

# MASTER'S THESIS

Usability of Ampersand for designing register systems.

Edelaar, G

**Award date:**  
2022

[Link to publication](#)

## General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

## Take down policy

If you believe that this document breaches copyright please contact us at:

[pure-support@ou.nl](mailto:pure-support@ou.nl)

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 19. Nov. 2022

**Open Universiteit**  
[www.ou.nl](http://www.ou.nl)



# Usability of Ampersand for designing register systems.

by

**Gerard Edelaar**

Date of presentation: 14 June 2022





Title of thesis: Usability of Ampersand for designing register systems.  
Student name: Gerard Edelaar  
Date of presentation: 14 June 2022

Graduation committee: chair: prof. dr. ir. Stef Joosten  
primary supervisor: dr. Bastiaan Heeren

Degree programme: Open University of the Netherlands,  
Faculty of Management, Science and Technology  
Master's Programme in Software Engineering

Course code: IM9703

# Contents

1	Summary	1
1.1	Dutch	1
1.2	English	2
2	Introduction	3
3	Context	6
3.1	Design method	6
3.1.1	Relation Algebra	6
3.1.2	Ampersand	7
3.2	Reactive approach	10
3.3	Wet-BIG	11
3.3.1	Law tax analysis	12
3.4	Registers	12
4	Method	14
4.1	Action research	14
5	Results	20
5.1	Usefulness	20
5.2	Ampersand as method	27
5.3	Ampersand as tool	34
5.4	Design	38
5.5	Registry systems	43
5.6	Analysing law	44
6	Discussion	48
6.1	Ampersand knowledge	48
6.2	Ampersand core in wet BIG	52
6.3	Setup law for Ampersand	53
6.4	Ampersand for government organization	54
6.5	Limitations	59
7	Conclusions and Recommendation	60
7.1	Conclusions	60
7.2	Recommendation	67
8	Reflection	68
9	Acknowledgements	69
	Bibliography	70
	Non-published Readings	72
A	Observations	74
B	Interviews	88
C	Associated laws and regulations	94
D	Content Analysis	95
E	List of Functional Requirements	108
F	Adl scripts	109
G	Prototype	119
H	Technical Datamodel	125
I	Conceptual analysis	126

# 1. Summary

## 1.1. Dutch

In deze thesis gaan we de bruikbaarheid van Ampersand onderzoeken door het ontwerpen van een register systeem bij een overheidsorganisatie. Ampersand wordt niet breed gebruikt en we vragen ons af waarom dat het geval is.

Het **CIBG** is een uitvoeringsorganisatie van het Ministerie van Volksgezondheid, Welzijn en Sport. Deze organisatie, waar ik werkzaam ben, beheert register systemen. Een register systeem, ook wel register genoemd, heeft voor ons altijd een wettelijke basis. Dus op basis van een wet, creëert het **CIBG** een register. Om dit onderzoeken uit voeren hebben we een authentieke case genomen. Het systeem dat de **Wet-BIG** ondersteunt staat op nominatie om vervangen te worden. De gekozen methode om het onderzoek uit te voeren is dan ook **action research**.

Tijdens het ontwerp proces zijn observaties over het verloop van de Ampersand analyse vastgelegd. Ook zaken van Ampersand die opvallen zijn meegenomen. De **Conceptual analysis** die opgeleverd is en qua opzet besproken met geïnterviewde personen.

Alle observaties en interviews zijn middels content analyse gerubriceerd en hebben de basis gevormd voor de beantwoording van de hoofd- en subvragen. Deze hebben geleid tot het trekken van conclusies op de vraag of Ampersand bruikbaar is voor het ontwerpen van register systemen bij een overheidsorganisatie. Het is hierbij opgevallen dat register systemen niet anders zijn dan andere informatie systemen. Het grote verschil is dat een register systeem een wettelijk basis heeft en een informatie niet per definitie. De bruikbaarheid van Ampersand om een wet te analyseren is goed, echter moet een organisatie bereid zijn om het te gebruiken. Natuurlijk zijn er verbeteringen mogelijk en er kan nog onderzoek plaatsvinden naar het gebruik van tooling om ontwikkeling te versnellen.

## 1.2. English

In this thesis we will investigate the usefulness of Ampersand by designing a registry system at a government organization. Ampersand is not widely used and we wonder why that is the case.

The **CIBG** is an implementing organization of the Ministry of Health, Welfare and Sport. This organization, where I work, manages registry systems. A register system, also known as a register, always has a legal basis for us. So based on a law, the **CIBG** creates a register. To carry out these investigations, we have taken an authentic case. The system that supports the **Wet-BIG** is nominated to be replaced. The chosen method to carry out the research is **action research**.

During the design process, observations about the course of the Ampersand analysis were recorded. Also items of Ampersand that stand out are included. The **Conceptual analysis** that was delivered and discussed in terms of structure with interviewees.

All observations and interviews have been classified by means of content analysis and have formed the basis for answering the main and sub questions. These have led to conclusions about whether Ampersand can be used for designing registry systems in a government organization. It has been noticed that registry systems are no different from other information systems. The big difference is that a register system has a legal basis and an information not by definition. Ampersand's utility for analyzing a law is good, but an organization must be willing to use it. Of course improvements are possible and research can still be done on the use of tooling to accelerate development.

## 2. Introduction

Ampersand [Joosten, 2017] is used to specify registration systems. We will investigate the usefulness of Ampersand for designing registration systems. In this study, we want to determine if Ampersand is useful for government organizations that build and manage registration systems.

The usability of Ampersand is apparently a problem, reflected in its low usage. The tool and its method of use work in practice. This has been shown by the efforts made in other projects, nevertheless the method is not widely used. The question is why.

Ampersand is a rule-based design tool. It is a way to design information systems that comply with all business rules. This makes business rules sufficient as a tool to design registration systems that comply with these rules. A method and tool to create error-free specifications of the registration system to support business processes. Ampersand supports the way in which the design and **Conceptual analysis** are created. We use the tooling to translate the design into a prototype and **conceptual analysis** that comply with all business rules.

Ampersand is based on Relation Algebra [Maddux, 2006]. The most commonly used components of Ampersand's relation algebra are the Concept, the Relationship, and the Rules. A Concept is an abstract representation of immutable items. A Relationship defines a connection between two or more Concepts. The Rules deal with validating Relationships between Concepts.

We use the Ampersand method to translate laws and regulations. Legislation and regulations are strongly rule-oriented and because Ampersand is a rule-oriented design method, it is therefore ideally suited for legislation and regulations. We will investigate whether this law is also suitable for the Ampersand analysis method.

The design process within Ampersand focuses on the creation of a prototype and the **Conceptual analysis**. We analyze the source texts through which the prototype and design are formed. The source text in this case is the legislation and regulations. The analysis of the legislation and regulations results in a script that generates a prototype and **Conceptual analysis** using Ampersand. Ampersand is both the method followed and a tool to obtain the result. One of the research results is the **Conceptual analysis** in which we can see which Concepts, Relations and Rules this law provides us with.

Ampersand is going to be deployed at the **CIBG**, which is an executive agency of the Ministry of Health, Welfare and Sport and therefore a government organization. These organizations operate on the basis of laws and regulations. The CIBG is a government organization that designs, builds and manages registration systems. These registration systems, called registers, are always based on laws and regulations. The purpose of a registry is to provide reliable and accessible information so that it can be used for individual consultation and as research data as used in **Schmidt et al. [2015]** and **Bakken et al. [2019]**. The reliability of the data in the registry must be ensured by the management organization.

The cause of the low use of Ampersand for designing register systems is unknown. There is not enough information to determine the causes. To find out the



reason for the low usage, we will conduct an exploratory study. Therefore, the exploratory approach chosen is **Action Research (action research)** of **Easterbrook et al. [2008]**. The exploratory approach of **Easterbrook et al. [2008]** lends itself to the research using Ampersand to design register systems to derive hypotheses and construct theories.

The researcher is not completely independent of the case that will be investigated. We adopt the approach of **action research** because the researcher is closely related to the research case. Other arguments in support of the **action research** approach relate to the **CIBG**. The **CIBG**, the employer of the researcher, has an interest in the research and is particularly interested in the design of the registry. Designing the successor to the registry system Zorro, the registry system for the **Wet op de beroepen in de individuele gezondheidszorg (Wet-BIG)**, is a matter that is readily accessible for research. Too few reference cases are available to conduct quantitative research. Objective analysis is therefore not possible here. Therefore, it makes sense to investigate the usefulness of Ampersand by actively participating in the design process and curiously examining what Ampersand does in that process.

In practice, we see that Ampersand is not that widely used. In recent years, experiments have been conducted at TNO, KPN, Bank MeesPierson, ING-Bank, Rabobank and Delta Lloyd. They have experimentally confirmed the method and provided insight into its practicality.

Despite the fact that Ampersand works in practice, it is little used. A possible cause of the current low usage may be the unfamiliarity of Ampersand. Which produces circular reasoning: Unknown means it is rarely used, when it is rarely used, it remains unknown. The popular products in the Open Source market are affiliated with a large organization that can push marketing and knowledge. For example: operating systems like Ubuntu which is maintained by Canonical Ltd. In addition, such an organization can also build a community to support the product. The usefulness of a product like Ampersand says nothing about its use. Because it is not always the case that a useful product is widely used. In the IT world, this happens with the Linux desktop versus a Windows desktop. It contains all the functionalities needed to work with and has a functional user interface. But many users still choose a Windows desktop. Windows apparently has features that are considered indispensable. Windows was also a forerunner in usability and has a great marketing department. Despite the user-friendliness of Linux, it is mainly chosen by IT people or users who want a free operating system.

The research techniques used are limited to interviews with stakeholders and an analysis of the material collected during the study. By showing stakeholders what Ampersand means to the registry, statements about the usefulness of the method can be elicited here. To show what Ampersand delivers, we will also build an environment using the Ampersand method. During the build, we will collect observations about issues we encounter.

Building the environment as a software engineer provides data on the usability of Ampersand. By classifying this data, we gain quantitative insight into the

points that stand out the most. The numbers provided do not necessarily say anything about usability, but they do say something about the issues the researcher encountered.

Software engineers can use Ampersand to design and prototype registration systems. What knowledge does an **Software Engineer** need to perform this analysis task and design with Ampersand. They are used to designing and developing in a programming language. Prototyping is often part of their development task. Ampersand is both a design tool and a development method. In addition, Ampersand provides the ability to have it evaluated via a prototype.

The stakeholders are able to make a statement about the usefulness of Ampersand for the use of registry system within a government organization. Because they have a relationship with the current registry system called Zorro. From that relationship they make statements about the Concepts and Relationships within the generated **Conceptual analysis**. They look at the usability of Ampersand in designing and developing systems, at the incorporation into the architecture and at the possibilities of using the **Conceptual analysis** and the prototype. Based on the observations and interviews, we make a swot analysis of Ampersand for the **CIBG** organization.

There have been previous studies in which Ampersand plays an important role. This has not encouraged the use of Ampersand to date. The work of **Baecke [2018]** looked at Argument assistance software in legal reasoning. There, Ampersand is used to build a prototype to support professionals in legal reasoning and not for the automation of argumentation.

In the dissertation of Pim Bos [**Bos, 2013**], this author investigates the feasibility of implementing the business rules of the VOG case using semantic web technologies. This work compares Ampersand with Semantic Web Rule Language at the level of relation algebra.

### 3. Context

The ampersand sign <sup>1</sup> and the Ampersand method both emphasize the meaning "and self-contained" (see figure 1). On the website of Ampersand <sup>2</sup> we find an interpretation of the statement "standalone (op zichzelfstaand)".

There are more case studies conducted in the past about the usefulness of Ampersand. Like the graduation study of [koopman \[2014\]](#) and the example study of [Baecke \[2018\]](#) about adapting Ampersand in legal environment. Also in the field of legislation by for example the UWV. With this case study we want to see if this is also possible for legislation coming from the Ministry of Health, Welfare and Sport.

For knowledge of Ampersand, the books by [Wedemeijer et al. \[2013b\]](#) and [Wedemeijer et al. \[2013a\]](#) from the Open University are available.

#### 3.1. Design method

We use Ampersand's method in this study. The Ampersand method is based on relation algebra. Ampersand makes it possible to create a Conceptual analysis of the source text. The conceptual and technical data model created with the descriptions in the Conceptual analysis allows the implementation to be based on it.

##### Relation Algebra

The field of relational algebra focuses on operations on sets. The characteristic item of relation algebra is the relation. This relation has its attributes. The attributes in the example of [Wet-BIG](#) would be the person's name, first names, gender, date of birth, nationality, and address, as well as the number and time of enrollment <sup>3</sup>. The relation consists of tuples. Since the relationship is always between two objects, one speaks of 2-tuples. The tuples contain the attributes of the relation.

De naam *ampersand* komt uit de woorden "et per se &" voort, gebruikt aan het einde van alfabetreeksen, hetgeen betekent: "en op zichzelfstaand &"; ook de volgorde "& per se et" werd gebruikt; de & volgt dan direct op de andere alfabettekens en zo speelde men met de dubbelzinnigheid: "& dat in zichzelf *et* betekent" of "en op zichzelfstaand *et*". Dit werd door de Engelsen aan de achterkant van hun alfabetrijmpjes voor kinderen geplakt in de vorm *and per se and*, wat weer tot "ampersand" verbasterd werd.



