/web/20080502035045/http://www.wsahara.net/sadr.html [accessed 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. *How many countries are there in the world?* URL: https://www.youtube.com/watch?v=4vsPB_lbo94 [accessed 26.04.2016].

[Yal] Yale Law School. *Convention on Rights and Duties of States*. URL: http://avalon.law.yale.edu/20th_century/intam03.asp [accessed 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/ [accessed 07.11.2015].

[Zwa] Vladimir Zwass. *Information System*. URL: http://www.britannica.com/topic/information-system [accessed 05.11.2015].