Figure 1: [www.contented.nl/wat-weet-jij-van-het-en-teken-de-ampersand](http://www.contented.nl/wat-weet-jij-van-het-en-teken-de-ampersand)

<sup>1</sup><https://www.contented.nl/wat-weet-je-van-het-en-teken-de-ampersand>

<sup>2</sup><https://ampersandtarski.gitbook.io/documentation/why-ampersand/business-rules-in-ampersand>

<sup>3</sup>[Wet op de beroepen in de individuele gezondheidszorg](#) article 3, paragraph 2

The operations on sets are the following:

- Union,  $R \cup S$
- Intersection,  $R \cap S$
- Difference,  $R - S$  or  $S - R$

A distinction must be made between relation algebra [Maddux, 2006] and relational algebra [Codd, 1970]. Ampersand uses relation algebra, so it is tuple related. The relational algebra is the basis of e.g. relational databases which includes projections, selections and joins. The latter are therefore not part of relation algebra.

### Ampersand

Ampersand is based on relation algebra and focuses on business rules [Wedemeijer et al., 2013a]. It supplies correct information systems. Our goal is to use it to create a correct and error-free registration system. Ampersand's other strengths are its support for conceptual analysis. It is a platform for reactive programming and generates prototypes. Ampersand script describes the goals rather than the steps.

Business rules are there to pursue a common goal. These rules are converted to an information system using the Ampersand method. The Ampersand method ensures that when a precise set of rules has been established, an information system can be generated. To learn how Ampersand works in real life, we design a registry in Ampersand that implements the Wet-BIG [van, 1993-2021] .

The principle of rule-based **Business process Management (BPM)** as mentioned in [Joosten and Joosten, 2007] is that any violation of a business rule may be used to trigger actions. This is described in the section **Reactive approach**.

Ampersand consists of concepts that in turn consist of atoms. An atom is an implementation of the concept. Inside the **Wet-BIG** is a concept *beroep* with associated atoms like *arts*, *tandarts*, etc" see listing 1. We give the concepts a name so that the Concepts are recognized by the company. This also applies to the definition and purpose of the terms. These attributes are not mandatory, but when one wants to generate a functional design, these descriptions of the attributes are very useful.

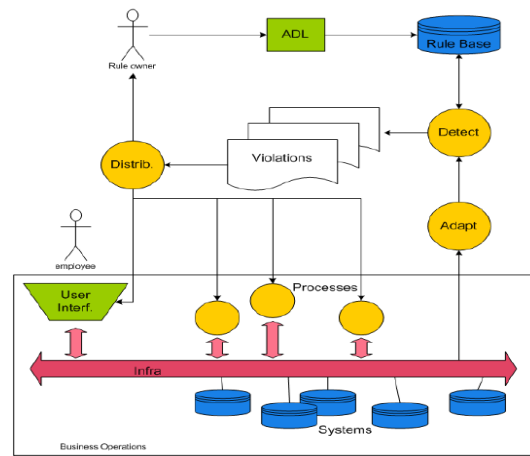


Figure 2: rule-based-proces

1 CONCEPT Beroep "Beroep van een persoon zoals bedoeld in de wet"  
 2 PURPOSE CONCEPT Beroep  
 3 {+Beroep dat uitgeoefend wordt+}  
 4 POPULATION Beroep CONTAINS [

```

5  "arts",
6  "tandarts",
7  "apotheker",
8  "gezondheidszorgpsycholoog",
9  "psychotherapeut",
10 "fysiotherapeut",
11 "verloskundige",
12 "verpleegkundige",
13 "physician assistant",
14 "orthopedagoog-generalist"
15 ]

```

Listing 1: Listing Concept Beroep

Concepts can have relationships with each other. If the data of the concepts is true and the rules yield consistent data, then the relationships between real data are facts. These facts together form one truth. Not all concepts are directly related. Within the domain of the **Wet-BIG** we could distinguish the concept *registratie* and the concept *beroep*. These terms come from **van [1993-2021]** in article 3 of **Wet-BIG**. Even the name of the relationship is mentioned in this article, which the legislator calls a practitioner. The law requires that details of the *registratie* be recorded, stating the corresponding profession. In Ampersand this is modeled as follows. On the one hand the *beroep* and also the concept *registratie*, see listing 2.

```

1  CONCEPT Registratie "De registratie van een persoon binnen het register"
2  PURPOSE CONCEPT Registratie
3  {+Vastlegging in het register geeft toegang tot uitvoeren taak binnen de gezondheidszorg+}

```

Listing 2: Listing Concept Registratie

Between the *registratie* and the *persoon* exists the relationship *beroepsbeoefenaar*, see listing 3.

```

1  RELATION beroepsbeoefenaar [Persoon*Registratie]
2  MEANING "geregistreerd persoon"
3  POPULATION beroepsbeoefenaar CONTAINS
4  [
5  ("Piet",1);
6  ("Susan",2);
7  ("Gerard",3);
8  ("John",4)
9  ]

```

Listing 3: Listing RELATION "beroepsbeoefenaar"

By adding the concepts of *persoon* (see listing 4) and *handeling* (see listing 5), people may perform medical actions, but only when they are qualified.

```

1  CONCEPT Persoon "Persoon die werkzaam wilt zijn binnen de zorg"
2  PURPOSE CONCEPT Persoon
3  {+Vastleggen van de identiteit van de persoon+}

```

Listing 4: Listing Concept Persoon

```

1  CONCEPT Handeling "Acties die uitgevoerd worden"
2  PURPOSE CONCEPT Handeling
3  {+Vastleggen van de mogelijke handelingen die uitgevoerd kunnen worden binnen de zorg+}

```

Listing 5: Listing Concept Handeling

These concepts can lead us to the following scheme.

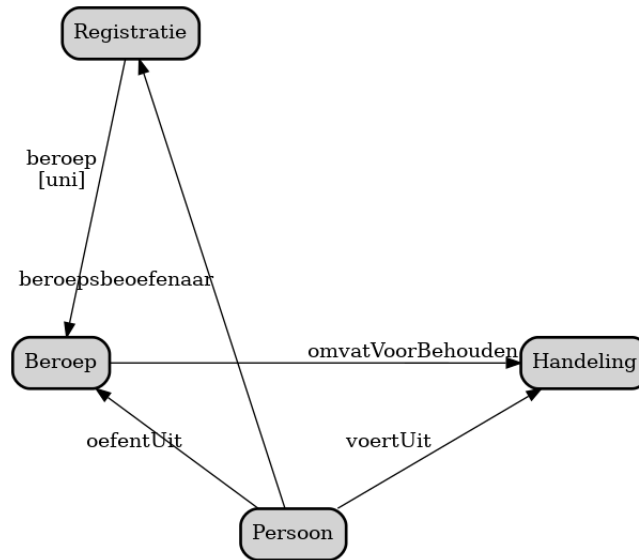


Figure 3: relations

The multiplicity must also be determined for each relation.

Table 1: multiplicity

function	The corresponding control question for the above relation <i>voerUit</i> is
Univalent	For each <i>Persoon</i> there is at most one <i>Handeling</i>
Total	For each <i>Persoon</i> there is at least one <i>Handeling</i>
Injective	For each <i>Handeling</i> there is only one <i>Persoon</i>
Surjection	For each <i>Handeling</i> there is at least one <i>Persoon</i>

Modeling using the Ampersand method determines which Concepts and Relationships arise within the research case. Ampersand helps to gain insight into these connections. This must be recognized by the analyst in the source text and defined in the script. Ampersand then helps generate the functional design (the **conceptual analysis**) and the prototype. Using the generated prototype, Ampersand validates the data with constraints on relationships. This prevents registrations (data) that do not meet the requirements. These restrictions are defined in rules within Ampersand (see for example list 6). A rule can be drawn up that determines whether a person is allowed to perform a certain action. In figure 3 the relations are named. It was previously established that there are 2-tuple relationships. Here we use the following notation: "*relation*[*Concept* × *Concept*]".

$$\begin{aligned}
 &voertUit[Persoon \times Handeling] ; omvatVoorBehouden[Beroep \times Handeling] \sim \\
 &\subseteq \\
 &beroepsbeoefenaar[persoon \times registratie] ; beroep[registratie \times beroep]
 \end{aligned}$$

The compared sets are

[*Persoon* × *Beroep*]

The rule then will determine if the previous equation is true.

If this is the case, then the rule is validated, otherwise the violation message OCCURS.

```

1  RULE HandelingDoorPersoon: voertUit; omvatVoorBehouden[Beroep*Handeling]- |·- beroepsbeoefenaar; beroep
2  MEANING "Een persoon mag handelingen uitvoeren wanneer hij een bepaald beroep uitoefend"
3  MESSAGE "Geen toegestane handeling."
4  VIOLATION (TXT "Persoon ", SRC I, TXT " voert de handeling uit ", TGT I, TXT " die niet tot zijn beroep behoren ", SRC I[Person];
   oefentUit)

```

Listing 6: Listing Rule HandelingDoorPersoon

### 3.2. Reactive approach

One of the benefits of Ampersand <sup>4</sup> mentioned is the following statement "Use Ampersand as a platform for reactive programming, to help you of workflows and workflow models."

This reactive approach started with the reactive manifest [rea, 2014]. The approach defines the requirements that a reactive system should meet. These include Responsive, Resilient, Elastic and Message Driven. These requirements lead to systems that are flexible, loosely coupled and scalable, making them easier to develop and maintain. Reactive Systems are made highly responsive and provide interactive feedback.

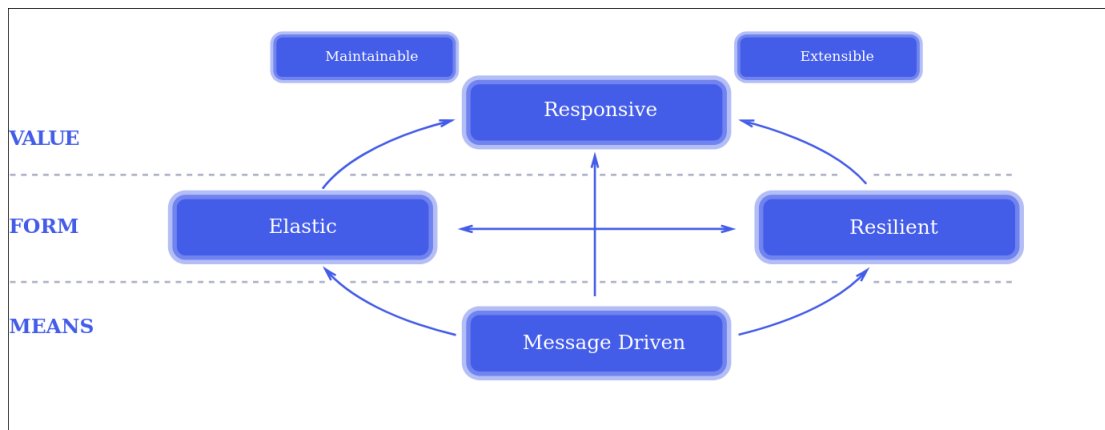


Figure 4: reactive manifesto

Ampersand is a form of Functional Reactive Programming (FRP) [Elliott and Hudak, 1997]. The basic of reactive programming is the fact that it involves asynchronous communication. The Reactive Manifest (see [rea, 2014]) states that like uses message-driven systems, but Ampersand is more than a message-driven system. This means that, as the Reactive Manifest prescribes, it uses message-driven systems, but Ampersand is more than a message-driven system. It is actually an event-driven system. The glossary of the rea [2014] indicates the difference between message driven systems and event driven systems. An event-driven system targets event-bus while a message-driven system targets recipients [Bainomugisha et al., 2013]. The essence is that the order of the flow cannot be determined in advance. The system will respond to events caused by constraints. Ampersand determines the dynamic flow [Joosten, 2018].

<sup>4</sup><https://ampersandtarski.gitbook.io/documentation/why-ampersand>

### 3.3. Wet-BIG

During the graduation project, research was conducted into the suitability of Ampersand for designing registers for the government. These registers are always based on legislation and regulations. The research focuses on a specific law, namely the **Wet-BIG**. The **CIBG** is the executing party for the **Wet-BIG**.

The first health care law was enacted in 1865. This law, together with eleven other laws, forms the basis of the **Wet op de beroepen in de individuele gezondheidszorg (Wet-BIG)** <sup>5</sup>. The **Wet-BIG [van, 1993-2021]** replaces the ban on medical action by unauthorized persons by granting responsibilities to healthcare providers. The professions regulated in Article 3 of the **Wet-BIG**, which include doctors, nurses and physiotherapists, have a compulsory registration with periodic re-registration, a statutory disciplinary law and a protected professional title. The former paramedical professions that are now regulated in Article 34 of the **Wet-BIG** have no registration obligation and no legally regulated disciplinary law. These Article 34 professions are only certified.

Then we have the **Wet-BIG**. It describes the following:

de tot dusverre geldende wettelijke regeling op het gebied van de uitoefening van de geneeskunst, inhoudende een het gehele gebied der geneeskunst bestrijkend verbod van beroepsuitoefening zonder hiertoe wettelijk verleende bevoegdheid, te vervangen door een regeling welke een ruimer gebied van individuele gezondheidszorg bestrijkt en waarbij slechts het verrichten van bij de wet aangewezen categorieën van handelingen wordt voorbehouden aan categorieën van daartoe overeenkomstig de wet gekwalificeerden, terwijl het voeren van wettelijk beschermde beroepstitels uitsluitend toekomt aan degenen die in de voor de desbetreffende beroepen overeenkomstig de wet ingestelde registers ingeschreven staan en ten aanzien van andere beroepen op het gebied van de individuele gezondheidszorg voorzien wordt in de mogelijkheid tot het regelen van de opleiding tot die beroepen;  
voor onderscheidene categorieën van overeenkomstig de wet gekwalificeerden een aan de gebleken behoeften aangepaste regeling van tuchtrechtspraak in het leven te roepen;

This law consists of 148 articles, of which a number are still pending. The articles are also regularly updated. More recently, in July 2020 there were still amendments to the law.

On the website [wetten.overheid.nl](https://wetten.overheid.nl) <sup>6</sup> are the Dutch laws including the **Wet-BIG**. Within this website, the content is kept up to date. All changes to the law are traceable. On this website a user can find any law in any given period of time.

This law does not stand alone. Appendix C contains an overview of the laws and regulations that relate to the **Wet-BIG**.

<sup>5</sup>[https://nl.wikipedia.org/wiki/Wet\\_op\\_de\\_beroepen\\_in\\_de\\_individuele\\_gezondheidszorg](https://nl.wikipedia.org/wiki/Wet_op_de_beroepen_in_de_individuele_gezondheidszorg)

<sup>6</sup><https://wetten.overheid.nl/BWBR0006251/2020-07-01>



When we talk about the new **Wet-BIG**, we can say that it is not there yet. Proposals have been written about a new law [**Bussemaker, 2019**], but it has not yet been enacted into law, and it is also a descriptive text and not an article-by-article summary.

### Law tax analysis

In the Netherlands, the tax authorities have also devised a method for analyzing laws. The tax authorities have developed a method [**Ausems et al., 2021**] that is intended to analyse tax laws and other laws. This is performed in these 6 steps:

1. Determining the work area.
2. Making the structure visible in legislation.
3. Defining the meaning of legislation.
4. Validate the analysis results.
5. Identify missing execution policy.
6. Setting up the knowledge model.

Emphasis is placed on the cooperation between the implementer, ICT and policy. By going through the method step by step, one arrives at a shared language. This shared language includes the definition of concepts by the collaborating parties. An important part of the approach is dividing the law into small pieces and always refer to these pieces of law in the implementation. As a result, the method meets the requirement of the justification of government decisions. The decisions are traceable, explainable, and it is possible to account for them. What is not clear from the webinar **Belastingdienst [2021]** is how these steps were converted into an implementation. The book **Ausems et al. [2021]** indicates that the legal analysis method does not contain a development tool, but that the Tax and Customs Administration has developed an instrument based on the legal model, which is not freely available.

### 3.4. Registers

We are investigating the usefulness of Ampersand for registry systems. Register systems are also known as registers. The current **Wet-BIG** is housed in a registry system.

The current system developed to support the **Wet-BIG** still has the project name Zorro. This stands for **ZOrgverlener Registratie Requirements Ontwikkeling**. This system was developed in 2008 as a successor to the Ribiz system. Zorro is a Microsoft.net (C#) application running on a windows platform with an underlying MS-SQL server. The architecture of the system is based on an internal workflow, but continuous design changes over the years has caused a maintenance issue that requires new construction for this system.

The **Application Lifecycle Management (ALM)** advice [de Kok et al., 2019] has shown that in addition to a maintenance problem, there are also issues in the field of security, outdated architecture and process support. In consultation with the policy directorate (beleidsdirectie) that is responsible for **Wet-BIG**, the CIBG has embarked on a process for new construction. The **ALM** advice has been there since 2019. Preparations for the new building have started, but construction has not yet started.

As mentioned, the current **BIG-registration System** is built as a workflow system. The idea behind this was that when adding a new profession within the **Wet-BIG** only a new professional title should be added. Practice proves to be more unruly, and numerous exceptions have been made within the software for trajectories within the professions and specializations. What does make the software complex is the integration of disciplinary law within **BIG-registration System**. Also, the support of the trajectory of foreign persons who want to work in the healthcare field. This interweaving has made the program great. The current **Software Improvement Group (SIG)** rebuild calculation has estimated it at 27 man-years.

Zorro is divided in several building blocks. Building blocks related to the persons called **health care person (HCP)**, concerning the workflow called Case and creating files called Dossier. The case building block has its focus on the process of registration. Looking at the main data model blocks, one can see that it involves metadata, product, state and of course the case and its requests and activities. On the other hand, the dossier is only about physical documents that are scanned and archived. Technically, this is solved via a SharePoint solution. Also, an outdated solution for archiving documents. The most important building block is about the people who want to be registered.

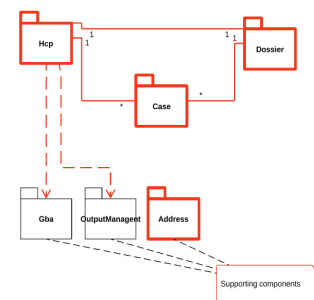


Figure 5: overview zorro

During the design, they clearly looked at what the **Wet-BIG** has in it. This is also reflected in the current data model. One sees in this data model the interdependence with the "tucht" process. It's called Judgment and JudgmentProvisionNote. All signs of disciplinary action. The physical implementation of this data model is much more complex. This is due to the inclusion of foreigners in the system.

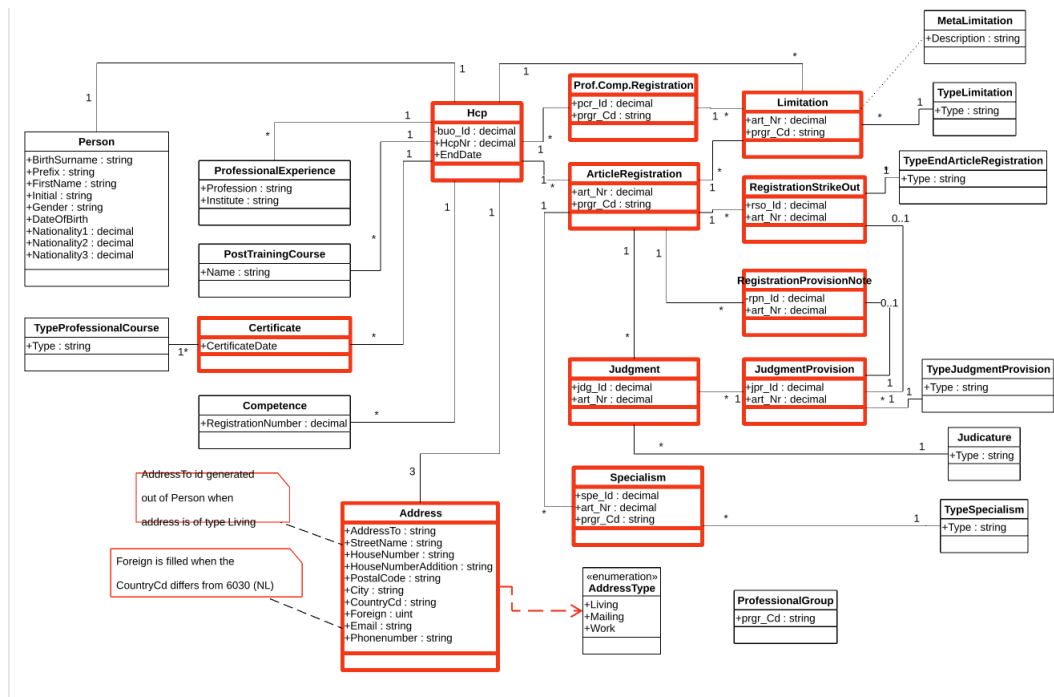


Figure 6: HealthCarePerson

## 4. Method

### 4.1. Action research

Action research investigates the usefulness of Ampersand. It has already been stated in the introduction that the use of Ampersand is lagging behind in practice. That is why we test Ampersand in practice on the usability of the method and of the tool.

The Ampersand's usability is measured along several axes. The definition of usability according to Shackel [2009] is "the capability to be used by humans easily and effectively" and "the effectiveness, efficiency, and satisfaction with which specified users can achieve goals in particular environments" is the definition given by the NEN<sup>7</sup>. The usability attributes according to Hornbæk [2006] are measured via Effectiveness, efficiency and satisfaction. Usability with which the NEN measures are the Appropriateness recognisability, Learnability, Operability, User error protection, Accessibility and User interface aesthetics.

Exploratory research is being conducted to measure the usefulness of Ampersand and according to Easterbrook et al. [2008] used as initial investigations of some phenomena to derive new hypotheses and build theories. The new hypotheses to be developed relate to the usability of Ampersand.

The development of the hypothesis takes place by measuring the usability attributes. To measure these attributes we use action research[Easterbrook et al., 2008]. The core of this research method is the action-oriented approach. The researcher is part of the organization in which the research takes place. The

<sup>7</sup><https://www.nen.nl/nen-iso-iec-25010-2011-en-157265>

data is collected within this organization. It is set up **action research** as a joint learning process of researcher and organization. the problem owner is willing to collaborate to both identify a problem, and engage in an effort to solve it. The original problem is authentic. There is widespread discussion about the methodology, and even debate on the validity of action research as an empirical method according **Easterbrook et al. [2008]** because of the appearance of ad-hoc.

The action consists of making a prototype, because that offers many opportunities to explore the usefulness of Ampersand. Prototyping requires a case on which the prototype can be based. The Ampersand method is used to build the prototype.

The case being executed relates to the replacement of the current registry system of the **Wet-BIG**. The **Wet-BIG** is supported within the **CIBG** by the registry system **ZOrgverlener Registratie Requirements Ontwikkeling (BIG-registration System)**. The **BIG-registration System** is deprecated. Based on an **Application Lifecycle Management (ALM)** recommendation (2019)[**de Kok et al., 2019**], it has been determined that the system is no longer adequate from the perspectives of security, maintenance, finances, functionality and process support. Within the **CIBG** initiatives have been started to replace the current system.

We are looking into the usability of the Ampersand. The action we perform is to design a prototype of the **BIG-registration System**. The **BIG-registration System** is managed by the **CIBG**. The **CIBG** is a government organization of the **Ministry of Health, Wellbeing and Sports**. This registration system is based on the **Wet-BIG**. Thus above leads to the following research question:

**How useful is Ampersand for designing registry systems by analysing public health legislation and regulations, in particular the **Wet-BIG**.**

In order to investigate the usability of Ampersand by making the prototype, it is necessary to acquire knowledge to be able to implement this.

To do this part of the research, we formulate the following sub-question:

*RQ1 - What knowledge, in the role of software engineer, is needed to use Ampersand.*

While making the prototype, we will work with Ampersand. In addition to registering observations, we are also interested in the results of the campaign. Are the Concepts, Relations and Rules found recognizable for the organization? Especially because the **CIBG** now also has a **BIG-registration System** running. We formulate the following sub-question:

*RQ2 - What are the Concepts, Relationships and Rules in the **Wet-BIG**.*

The prototype is based on the **Wet-BIG**. The Ampersand method prescribes that, in this case, we should take the law as our starting point. Now this is also the case with a traditional design. But then there is an interpretation battle over the user representation. At Ampersand, the law is taken literally as a guideline. The prototype is made on the basis of the law.

That leads us to the following sub-question:

RQ3 - *How are the laws and regulations set up so that they can be used in a useful way for the Ampersand method.*

In order to estimate the usability for the receiving organization, it is necessary to look at what the usability attributes mean for the organization. The usefulness of the Ampersand method is not only technical, but organizational. There are also less rational aspects in this area. To get to the bottom of this, we will determine the strengths of using Ampersand as a registration system for the CIBG. And also what the weaknesses of using Ampersand are.

We formulate the following sub-question:

RQ4 - *What are the strengths and weaknesses (SWOT) in using Ampersand for registry systems for a government organization.*

While making the prototype, we collect data. Then we collect two types of data. On the one hand, during the process of making the prototype, collecting observations. Writing down everything we encounter along the way. Each observation is given a number and a date stamp (see appendix A). A direct allocation is made to the sub-questions. This is based on the assumption that the observations can lead to answering the sub-questions. The other source of information is obtained from a number of interviews. Summaries are made of these interviews (see appendix B). We will carry out both processes time-boxed.

The chosen approach of elaboration relates to the content analysis [Kohlbacher, 2006]. The steps described are also the steps followed within this action research. During these steps we collect evidence according to the approach of Kohlbacher [2006]. Studying the legal texts and converting them in part to Ampersand. During this conversion, it was always recorded which observations had been made. The next step is to analyse case study evidence. This includes examining, categorizing and combining data. There are several approaches like Hsieh and Shannon [2005] and Mayring [2019]. Where the outcomes include relying on theoretical propositions, thinking about rival explanations of developing a case description. Last step is reporting phase, fulfilled in section 7.

Based on research question, we cluster according to its terms The clustering is to:

Table 2: Category definitions

<b>Category</b>	<b>Definition</b>
Useful	Ampersand can help you to achieve what you want.
Ampersand as method	The way of working with Ampersand.
Design	An activity or process that identifies the requirements and then comes up with a solution capable of meeting the requirements.
Registry systems	Using scripting and target generation.
Analysing law	Systems with which data is recorded, whereby the definitions of subjects to be registered are used by all supplying persons and bodies.
Ampersand as tool	The discipline and profession concerned with the customs, practices, and rules of conduct of a community that are recognized as binding by the community
Not-categorised	

The purpose of content analysis is to validate the claim that we can determine the cause of Ampersand’s low usage. Research will provide predictions about variables mentioned in table 2. Using a directed content analysis [Hsieh and Shannon, 2005] approach, is more organized than using a conventional approach. Where Hsieh and Shannon [2005] calls it a directed content analysis, for Mayring [2019] this is a deductive category application. The procedure is deductive because the category system is established before coding the text. The categories are deduced from keywords of the main question. A definition has been drawn up for each category that the category must meet according table **Category definitions**. Based on the research question, the data and the objectives of the researcher, the following strategy can be followed in labeling/coding. The strategy starts with labeling using the predefined codes see table 2. Theoretical considerations can lead to a further categories or rephrasing of categories, but the categories are not developed out of the text material like in inductive category formation.

The existing theory is that Ampersand is very suitable for application in legislation and regulations and also for use within a government organization. The impression is that the unfamiliarity in particular stands in the way of the use and usability of Ampersand. The data collected will show whether there is more than obscurity here. That is also the way the data is classified and treated. The content analysis performed leads to a number of claims about the usability of Ampersand. These claims are the result of the investigation.

We collected information about things that stood out while building the proto-

type. This information is included as observations in appendix A and via interviews in appendix B.

We then work with pre-formulated categories, which are derived from the main question and establish the relationship between the categories and the observations. This step consists of a methodologically controlled assignment of the category to a text passage of the observation.

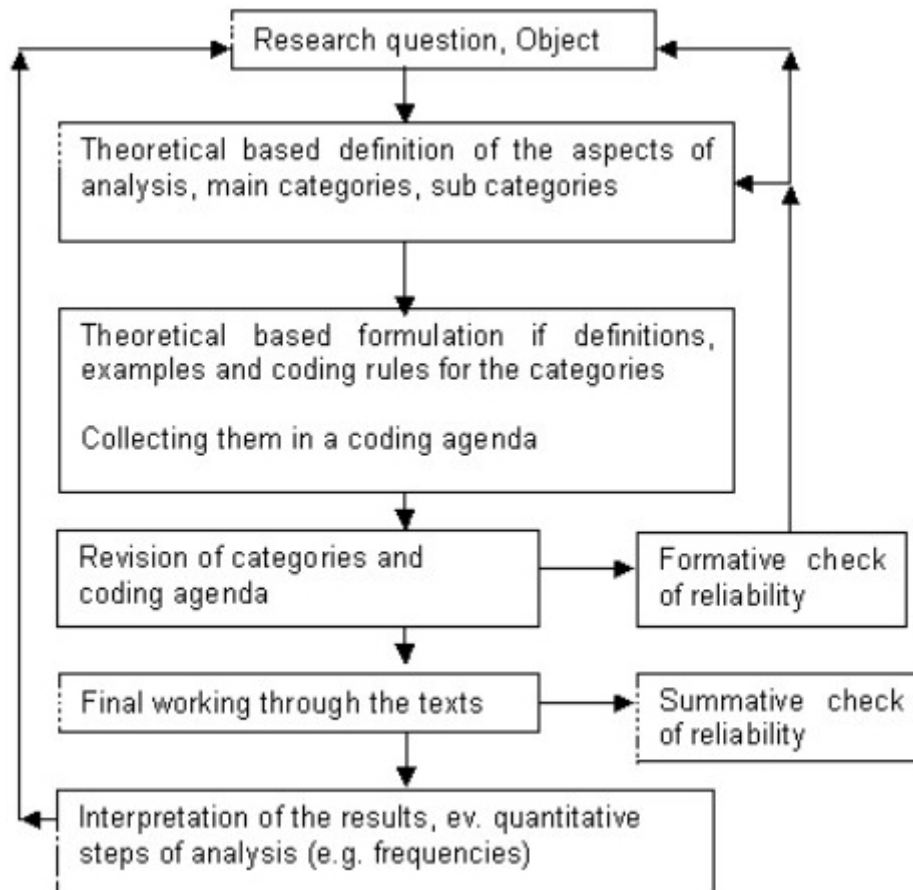


Figure 7: Step model of deductive category application[Mayring, 2000]

The starting point is that the usability of Ampersand is an issue. We see this because Ampersand is rarely used. We hypothesize that we can determine the cause of the low usage through research. The research therefore focuses on the main question.

While making the prototype, we record what we notice. The record is tagged with the date and time of the observation. So that the observation remains traceable and we hold discussions with stakeholders in the form of free interviews.

We determine the coding rules for each deductive category and provide an example. We then work through the observations and determine to which category the text belongs. Here we use our chosen identification of observation. The interview reports are divided into paragraphs and label it with an identifier. Then

we determine to which category the paragraph belongs. By means of a loop-back the theoretical basis can be further refined as can be seen in figure 7.

Philip Mayring is also the founder of the website <https://www.qcamap.org>. The content analysis tool on the said website allows us to perform part of the content analysis. We define a project within QCAMap, the working method is linked to this project. For our research the deductive approach.

After assigning the data to the category, it appears that further refinement is required within the categories. In the section **Results** the results are logically linked so that we can also label the refined clustering.

In section 6 we refer to the categorization and the refined labeling. But here we assign these to the sub-questions that we formulated earlier. These sub-questions are answered in this section. To arrive at the section 7.1 to answer the main question.

The tool to be used to validate is triangulation [[Carter et al., 2014](#); [Farquhar et al., 2020](#); [Runeson and Höst, 2008](#)]. Triangulation offers the possibility to view the source from multiple perspectives. The perspective of the law itself. In addition, the engineers have the necessary knowledge and information about the application of the law and from the business perspective. This is a way of assuring the validity of research through the use of a variety of methods to collect data on the same topic, which involves different types of samples as well as methods of data collection<sup>8</sup>.

---

<sup>8</sup>[https://en.wikipedia.org/wiki/Triangulation\\_\(social\\_science\)](https://en.wikipedia.org/wiki/Triangulation_(social_science))



## 5. Results

In addition to the **Wet-BIG**, the investigation into the suitability of Ampersand also covers the list of regulations (see list **C**). Due to time-boxing it is not possible to analyze and process all associated laws and regulations. The aim is not to provide a fully elaborated conceptual analysis, but to test Ampersand for usability.

Notable findings relate to the setup of Ampersand, the setup of the documentation, maintaining the overview and setting up a team working on the analysis.

In the following paragraphs we discuss the results obtained. In section **4** it is indicated that we perform the content analysis according to **Mayring [2000]**. From that perspective, there is a table per category that contains the category-id, category title and definition (see **2**). The table also includes one or more examples of observations or interview parts that relate to the category. The coding rules are a testing instrument for the observations and interview parts. When an item meets the coding rules, it can be placed in the relevant category.

In the next section, section **6** the interpretations of the results will be discussed.

### 5.1. Usefulness

When asked whether Ampersand can be used, different axes can be considered. This can be measured along the line of **Hornbæk [2006]**, which looks at effectiveness, efficiency and satisfaction. The other line is the line prescribed by ISO <sup>9</sup> and then it is about Appropriateness recognisability, learnability, operability, user error protection, accessibility and user interface aesthetics.

Table 3: Category **Useful**

Category	CAT1-1
Category Title	<b>Useful</b>
Definition	<b>Ampersand can help you to achieve what you want.</b>
Anchor examples	<ul style="list-style-type: none"><li>• <b>rq1-17 Applying a rules takes a lot of patience and practice. This is quite a steep learning curve</b></li><li>• <b>rq1-46:24-10: There is no find able relationship between the relation and the concept in the script</b></li></ul>
Coding rules	is it suitable, is it adaptable, has it relevance, has it value

<sup>9</sup><https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=1&start=4>

## Setup (Useful)

**Used observations**(see appendix D and B for full text)

- obs.47: rq1-11 Implementation in Docker with RAP creates new directories all the time
- obs.2: rq1-18 Can not find an example on the internet, only in the repo of Ampersand itself. That is difficult to find
- obs.1: rq1-13:17-10: The setup of Ampersand in local environment is specific and not self-explanatory. Help is needed here to get this working. Attempts to get the process working in localhost were unsuccessful. The manual on the Ampersand site showed how to do this. But it still didn't work
- obs.102: rq1-36:29-9: Failed to run prototype under localhost in Windows10. The service would not start in localhost. We did manage to get the service running within Docker. There was an error in the installation documentation. Turns out that is was not the installation directory RapInstall, but the directory RAP
- obs.5: rq1-47:27-10: Detecting a bug. Placing these in github issues at the Ampersand repository will get a response within a day and resolve it. In this case it was a bug in Ampersand that was quickly fixed with a new version
- obs.48: rq1-6:21-10: Docker is also another thing to learn. There should also be an introductory course to quickly understand Docker usage for Ampersand. A waste of time to have to look this up yourself or it is preconditions to be able to use Ampersand
- obs.11: rq4-8:22-11: The team behind Ampersand is very dedicated

To use Ampersand, a number of conditions must be met. In practice, the Ampersand setup is not always smooth and error-free (obs.5, obs.1, int.35, obs.47, obs.102). We can install Ampersand as a development tool in several ways. These ways of installing is on localhost, or using the RAP <sup>10</sup> environment or installing Ampersand within a Docker <sup>11</sup> environment (obs.1). Docker installation, with help from the Ampersand team, has proven to be the most successful (obs.11). Note that knowledge of Ampersand also requires basic knowledge of Docker (obs.48). This help was also needed because there is not much information on the internet (obs.2).

## Script creation (Useful)

**Used observations**(see appendix D and B for full text)

- obs.6: rq1-60:9-11: Training and education is required to write an Ampersand script
- obs.7: rq1-96:30-12: Skill in scripting within Ampersand is quickly lost if you don't do this frequently
- obs.119: rq1-17 Applying a rules takes a lot of patience and practice. This is quite a steep learning curve

To make a useful Ampersand script, it is necessary to gain knowledge of how to create scripts (obs.6) and requires quite some knowledge of Ampersand and relation algebra. Concepts, Relations and Rules are used within the scripts. It should be noted that making Rules in particular is a difficult task (obs.119). This knowledge of Rules and also other knowledge of Ampersand must be kept up to

<sup>10</sup>Student development environment of Ampersand(<https://rap.cs.ou.nl/page/home>)

<sup>11</sup><https://www.docker.com/>

date, otherwise it will disappear very quickly (obs.7).

### Source handling (Useful)

**Used observations**(see appendix D and B for full text)

- obs.4: rq1-2 Ampersand has no annotation option, therefore requires a separate action or document to keep track of what has been passed
- obs.85: rq1-25:12-9: First make overview of all laws and regulations
- obs.97: rq1-31:14-9: Besides XML and JSON, RTF and PDF are also an option. In rtf (doc) you can add items in the margins via "comments". With a PDF, annotations and color highlighting can be given this feature
- int.7: I-3.7,The law consists of the following parts. Going through the law should be a first step for the conceptual analysis

To maintain an overview at work, resources are needed to maintain this overview (obs.85, int.7). An overview is needed to keep track of where people have left off in the text of the law (obs.4, obs.97). An overview of the created scripts is also necessary to avoid duplication of Concepts and Relations. Due to the lack of an overview and the refactoring of scripts, possible duplication's arise. These are reflected in the conceptual design by multiple display.

### Script overview (Useful)

**Used observations**(see appendix D and B for full text)

- obs.25: rq1-42:19-10: Immediately add the description when recording a concept and relation. Later it is difficult to find out why the recording took place
- obs.3: rq1-45:24-10: Overview within an Ampersand script is difficult to obtain
- obs.27: rq1-46:24-10: There is no find able relationship between the relation and the concept in the script
- obs.52: rq1-97:30-12: By puzzling with Ampersand people quickly forget to make correct documentation. Often you are happy that something works

In addition to the necessary overview at work, it also requires self-discipline to maintain the script properly (obs.25). Ampersand requires that the description, meaning and purpose also be established during the development of Concepts, Relations and Rules. Due to inexperience with Ampersand it happens that meaning and purpose are not captured (obs.52). As the scripts grow, the need for overview increases (obs.3) as more and more Concepts and Relationships are added (obs.27).

### Data add (Useful)

**Used observations**(see appendix D and B for full text)

- obs.54: rq3-7 Adding documentation with the correct description to a concept and relation is not so easy. Easy to stray and add your own interpretation
- int.2: I-3.2,A draft of the conceptual analysis is available and this is experienced as trusted by the lawyer. In fact, these are recognizable texts because they have been taken directly from the law

Adding the information to the Ampersand concepts is not very self-evident (obs.54). It is a matter of searching the legal text for the correct phrase. Taking

into account the final layout in the Conceptual analysis. The Conceptual analysis must be clearly legible (int.2).

### Deviation (Useful)

**Used observations**(see appendix D and B for full text)

- obs.76: rq1-42:21-10: It is easy to deviate from the legal texts. Because they are so hard to read. Some knowledge of the law or the process means that your own interpretation is quickly made. Action research also means that you quickly fall into this trap

Noting the meaning and purpose requires knowledge of being able to read legal texts (obs.76). Knowledge of the information domain seems to be an advantage, but can cause prejudice. This means that the text is less carefully looked at.

### Architecture and registerkern (Useful)

**Used observations**(see appendix D and B for full text)

- obs.19: rq1-62:10-11: There has be the architecture link between the law core and the register core
- obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses
- int.17: I-2.10,Ampersand's approach is in line with the Registerkern, but not at the implementation level. It doesn't seem possible to implement Ampersand directly, but the analysis seems quite useful for extending Registerkern. Where generality is discovered and for the specific parts of the law. Then we are talking about a conceptual link and not a technical one
- int.21: I-1.3,The Registerkern, an architecture model of CIBG, uses shared concepts. With this it has similarities with Ampersand. There is also an overlap of Ampersand with Registerkern. Within Ampersand are concepts that are also in Registerkern. Registerkern is a defining part of the architecture. Other parts will have to conform to this architecture

The design of the architecture partly determines the usefulness of Ampersand (obs.19), despite its flexibility by extends concepts and relationships (obj.55). In particular, the overlap of what Ampersand contributes with the existing structure seems to be a bottleneck (int.17, int.21).

## Api (Useful)

**Used observations**(see appendix D and B for full text)

- int.9: I-2.2, Ampersand has APIs and that is interesting to be able to link with. Whether that can also be linked with Registerkern is not clear at the moment
- obs.17: rq4-2 The api link works fine, but entire messages return. These should actually get codes
- obs.18: rq4-5 Postman used for api link with Ampersand
- obs.13: rq1-70:14-11: Postman works with api/v1/resource, e.g. GET localhost/api/v1/resource/Person/P001/Person, retrieves that of an existing person. So the validation structure of ampersand can be used from outside Ampersand by means of api
- obs.14: rq1-72:14-11: Besides the GET(get), the POST(append) and PUT (mutate) also work
- obs.15: rq1-73:14-11: Ampersand can be used from other applications through APIs, but the return values are next to the requested information also messages and not message codes. These codes could be included in the reports, but now remain "unstructured" data
- obs.16: rq1-74:16-11: Link between an external front-end and an Ampersand back-end (Ampersandapi). A change in the back-end, so an Ampersand change, then the front end almost certainly has to change with it
- obs.12: rq1-8:14-11: No swagger is created for the api;

The availability of APIs ensures that Ampersand can be used from outside (obj.18). In an organization, frameworks and tools are usually available that work with APIs (int.9). Tooling such as Postman also works with APIs and makes an Ampersand model externally testable (obj.13, obs.14, orefobs:rq1-74:16-11). One point is that the API model is not documented, for example in Swagger (obj.12). Another point is that no response code is returned after calling an API, but a text that is defined in the script (obj.17, obj.15).

## Model maintenance (Useful)

**Used observations**(see appendix D and B for full text)

- obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses
- obs.10: rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)
- int.38: I-4.8,How is the maintenance of the system? A new model is always made with the help of Ampersand. The data will have to be migrated itself. Ampersand does not support that. Usually the data structure is taken into account in advance so that as little conversion as possible has to take place. This means that a system is getting bigger and less manageable. So the strength of Ampersand is that this is prevented because a new core system is always being built and the effort is in the data conversion and the connection of adjacent systems
- int.12: I-2.5,When maintenance takes place on the model, how do we get from one model to another. So how does the IST go to SOLL situation. Ampersand is always creating a new model. So when the law is changed and a new model is needed as a result, Ampersand will produce a completely new model. As a result, no technical debt will remain in the model. It is always a new model. However, the challenge will be in the data migration from the old to the new model

After a change in the law, there is a change in the Ampersand model (obj.10). There is no possibility to implement a modified model and simply transfer the data to the new model (int.12). Ampersand's new model stands alone and has no physical relationship with the originally implemented model (int.38). It is very easy to extend and modify an existing model (obj.55), Ampersand is very flexible in that regard.

## Ampersand design method (Useful)

**Used observations**(see appendix D and B for full text)

- obs.46: rq2-18:16-11: Good to realize that the meaning you write down also ends up in the Conceptual analysis. So looking at the way of writing it down can form a story in the analysis
- int.36: I-4.6,Ampersand method is a way of writing things down. That is not necessarily better or worse than any other method. So when something is being written down, so analysis is being done, why not with this. More is possible with it than with a Word document. The output is good to use and the structure too

A project always has to be designed. One of the interviewees commented that it doesn't matter much in which tool that happens (int.36). Many tools work from a model, model driven development [Kulkarni and Reddy, 2008], after which code is generated. Ampersand is declarative and generates workable and reliable software from the declaration and generates documentation with associated models (obj.46).

## Law effective (Useful)

**Used observations**(see appendix D and B for full text)

- int.20: I-1.2,Ampersand can be interesting, because it will be able to clear conflicting matters from the law. By performing the analysis, these will show up in the analysis. This makes it a resource to use before the law is enacted
- int.24: I-1.6,In addition, the implementation must be such that the effective dates of the specific amendments to the law are also taken into account. For example, at the time of an application, it is decisive whether the processing will take place in accordance with the old situation or the new situation

Because the analysis takes place at the source, it is possible to find conflicting matters in the law (int.20). In the current system **BIG-registration System** it is possible to specify a date when a change in law takes effect. The interviewee wonders whether this functionality can also be set via Ampersand (int.24)

## Excluded (Useful)

**Used observations**(see appendix D and B for full text)

- **int.18:** I-2.11,An addition of Ampersand is that a prototype is made that can also be tested. This allows the entire system to be tested because this combination must comply with validation from the law
- **int.41:** I-4.11,The question is whether the system will only work for simple registers or whether we can also use it to tackle complex registers. The Wet-BIG is complex, but not fully analyzed either
- **int.42:** I-4.12,A follow-up study could be to make a comparison between a system built traditionally and a system built on the Ampersand method. It is expected that due to code generation and being closer to the law, the amount of code will be a lot less. And with that also a better SIG qualification
- **int.32:** I-4.2,The prototype shown is not easy for the user to understand. The user not only looks at the functionality, but also at the design. The current design does not comply with the national government web guidelines. The question is whether the user will be able to see through this. It was not part of the research, but it was stated that adjusting the CSS could bring closer to the web guidelines
- **int.33:** I-4.3,Ampersand's deployment could be applied to new tasks. These have no history and can be built from scratch using the Ampersand method
- **int.39:** I-4.9,The learning curve doesn't seem that big. Even less technical people can work with this. With the adjustment in the styling, a prototype can be quickly made with which a working system can be demonstrated. On the other hand, only the conceptual analysis can be used. Based on this analysis, test scenarios can be devised and executed
- **obs.120:** rq1-39:3-10: Do not forget to create delete rules in addition to append and edit rules in the rules in the context of the Lifecycle approach
- **obs.110:** rq1-7:10-11: Each relation is part of a record structure
- **obs.69:** rq2-12:19-10: TOT has the property that this must be entered in the interface because otherwise the data will not be saved. A variant of this is a rule with this property As a result, the other items are stored in the database, but a notification of incompleteness continues to appear
- **obs.101:** rq2-16:19-10/11-11: Ampersand has a hard time determining a period. Ampersand cannot calculate out of the box. This requires the php functions, which are also not easy to allocate
- **obs.45:** rq2-4:30-9: Which agreements must be made regarding the structure of the descriptions for Conceptual analysis. Do agreements have to be made about it or leave it unstructured?
- **obs.81:** rq3-16:24-10: For the Netherlands, we have a country table from the RvIG. These are nationally established and maintained tables. No maintenance function is therefore required

The following observations have not been explicitly included.

## 5.2. Ampersand as method

The way of working with Ampersand requires preparation in the design and structuring of the work. Experience plays a major role in this. In the section on usefulness (see 5.1) it was pointed out to maintain overview. Keeping an overview helps when there are agreements about the naming of the Concepts, Relations and Rules.



Table 4: Category **Ampersand as method**

Category	CAT1-2
Category Title	<b>Ampersand as method</b>
Definition	<b>The way of working with Ampersand.</b>
Anchor examples	<ul style="list-style-type: none"> <li>• <b>rq1-25:12-9: First make overview of all laws and regulations</b></li> <li>• <b>rq1-80:20-11: A consistent naming of a concept is necessary</b></li> </ul>
Coding rules	Approach and working with Ampersand

### **Naming (Ampersand as method)**

<p><b>Used observations</b>(see appendix D and B for full text)</p> <ul style="list-style-type: none"> <li>• <b>obs.24: rq1-30:12-9: Defining the meaning and definition of the concept is free of rules. There is no fixed pattern for documentation</b></li> <li>• <b>obs.93: rq1-33:14-9: The use of patterns within Ampersand is important. These are the subsystems of the information system. The question is whether this should be classified in advance or whether it builds up on its own</b></li> <li>• <b>obs.95: rq1-38:3-10: Should the subsystems be mapped in advance</b></li> <li>• <b>obs.27: rq1-46:24-10: There is no find able relationship between the relation and the concept in the script</b></li> <li>• <b>obs.28: rq1-80:20-11: A consistent naming of a concept is necessary</b></li> <li>• <b>obs.45: rq2-4:30-9: Which agreements must be made regarding the structure of the descriptions for Conceptual analysis. Do agreements have to be made about it or leave it unstructured?</b></li> </ul>
--

Having consistent naming conventions (obs.24, obs.45) in the scripts and thus in the conceptual design is important for readability (obs.28). Working with Includes and Patterns to get small delineated bits of functionality is important for overview (obs.93, obs.95). The agreements are also necessary to find concepts and relationships in the scripting. Agreements are needed to be able to find Concepts and Relations in the script (obs.27).

## Multiplicity (Ampersand as method)

**Used observations**(see appendix D and B for full text)

- obs.88: rq1-3 Created a separate excel to write out and discover the multiplicity of the relations
- obs.89: rq2-2 Only UNI, TOT, INJ and SUR are used
- obs.70: rq1-21:7-11: TOT is usually overcome by a tot-rule, it turns out that a TOT causes something to be saved when entered, while a tot-rule allows a save to occur while the notification remains open to stand
- obs.63: rq1-58:8-11: Per interface max one multiplicity, otherwise you won't get data stored
- obs.111: rq2-11:19-10: An relation that is univalent is a function. A one function there can only come out one thing. The description of UNI is therefore  $P \rightarrow 0-1 H$  at most (see 2-5)
- obs.69: rq2-12:19-10: TOT has the property that this must be entered in the interface because otherwise the data will not be saved. A variant of this is an rule with this property As a result, the other items are stored in the database, but a notification of incompleteness continues to appear
- obs.90: rq2-13:19-10: What applies to multiplicity TOT, also applies to SUR
- obs.87: rq2-5:2-10: Making the multiplicity explicit

In order to maintain an overview of the relations, an Excel sheet(see 8) was drawn up in which the relations could be compiled (obs.87, obs.88, obs.111). This was to have support in the allocation of the multiplicities and was it possible to copy the relation definition to a script. Noticed that only the UNI, TOT, INJ and SUR are used, where the UNI is mainly used and the TOT is often solved via a Rule. The invariant violation prevents the prototype from storing data when multiple invariant violations occur simultaneously and that is when multiple TOT or SUR constraints are used (obs.70, obs.69, obs.90, obs.63).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Pattern	RELATION	RELATION	CONCEPT1	CONCEPT2	UNI	TOT	INI	SUR							
Personen	RELATION voornaam [Persoon*Voornaam][UNI,TOT]	voornaam	Persoon	VoorNaam	x	x					UNI,TOT	UNI	TOT		
Personen	RELATION geslacht [Persoon*Geslacht][UNI,TOT,SUR]	geslacht	Persoon	Geslacht	x		x				UNI,TOT,SUR	UNI	TOT		SUR
Personen	RELATION nationaliteit [Persoon*Nationaliteit][TOT]	nationaliteit	Persoon	Nationaliteit		x					TOT	TOT			
Personen	RELATION adres [Persoon*Adres][TOT,SUR]	adres	Persoon	Adres			x	x			TOT,SUR	TOT			SUR
Personen	RELATION inschrijftijd [Persoon*inschrijftijd][UNI,TOT,SUR]	inschrijftijd	Persoon	inschrijftijd	x	x		x			UNI,TOT,SUR	UNI	TOT		SUR
Personen	RELATION geboortedatum [Persoon*Geboortedatum][UNI,TOT]	geboortedatum	Persoon	Geboortedatum	x	x					UNI,TOT	UNI	TOT		
Register	RELATION register [Registerid*Register][UNI,TOT,INI]	register	Registerid	Register	x	x	x				UNI,TOT,INI	UNI	TOT		INI
Register	RELATION begintdatum [Registerid*RegisterBeginDatum][UNI,TOT,SUR]	begintdatum	Registerid	RegisterBeginDatum	x	x		x			UNI,TOT,SUR	UNI	TOT		SUR
Register	RELATION einddatum [Registerid*RegisterEndDatum][UNI,SUR]	einddatum	Registerid	RegisterEndDatum	x			x			UNI,SUR	UNI			SUR
Registratie	RELATION registratie [Registratie*inschrijving][UNI,TOT,INI]	registratie	Registratie	inschrijving	x	x	x				UNI,TOT,INI	UNI	TOT		INI
SpecialistenRegister	RELATION specialisme [Persoon*SpecialistenRegister][UNI]	specialisme	Persoon	SpecialistenRegister	x						UNI	UNI			
SpecialistenRegister	RELATION specialistDoorhaling [Persoon*Doorhaling][UNI]	specialistDoorhaling	Persoon	Doorhaling	x						UNI	UNI			
ArtsRegister	RELATION specialisme [Registerid*Specialisme][UNI]	specialisme	Registerid	Specialisme											

Figure 8: excel relation overview

## Rules (Ampersand as method)

**Used observations**(see appendix D and B for full text)

- obs.116: rq1-4 Automatically executed rule are easy to describe, but implementation here also takes a lot of patience and trying
- obs.124: rq1-61:9-11: There should be a check on the draft date of birth(rule), so that someone must be at least 18. Sounds logical, but is a derived rule. This is already implicit in the training requirement. The duration of the training means that someone is at least 18 years old before the training is completed
- obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses
- obs.121: rq1-67:11-11: If there is an automatic rule, should there still be a validation rule on it?
- obs.8: rq2-19:16-11: Ampersand returns constraints and no executable
- obs.122: rq2-6:2-10/13-11: A rule is not easy to realize. There are tricks to realize this. rq1-17 Applying a rule requires a lot of patience and practice

The method Ampersand uses involves applying rules to relationships. In concept the rules are easy to define, but in the Ampersand script a bit more difficult to construct (obs.116, obs.122). It requires knowledge of Relation algebra to work with this correctly and it requires knowledge of Ampersand because it requires a certain way of notation (obs.55). Sometimes it seems logical to implement a rule, but there are implicit restrictions that make it unnecessary (obs.124, obs.121). In the appendix F are examples of how to use Rules (obs.8).

## Concept reuse (Ampersand as method)

**Used observations**(see appendix D and B for full text)

- obs.23: rq1-79:20-11: Once a concept for a date or other element is defined, it can be used anywhere in the context. The question then is how to deal with shared Concepts and how to manage them
- obs.31: rq1-91:14-12: A concept and a relation can be defined several times within your own patterns. So that the patterns can stand on their own

The method allows multiple uses of the same Concepts and Relations (obs.31, obs.23) This is only visible when the documentation is generated. By allowing the Concepts and Relationships multiple times, it is possible to add different definitions to them.

## Team (Ampersand as method)

**Used observations**(see appendix D and B for full text)

- int.22: I-1.4,The danger of using legislation and regulations is that there is a possible incomplete picture of the concepts. This by adopting the rules one-on-one, without the interpretations. More laws are also used in an analysis than just the Wet-BIG. The question is how far is the analysis of the various laws going
- int.7: I-3.7,The law consists of the following parts. Going through the law should be a first step for the conceptual analysis
- int.43: I-4.13,To start with, a team should be set up to deal with this. This team of lawyers and analysts should be doing the analysis of a law and have it built
- obs.25: rq1-42:19-10: Immediately add the description when recording a concept and relation. Later it is difficult to find out why the recording took place
- obs.44: rq1-51:2-11: Discussing the Conceptual analysis should be done theme by theme
- obs.29: rq1-84:30-11: A concept is immutable. for example a person is concept, not doctor. It must be an intrinsic property, which cannot be changed

In addition to the technical approach, there is also a working method in which the user is closely involved. In this case, it is important that a lawyer is involved at the start of the work to get a picture of the law (int.7, int.22). Because Ampersand's approach to the definition is not very technical, a lawyer can easily understand this (obs.29). The **conceptual analysis** is also understandable for the lawyer. It is therefore important to go through this theme (pattern) by theme (obs.44) An analyst, as mentioned earlier, can perform this together with a lawyer (int.43). When processing it should not be forgotten that this information that is extracted from the **Wet-BIG** is also directly processed in the script (obs.25)

## Prototype use (Ampersand as method)

**Used observations**(see appendix D and B for full text)

- **int.18:** I-2.11,An addition of Ampersand is that a prototype is made that can also be tested. This allows the entire system to be tested because this combination must comply with validation from the law
- **int.1:** I-3.1,For a lawyer, the IT environment is an unfamiliar environment. The lawyer actually wants to form a picture of the system. In particular, what it looks like and what it can do. This while we want to involve the lawyer at the beginning of the process. Especially when we don't have the system yet
- **int.2:** I-3.2,A draft of the conceptual analysis is available and this is experienced as trusted by the lawyer. In fact, these are recognizable texts because they have been taken directly from the law
- **int.32:** I-4.2,The prototype shown is not easy for the user to understand. The user not only looks at the functionality, but also at the design. The current design does not comply with the national government web guidelines. The question is whether the user will be able to see through this. It was not part of the research, but it was stated that adjusting the CSS could bring closer to the web guidelines
- **obs.126:** rq1-57-1:7-11: Using Ampersand for validation
- **int.26:** I-1.8,Ampersand's possible positioning is to use it as an interpreter of legislation and regulations. Then maintain the current analysis and development process and use the prototype to validate the analysis. The question is whether this approach will not result in additional work compared to the current working method. There is a certain skepticism towards Ampersand

We validate the data using the prototype and the **Conceptual analysis** (int.26). During the research there was limited substantive validation, because the research focuses on the process and to a lesser extent on a validated design. The conceptual analysis is labeled as understandable, recognizable and useful (int.1, int.2). The prototype can be used to test the registry for the application of the law (int.18). Test cases can also be developed early in the development process on the basis of **conceptual analysis**. The prototype was looked at through ICT glasses (obs.126) and it turned out that it does not meet the **CIBG** requirements for a website. The tutorials indicate that this is possible by means of a CSS adjustment, but how exactly this should be done is not entirely clear and is also outside the scope of this research (int.32).

## Organisation Ampersand use (Ampersand as method)

### Used observations(see appendix D and B for full text)

- int.19: I-1.1,The assumption made by the interviewee that Ampersand is a tool that performs an interpretation on the law itself, is not correct. A manual stroke has to be done over the text of the law to recognize the concepts and relationships. This is seen as an intensive action
- int.27: I-1.9,Ampersand relies on facts and not on processes. While a practitioner is strongly process oriented. For example, the law does indicate that a diploma is required and also which type, but not exactly which diploma. So the law tells you what to do, but in most cases not how
- int.40: I-4.10,The Ampersand approach is different from most products. Most workbenches work from a drawn model and from there generate code from the documentation or possibly. Ampersand does this from a script and generates the models and documentation itself
- int.44: I-4.14,One could also only use the output of the analysis to build a system. Multiple scenarios are possible

Ampersand's acknowledgment as instigator of real-time signal violations is not reflected in the interviews. It is expected that Ampersand can be used in the preliminary phase of the design (int.44). There was also the expectation that Ampersand would be able to independently process texts into scripts (int.19) Within the organization people are used to working according to fixed processes, the reactive approach of Ampersand is therefore not always immediately understood (int.27). Ampersand's method focuses on text analysis and from there on models (int.40) as opposed to methods that work from the models.

## Excluded (Ampersand as method)

### Used observations(see appendix D and B for full text)

- int.35: I-4.5,For use, the question is how quickly a base is set up. It may be difficult to get to a 100% model. It may also be okay if this covers an 80% charge. LCSH as new project could be a good candidate
- int.39: I-4.9,The learning curve doesn't seem that big. Even less technical people can work with this. With the adjustment in the styling, a prototype can be quickly made with which a working system can be demonstrated. On the other hand, only the conceptual analysis can be used. Based on this analysis, test scenarios can be devised and executed
- obs.117: rq1-55:2-11: At the rule it is necessary to add a ROLE with a MAINTAINS, otherwise the rule will not work
- obs.110: rq1-7:10-11: Each relation is part of a record structure
- obs.59: rq1-90:14-12: Collection model of regulations than by means of includes keep it small and therefore clear. This is for the reusability of the script. One module per feature
- obs.112: rq2-10:19-10: The naming of a relation is usually assigned to the TRG attribute of the set. Such as [Persoon \* Voornaam] with relation name "voornaam"
- obs.123: rq2-14:19-10: The role gives control to the user. A user is authorized for use. It indicates which user is allowed to use the function
- obs.115: rq2-8:7-10/10-10: Date of birth must be formatted as date. The represent seems to have to fulfill that role. Represent defines a type of a concept, but DATETIME causes interface problems

The following observations have not been explicitly included.

### 5.3. Ampersand as tool

To view Ampersand as a tool, we look more closely at the technical functioning of Ampersand.

Table 5: Category **Ampersand as tool**

Category	CAT1-6
Category Title	<b>Ampersand as tool</b>
Definition	The discipline and profession concerned with the customs, practices, and rules of conduct of a community that are recognized as binding by the community
Anchor examples	<ul style="list-style-type: none"> <li>• rq1-49:30-10: Isolating a pattern or subsystem for testing does not work. This has to do with setting up Docker and possible ignorance on my part</li> <li>• rq1-53:2-11: The crud (Create, Read, Use, Delete) and CRUD in the interface don't always work as it should be. There is no full validation on usage. So an on/off does not make sense everywhere. rq1-37:3-10: CRUD/crud options also need some study before they can be applied properly</li> </ul>
Coding rules	Technical operation of Ampersand

## Includes (Ampersand as tool)

**Used observations**(see appendix D and B for full text)

- obs.57: rq1-15:4-10 With include statements the order of the contents of the document is determined. The expectation was that includes are needed to link parts of code together but includes are not everywhere necessary to get the code working
- obs.56: rq1-81:20-11: Compilation error due to a include that no longer existed. Observation here is that an adl has been renamed or moved or deleted. The tool Visual Studio Code does not support a refactoring stroke on said changes
- obs.58: rq1-82:20-11: include don't always seem necessary on compilation. It is not entirely clear when this is necessary or not. Another function of includes is to format the analysis
- obs.59: rq1-90:14-12: Collection model of regulations than by means of includes keep it small and therefore clear. This is for the reusability of the script. One module per feature

When the script is ready for compilation, removing Includes appears to cause a compilation error (obs.56). On the other hand, it also doesn't seem necessary to include everything via includes (obs.58) and appears to be includes especially for structuring the Conceptual analysis (obs.57). Keep the includes small (obs.59).

## Common objects (Ampersand as tool)

**Used observations**(see appendix D and B for full text)

- obs.92: rq1-49:30-10: Isolating a pattern or subsystem for testing does not work. This has to do with setting up Docker and possible ignorance on my part
- obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses
- obs.105: rq1-85:30-11: A new structure where the registers can operate independently of each other, with only the generic elements as common items
- obs.30: rq1-89:7-12: Items named as common concepts
- obs.106: rq1-92:14-12: Basically trying to create its own container per register. Multi-context problem. This makes it impossible to isolate these containers

Ampersand is flexible and easily expandable (obs.55), but it is not possible to split a model (obs.92). In the situation where there is a common model and several individual models, it is not possible to separate these individual models from each other by the common part. The case is that we wanted to realize an implementation for each registry. The common part is the recording of the person and the registration (obs.105, obs.30). This component is used for all registries, but there may be differences in the requirements for each registry. The moment a second registry is set up, the common part is also overwritten and all data, including database structure, is gone from the first defined registry (obs.106).



## Crud (Ampersand as tool)

**Used observations**(see appendix D and B for full text)

- obs.60: rq1-12: At the start it is not clear when a capital letter or small letter should be used with the crud in the interface
- obs.113: rq1-50:30-10: The represent statement makes the interface react differently. When using the represent statement, the append option ("+") disappears
- obs.72: rq4-4 The interface produces many messages and these remain
- obs.61: rq1-40:10-10: The concepts used in the interface must be of type "object" (represent). The concept may therefore not be alpha or integer
- obs.62: rq1-53:2-11: The crud (Create, Read, Use, Delete) and CRUD in the interface don't always work as it should be. There is no full validation on usage. So an on/off does not make sense everywhere. rq1-37:3-10: CRUD/crud options also need some study before they can be applied properly
- obs.65: rq1-83:27-11: Experiment with HTML view within the interface fails. Documentation of this is not conclusive. The examples are not enough

An important part of the prototype is the use of the Interface <sup>12</sup>. The **CRUD** interface defines whether **CRUD** items are active at the place in the interface (obs.60). Uppercase is active and lowercase is not active (obs.60). In not all cases the **CRUD** works correctly. Validation on used is missing in some cases (obs.62). After performing the validations on the data and in case of input errors, the interface delivers many messages and these remain active across the screens (obs.72). The Respresent <sup>13</sup> definition (obs.113) of a Concept also has an effect on the Interface (obs.61). Experiment with HTML in the interface was not successful, due to the lack of examples (obs.65).

## PhP (Ampersand as tool)

**Used observations**(see appendix D and B for full text)

- obs.98: rq1-9 Adding pieces of php code in the script is possible, but it is not clear how
- obs.26: rq1-48:27-10: The concept current date is solved very complicated. But eventually it works. Current time does not seem to have developed yet. Although the example scripts seem to say something different
- obs.101: rq2-16:19-10/11-11: Ampersand has a hard time determining a period. Ampersand cannot calculate out of the box. This requires the php functions, which are also not easy to allocate

Ampersand supports creating new functions via php. The documentation doesn't tell you exactly how to do this (obs.98). But in the examples and during the research it appears that it is possible (obs.26, obs.101). This is done in PHP and can then be called within a line using "ExecEngine" (see 8).

<sup>12</sup><https://ampersandtarski.gitbook.io/documentation/the-language-ampersand/services/layout-of-user-interfaces>

<sup>13</sup><https://ampersandtarski.gitbook.io/documentation/the-language-ampersand/atoms>

## Model maintenance (Ampersand as tool)

**Used observations**(see appendix D and B for full text)

- int.23: I-1.5, To be able to use Ampersand it would be useful to avoid having to write code in C#
- int.25: I-1.7, Ampersand does not support a maintenance cycle. There must be a solution for this
- int.11: I-2.4, Does Ampersand support databases other than just MariaDB? Not at the moment, but it is to be expected that this will be possible
- int.12: I-2.5, When maintenance takes place on the model, how do we get from one model to another. So how does the IST go to SOLL situation. Ampersand is always creating a new model. So when the law is changed and a new model is needed as a result, Ampersand will produce a completely new model. As a result, no technical debt will remain in the model. It is always a new model. However, the challenge will be in the data migration from the old to the new model
- obs.10: rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)

Ampersand makes it possible to set up an information system without using a programming language. In case of CIBG it is therefore not necessary to use C#, the language used there (int.23). When performing the analysis, we always create a new model (int.25). There is no question of maintaining a model, because a new model is always being developed (int.12). When we go from model version one to model version two, we now have no tools available to facilitate the data conversion (obs.10). The data from Ampersand prototype is stored in a MariaDB database (int.11).

## Excluded (Ampersand as tool)

**Used observations**(see appendix D and B for full text)

- int.34: I-4.4, A use case can also be devised for the use of rebuilding existing systems. Through the analysis with the help of Ampersand, a system can be rebuilt in which the waste has been cut away. The question is how much this waste would be. Worth a try
- obs.68: rq1-10 The function HTML href with target blank does not work within the interface rq1-77:20-11: In HTML mode the <a href="x" target=\_blank> is not supported. The target is removed in the compilation
- obs.114: rq1-65:10-11: DATETIME (represent) field could not be converted to Excel. The compilation process hangs on this
- obs.86: rq1-66:10-11: XLSX files format is created partly on the basis of multiplicity. one on n relation produces its own tab
- obs.31: rq1-91:14-12: A concept and a relation can be defined several times within your own patterns. So that the patterns can stand on their own
- obs.66: rq1-98:30-12: When using linkto in the interface as last element in the interface and the signature occurs more often than a dropdown to all subinterfaces (of the same signature) appears
- obs.100: rq2-9:7-10: Subscription time is added automatically. This is done by means of a rule

The following observations have not been explicitly included.

## 5.4. Design

The requirements for the design are limited beforehand. The basis of the information was the law and it must fit within the architecture.

Table 6: Category **Design**

Category	CAT1-3
Category Title	<b>Design</b>
Definition	<b>An activity or process that identifies the requirements and then comes up with a solution capable of meeting the requirements.</b>
Anchor examples	<ul style="list-style-type: none"><li>• <b>rq1-36:3-10: What about prototype test scenarios</b></li><li>• <b>rq2-18:16-11: Good to realize that the meaning you write down also ends up in the Conceptual analysis. So looking at the way of writing it down can form a story in the analysis</b></li></ul>
Coding rules	requirements, solution, interface, documentation

## Architectural fit (Design)

**Used observations**(see appendix D and B for full text)

- **int.30:** I-1.12, The terms case, submission and application are strongly represented in the handling of the registers. These terms do not appear in the Ampersand analysis. The term “case” is not mentioned at all in the law-big. Because the process part is missing, this is considered a weakness of Ampersand. It is clear that Ampersand state is oriented and reactive and not process oriented
- **int.21:** I-1.3, The Registerkern, an architecture model of CIBG, uses shared concepts. With this it has similarities with Ampersand. There is also an overlap of Ampersand with Registerkern. Within Ampersand are concepts that are also in Registerkern. Registerkern is a defining part of the architecture. Other parts will have to conform to this architecture
- **int.8:** I-2.1, CIBG’s architecture for new registers consists largely of Registerkern. This was introduced not so long ago and is still being expanded
- **int.13:** I-2.6, Registerkern its terminology includes things and products. Every service, read implementation of a law, we call a product. There are standard parts that always appear in every register. These are pre-modeled within Registerkern. This includes a base for each registry and can be expanded according to the needs of the registry. The basis is the minimum common denominator of the registers. Extendable to specific elements arising from the law. There is certainly overlap in the data obtained from the analysis of the big law and the Registerkern. About 80% of the Registerkern is generic and the other 20% is customised. So all new registers have the same basic principles and for the most part run on the same software
- **int.14:** I-2.7, Another aspect of the terminology is that items with the same definition are named differently within the law and within the Registerkern. In Registerkern we are talking about business and products, while the law is big about registrations, applications and professional registers. A mapping of the terms used will have to take place
- **int.15:** I-2.8, Due to the overlap between Registerkern and the Conceptual analysis of Big, it is difficult to find the demarcation line between the two systems. Ampersand is state oriented and the Registerkern is process oriented. The link and cooperation must be sought
- **obs.9:** rq4-1 Ampersand cannot calculate. But since Ampersand is static, process data can be monitored in other ways
- **obs.20:** rq4-3 Embedding in architecture, the core of the law with shared concepts and processes. The core of law is specific law. Shared concepts are also part of the law but also occur elsewhere. This is part of embedding in architecture
- **obs.55:** rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses
- **obs.30:** rq1-89:7-12: Items named as common concepts

Ampersand’s design must fit into the architecture of the organization where it is used (obs.55). We see that the design using Ampersand overlaps with the existing **Registerkern** (int.21). Mapping is necessary to get Ampersand aligned with **Registerkern** (int.14, obs.30, int.30, int.13). The angles that **Registerkern** and Ampersand use are different. Ampersand is reactive and **Registerkern** is strongly workflow oriented (int.15, obs.20, obs.9). In particular, the generic part of the design shows coherence with **Registerkern** (int.8). Ampersand cannot calculate out of the box.

## Conceptual analysis (Design)

**Used observations**(see appendix D and B for full text)

- obs.38: rq1-1 Formatting in Ampersand (patterns) has consequences for the Conceptual analysis
- obs.53: rq1-75:20-11: Some more experimentation with the documentation in the prototype. When describing the purpose of the context, it takes a while to figure out how this text can be properly conveyed. An <h1> results in an extra chapter in H4 and H4 then becomes H5 and H5 has then become a meaningless piece. With an <h2> and <h3> it works well
- obs.49: rq1-78:20-11: The documentation generated in HTML loaded in firefox and no PNG's are visible. Chrome is doing well
- obs.42: rq1-99:6-1: when generating a Conceptual analysis the doc gets the name of the first concept
- obs.46: rq2-18:16-11: Good to realize that the meaning you write down also ends up in the Conceptual analysis. So looking at the way of writing it down can form a story in the analysis

The design is mainly expressed in the **Conceptual analysis** and the prototype. While building the script, there are options to influence the layout and readability of the analysis (obs.46). Adding HTML headers can disrupt the layout of the analysis considerably (obs.53, obs.49, obs.42). Using includes and patterns structures the **Conceptual analysis** (obs.38).

## Lifecycle law (Design)

**Used observations**(see appendix D and B for full text)

- obs.120: rq1-39:3-10: Do not forget to create delete rules in addition to append and edit rules in the rules in the context of the Lifecycle approach
- obs.34: rq1-95:29-12: The format of a concept big number is not included in the law

During the analysis of the law, it appears that the law focuses on the origin of the registration (obs.34). The law does not clearly consider the life cycle of and registers. You will not find the steps to dismantle a register in the description of the law (obs.120).

## Register unbundling (Design)

**Used observations**(see appendix D and B for full text)

- obs.106: rq1-92:14-12: Basically trying to create its own container per register. Multi-context problem. This makes it impossible to isolate these containers
- obs.107: rq1-93:19-12: Implementation choice for separate registers has an impact on the whole. How to deal with shared modules. How to deal with shared data (such as person). Should the choice be made to only share the concepts and relationships and not implementation
- obs.109: rq3-8:19-9: The law states that there are multiple registers. There is a register per profession. The scripts may also need to be formatted that way

The law speaks very clearly about multiple registers (obs.106, obs.109). The current implementation of **BIG-registration System** is set up as one register with differentiation per profession. When the law is explicitly followed, this has consequences for the design (obs.107). This will then be set up with multiple registers.

## Test scenario (Design)

**Used observations**(see appendix D and B for full text)

- int.32: I-4.2, The prototype shown is not easy for the user to understand. The user not only looks at the functionality, but also at the design. The current design does not comply with the national government web guidelines. The question is whether the user will be able to see through this. It was not part of the research, but it was stated that adjusting the CSS could bring closer to the web guidelines
- obs.104: rq1-36:3-10: What about prototype test scenarios
- obs.103: rq1-69:14-11: Postman application installed and works with the prototype

The use of Ampersand for the design means that there is the possibility to develop test scenarios early in the process (obs.103). Doing analysis and making the prototype gives the opportunity to create test scenarios (obs.104, int.32).

## Brp (Design)

**Used observations**(see appendix D and B for full text)

- int.37: I-4.7, In the current trend, validations are usually located in the business layer. Is that also the case with Ampersand? The validations are spread over the database and surrounding code
- obs.35: rq2-17:10-11: A dutch person has an concept address that must conform to the BRP format (should be a standard building block for it!). A foreign address is unclear what to do with this
- obs.83: rq3-10:12-10: Formatting of the name is not stated literally in the law, but must conform to BRP standards
- obs.84: rq3-11:12-10: Matters such as authorization decisions that allow an information system to retrieve BRP data are not found in the law

In the context of the design you want to have things made explicit (int.37, obs.35). Matters such as address management, country management and address formatting are not included in the law (obs.83). But these items are part of the design. The manner of management must be sought elsewhere than in the law. For example, address formatting is something that the BRP prescribes. This is not an Ampersand item, but one that is encountered when analyzing legislation and regulations (obs.84).

## Total design (Design)

**Used observations**(see appendix D and B for full text)

- obs.64: rq1-71:14-11: The interface also belongs to the design and not just to the prototype. Changing the Create, Read, Use, Delete changes the behavior of the API

There is a lot to be said about the use of Ampersand in the design in many detail points. But bottom-line, the design of the system grows by performing the analysis. The analysis is performed using the scripts (obs.64) (see appendix F). A consequence of this analysis is the design, the **conceptual analysis** (see appendix I and the prototype (see appendix G).

## User experience (Design)

**Used observations**(see appendix D and B for full text)

- int.28: I-1.10, In addition, the practitioner's usual working method is that he works from overviews and lists. Ampersand will have to be designed for this with user requirements, because these things are not mentioned in the law. The law does not support a method and approach. This will have to be a so-called co-creation between IT and business
- int.4: I-3.4, The aim should not be to record everything that is stated in the law in an ICT system. That makes it very rigid. Make sure that 80% of the situations are supported and leave the rest to the employees. Ampersand is very suitable for this, precisely because it has a reactive approach and therefore does not prescribe how the practitioners should act
- int.5: I-3.5, The aim of an ICT system should be to do as little manual work as possible. And when a new law is being developed, an ICT representative should be present
- int.38: I-4.8, How is the maintenance of the system? A new model is always made with the help of Ampersand. The data will have to be migrated itself. Ampersand does not support that. Usually the data structure is taken into account in advance so that as little conversion as possible has to take place. This means that a system is getting bigger and less manageable. So the strength of Ampersand is that this is prevented because a new core system is always being built and the effort is in the data conversion and the connection of adjacent systems

The user usually works from lists and overviews. Ampersand also provides lists and overviews via the interface (int.28). The purpose of an application is to unburden users and to support these users in processing the registry data. Ampersand is well suited for this because the conceptual analysis, from which the implementation is derived, specifies exactly what is needed for the registry (int.4, int.5). Ampersand's design is such that a new design is always delivered (int.38).

## Excluded (Design)

**Used observations**(see appendix D and B for full text)

- int.34: I-4.4, A use case can also be devised for the use of rebuilding existing systems. Through the analysis with the help of Ampersand, a system can be rebuilt in which the waste has been cut away. The question is how much this waste would be. Worth a try
- obs.17: rq4-2 The api link works fine, but entire messages return. These should actually get codes
- obs.72: rq4-4 The interface produces many messages and these remain
- obs.10: rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)
- obs.123: rq2-14:19-10: The role gives control to the user. A user is authorized for use. It indicates which user is allowed to use the function
- obs.71: rq2-15:19-10: In the interface a FOR can also be used. This populates user roles
- obs.100: rq2-9:7-10: Subscription time is added automatically. This is done by means of a rule

The following observations have not been explicitly included.

## 5.5. Registry systems

There are few observations about registration systems. In the interviews, some comments were made about information systems, but that is often not very specific registration systems.

Table 7: Category **Registry systems**

Category	CAT1-4
Category Title	<b>Registry systems</b>
Definition	<b>Using scripting and target generation.</b>
Anchor examples	<ul style="list-style-type: none"> <li>• <b>rq3-9:19-9: The structure of the register's is the same, registers are also called registrations</b></li> </ul>
Coding rules	All about use of information systems

### Registerkern (Registry systems)

<p><b>Used observations</b>(see appendix D and B for full text)</p> <ul style="list-style-type: none"> <li>• <b>int.17: I-2.10, Ampersand's approach is in line with the Registerkern, but not at the implementation level. It doesn't seem possible to implement Ampersand directly, but the analysis seems quite useful for extending Registerkern. Where generality is discovered and for the specific parts of the law. Then we are talking about a conceptual link and not a technical one</b></li> <li>• <b>int.13: I-2.6, Registerkern its terminology includes things and products. Every service, read implementation of a law, we call a product. There are standard parts that always appear in every register. These are pre-modeled within Registerkern. This includes a base for each registry and can be expanded according to the needs of the registry. The basis is the minimum common denominator of the registers. Extendable to specific elements arising from the law. There is certainly overlap in the data obtained from the analysis of the big law and the Registerkern. About 80% of the Registerkern is generic and the other 20% is customised. So all new registers have the same basic principles and for the most part run on the same software</b></li> <li>• <b>int.14: I-2.7, Another aspect of the terminology is that items with the same definition are named differently within the law and within the Registerkern. In Registerkern we are talking about business and products, while the law is big about registrations, applications and professional registers. A mapping of the terms used will have to take place</b></li> <li>• <b>int.16: I-2.9, The usual procedure within a register is the application process for a registration. The Registerkern has a wizard for this, which includes a diploma check, for example. This diploma check is also part of the current implementation of the Wet-BIG</b></li> </ul>
--

Registration systems are aligned with **registerkern** at the CIBG. The approach advocated by Ampersand is conceptually in line with **registerkern** (int.17). The terminology used by **registerkern** differs in many areas from what the **Wet-BIG** prescribes. This is not an Ampersand issue, but means that a mapping of terms must take place after the analysis (int.13, int.14). A number of functions that we model in Ampersand also appear as generic functions in **registerkern** (int.16). Since



Ampersand is a reactive system, process support is controlled from **registerkern** (int.16).

### Demarcation (Registry systems)

<p><b>Used observations</b>(see appendix D and B for full text)</p> <ul style="list-style-type: none"> <li>• obs.105: rq1-85:30-11: A new structure where the registers can operate independently of each other, with only the generic elements as common items</li> <li>• obs.106: rq1-92:14-12: Basically trying to create its own container per register. Multicontext problem. This makes it impossible to isolate these containers</li> <li>• obs.109: rq3-8:19-9: The law states that there are multiple registers. There is a register per profession. The scripts may also need to be formatted that way</li> <li>• obs.108: rq3-9:19-9: The structure of the register's is the same, registers are also called registrations</li> </ul>
--

Across registers, independent of **registerkern**, there are common values between the laws. Here also the generic elements must be marked, so that reuse can take place (obs.105). When reusing and separating elements, the multicontext problem is encountered (obs.106). The format of a register is subjective (obs.109, obs.108).

## 5.6. Analysing law

Reading and understanding the legal texts requires special skills.

Table 8: Category **Analysing law**

Category	CAT1-5
Category Title	<b>Analysing law</b>
Definition	<b>Systems with which data is recorded, whereby the definitions of subjects to be registered are used by all supplying persons and bodies.</b>
Anchor examples	<ul style="list-style-type: none"> <li>• rq1-23:24-10: law Reading is a skill</li> <li>• rq1-26:12-9: Also the laws and the regulations can still have references to other laws and regulations. Because they can be based on these laws or extend it</li> </ul>
Coding rules	data (law) oriented, recording data, definitions of subjects, usage of data

## Environment (Analysing law)

**Used observations**(see appendix D and B for full text)

- obs.75: rq3-4:12-9 There are more laws involved than just the Wet-BIG. rq3-6 12-9 In addition to the law, decisions are also important
- obs.78: rq1-26:12-9: Also the laws and the regulations can still have references to other laws and regulations. Because they can be based on these laws or extend it
- obs.79: rq1-27:12-9: There are also laws and regulations that are not included in this particular law, but are valid from a higher law (implicit references). In case of Wet-BIG this could be eg the Archives Act or the Time Limits Act and Criminal Law
- obs.77: rq1-29:12-9: Not all law- and regulations using Wet-BIG can be found under the search term "big"
- obs.82: rq3-15:24-10: In the law the nationality is mentioned, it also refers to the EU and non-eu residents. It is not recognized that the nationality definition is defined per country

When analyzing the law, other laws must be taken into account. The **Wet-BIG** contains references to other laws and regulations (obs.75). Such as article 14 <sup>14</sup> paragraph 11 refers to the kaderwet <sup>15</sup>. It must always be examined whether the law is in the scope for the analysis (obs.79, obs.77). In other situations no reference is given and knowledge must be available that general laws, such as the "wet algemeen bestuursrecht <sup>16</sup>" must be included (obs.78, obs.82).

## law (Analysing law)

**Used observations**(see appendix D and B for full text)

- obs.80: rq1-23:24-10: law Reading is a skill
- obs.43: rq1-35:14-9: The law has been drawn up in Dutch, which means that the Conceptual analysis can also be done in Dutch
- obs.50: rq1-41:19-10: The "wettenbank" website contains a persistent hyperlink, which can be used in the documentation as reference
- obs.76: rq1-42:21-10: It is easy to deviate from the legal texts. Because they are so hard to read. Some knowledge of the law or the process means that your own interpretation is quickly made. Action research also means that you quickly fall into this trap

A risk that is run by the difficult texts (obs.80) is that people do not read carefully enough and that their own interpretation takes place (obs.76). That risk increases the more familiar the domain is to the researcher. The text has been analyzed in Dutch because it is a Dutch law (obs.43). Persistent hyperlinks are also included in the law, these could also have been included in the meanings and purpose (obs.50).

<sup>14</sup><https://wetten.overheid.nl/jci1.3:c:BWBR0006251&hoofdstuk=II&paragraaf=2&artikel=14&z=2022-04-01&g=2022-04-01>

<sup>15</sup><https://wetten.overheid.nl/jci1.3:c:BWBR0020495&g=2022-04-04&z=2022-04-04>

<sup>16</sup><https://wetten.overheid.nl/BWBR0005537/2022-03-02>

## Parts (Analysing law)

**Used observations**(see appendix D and B for full text)

- obs.37: rq2-1 Substantively includes Wet-BIG also includes disciplinary law (tuchtrecht), which is another branch of sport
- obs.33: rq3-2 By reading the law, a structure becomes clear. The concept Person, Registration and Registration with management and Discipline(Discipline) with measures

In the **Wet-BIG** (obs.33) two large parts can be distinguished. On the one hand, a description of professional protection and on the other, disciplinary law. Both look at the registers from a different side. Professional protection then concerns the exercise of the profession. Disciplinary law usually concerns the wrong actions or treatments that have been performed and the possible measures to be taken. The current implementation at **CIBG** also reflects this, because there is one department that deals with the surveillance of the professions and another department that focuses on the disciplinary part. On the advice of the lawyer, we have not analyzed the disciplinary part (obs.37). The disciplinary section usually contains guidelines for the disciplinary committee.

## Tools (Analysing law)

**Used observations**(see appendix D and B for full text)

- obs.96: rq1-24:12-9 XML download from wetBig seems like a logical step for the analysis and processing, but it is too complex. This also applies to the JSON structure. Both structures are not pleasant to read. The thought that comes to mind here is why SDU doesn't directly annotate the concepts and relationships
- obs.85: rq1-25:12-9: First make overview of all laws and regulations
- obs.97: rq1-31:14-9: Besides XML and JSON, RTF and PDF are also an option. In rtf (doc) you can add items in the margins via "comments". With a PDF, annotations and color highlighting can be given this feature

The overall approach to the analysis of the law is to first get an overview of the law (obs.85). Going through the law and clarifying the highlights of the articles. Keeping an overview via XML, PDF, RTF or JSON is very laborious and also too complex to execute (obs.96, obs.97).

## Suitability of the law (Analysing law)

**Used observations**(see appendix D and B for full text)

- **int.29:** I-1.11, The question is whether the wet-big is very suitable for this approach. The original law dates from 1993 and it is based on the legislation of 1865
- **int.26:** I-1.8, Ampersand's possible positioning is to use it as an interpreter of legislation and regulations. Then maintain the current analysis and development process and use the prototype to validate the analysis. The question is whether this approach will not result in additional work compared to the current working method. There is a certain skepticism towards Ampersand
- **int.3:** I-3.3, The Wet-BIG offers a lot of room for interpretation. This interpretation possibility means that the law may lend itself less to an Ampersand translation than a recent law would. The new laws have therefore been drafted more carefully. The law provides a framework and the question is how far one should go with recording. This law gives the freedom to fill in matters yourself
- **int.6:** I-3.6, Because the law was drafted some time ago, the definitions are not always unambiguous. And because of the aforementioned interpretation possibility, the legislator can interpret the law slightly differently through jurisprudence
- **int.31:** I-4.1, The Wet-BIG is big and also old. Ampersand could help detect inconsistencies in the law

From various interviews the statement was made whether **Wet-BIG** is the most suitable law to analyze it with Ampersand (int.26). The reason is that the law of origin is very old (int.29) see section 3.3 and it is quite comprehensive. The law has been updated several times, but the structure is not easy to convert to an ICT system (int.31). In addition, the law contains many implicit and explicit references to other laws and regulations. And the law itself is not explicit enough (int.6). There are quite a lot of interpretation possibilities (int.3).

## Excluded (Analysing law)

**Used observations**(see appendix D and B for full text)

- **obs.107:** rq1-93:19-12: Implementation choice for separate registers has an impact on the whole. How to deal with shared modules. How to deal with shared data (such as person). Should the choice be made to only share the concepts and relationships and not implementation
- **obs.34:** rq1-95:29-12: The format of a concept big number is not included in the law
- **obs.125:** rq3-13:17-10: There is no list of specialties in Wet-BIG, where is it?

The following observations have not been explicitly included.

## 6. Discussion

The most common keywords within the observations in appendix A are with the terms "Ampersand", "Api ", "Concept", "Conceptual analysis", "Interface", "Law", "Multiplicity", "Relation" and "Rule". These are terms that in most cases are part of the Ampersand method. Only the term "law" refers to the case itself.

This chapter focuses on the relationship between the results (see section 5 and the sub-questions. A separate section has been included for each sub-question in which the sub-question is answered on the basis of the included results. Each paragraph contains a reference to the respective results to which it relates, in whole or in part.

Many observations were made during the research. The analysis showed that not all observations are relevant. These were therefore not included in the further analysis. Even observations that were initially assigned to a category turn out to be irrelevant on closer inspection and have been included as such.

### 6.1. Ampersand knowledge

When working with Ampersand, a development environment must first be set up. Ampersand's documentation assumes that XAMPP <sup>17</sup> can be configured for this become. The preferred configuration is done with the help of Docker. If there is little or no knowledge of Docker, you can choose to set up XAMPP. However, it was not possible to get the local installation working with the help of the documentation. However, with help we managed to get it working in the Docker environment. So starting with Ampersand, it's not just Ampersand that needs to be studied, it's also the environment in which it operates that needs to be studied. This part always works. Setting up the environment and Ampersand is also fine from the website and from Github. There is also some information on Stackoverflow. But beyond that there is nothing to be found outside of a number of scientific articles. Fortunately, there are articles like Michels et al. [2011] that explain Ampersand.

The sub-question "What knowledge, in the role of software engineer, is needed to use Ampersand" examines the knowledge of the software engineer when using Ampersand. To answer this question we can use the next results: 5.1.1 Setup (Useful), 5.1.2 Script creation (Useful), 5.1.3 Source handling (Useful), 5.1.4 Script overview (Useful), 5.2.1 Naming (Ampersand as method), 5.2.2 Multiplicity (Ampersand as method), 5.2.3 Rules (Ampersand as method), 5.2.4 Concept reuse (Ampersand as method), 5.4.2 Conceptual analysis (Design), 5.3.1 Includes (Ampersand as tool), 5.3.2 Common objects (Ampersand as tool), 5.3.3 Crud (Ampersand as tool) and 5.3.4 PhP (Ampersand as tool)

In order to be able to use Ampersand itself, in addition to knowledge about the structure of the environment, knowledge of Ampersand itself is required. There is not a lot of information about Ampersand on the internet and there are examples, but they do not cover the whole load. As with any tool and method, knowledge

---

<sup>17</sup><https://github.com/AmpersandTarski/documentation/blob/master/installing-ampersand/installing-the-tools-manually.md>

will have to be kept up to date to keep Ampersand usable. This is not specific to Ampersand, but of course also applies to Ampersand. (Ref. to [5.1.1 Setup \(Useful\)](#))

Proper use of Ampersand requires proper documentation setup. This setup consists of knowledge of the way Ampersand handles the information in the scripts. The positioning of the meaning of the Terms and the Relationships and the purpose of the Rule and the use of includes. One should be aware that the meaning, purpose and definitions appear directly in the documentation and the inclusions help determine the order of the story and that this should be a unifying story. Agreement must be reached in advance about the structure of the spelling, the reference to legislation and regulations. The notation method, the naming convention of Concepts and Relations must also be unambiguous in order to have a consistent and professional appearance. As one of the interviewees pointed out, when analysis and documentation needs to be done, why not through Ampersand. Unfortunately, the researcher started to standardize somewhat later, so that this was not implemented everywhere. (Ref. to [5.1.1 Setup \(Useful\)](#), [5.1.2 Script creation \(Useful\)](#))

When working with Ampersand and going through the text, it seems logical to go through the text chronologically. However, this will not always work and a method must be found to maintain the overview. Maintaining overview is difficult when using Ampersand because the source can be huge. This aspect has been approached in various ways. The source text in XML has been looked at, with the intention of adding extra XML tags. The intended side effect of this was that we could generate the model from the XML. This didn't work because the original XML is way too complex and would make the XML parsing very complex as well and the work has to be redone with a new version of the law. The RTF format was like a Word document and could be provided with comments. The same was true for the PDF format. *Annotation*<sup>18</sup> is also possible here and one can also underline with colors. In the end, the old-fashioned choice was made for the combination of hard copy and the PDF. Hard copy for streaking and writing and the PDF for searching and copying text. (Ref. to [5.1.3 Source handling \(Useful\)](#))

Maintaining the overview in the created scripts is also a challenge. By using Visual Studio<sup>19</sup> there are no *refactor*<sup>20</sup> options. Visual Studio also seems to lack integration between the scripts. The consequence of this is that it is possible that the same Concepts and also Relationships are defined in several places. By not being aware of the overlap, a different definition can occur for the same Concept. The differences will not be very large, but certainly worded differently. This only came to light when the documentation was generated. The advantage of working with text and generating the model from it appears to be a disadvantage here. (Ref. to [5.2.4 Concept reuse \(Ampersand as method\)](#), [5.3.2 Common objects \(Ampersand as tool\)](#))

---

<sup>18</sup>Appx. E func.req.:An annotation tool is required to maintain an overview of the text to be processed. This prevents things from being processed twice or not.

<sup>19</sup>Visual Studio Code observations are not categorised, no direct involvement with Ampersand

<sup>20</sup>Appx. E func.req.:There is a need to enable refactoring within an IDE. We can then prevent issues when removing, for example, Includes. Changing naming or viewing where a Concept or Relation occurs is highly desirable.

When creating a script where the analyst does not yet have that much experience, it happens that the meaning and purpose are not filled in. This is caused by the analyst being too busy getting Relations and Rules working within the scripts. The consequence of this is that meanings and purposes are not filled in and they therefore become visible in the **conceptual analysis**. It is almost impossible to update it afterwards. This can be prevented by working as a team, where the team members keep each other sharp on these matters and there is experience in the team. (Ref. to **5.1.4 Script overview (Useful)**)

Article 3(1) of the **Wet-BIG** states that there are several registers. In order to implement this, an attempt has been made to install a single register in combination with the shared piece. The *shared components*<sup>21</sup> appear in each register and include the person registration and the registration leading to the actual registration. The implementation of the prototype of a profession with the common part was no problem (9. Until the next profession was established (10).

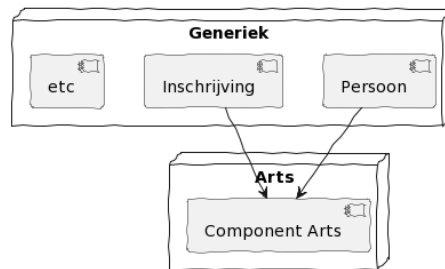


Figure 9: Arts with generiek

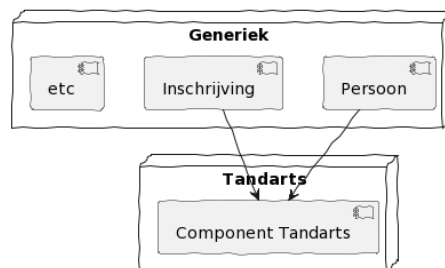


Figure 10: Tandarts with generiek

Then the first professional group was removed from the database and the common data was also removed and the second group was fully initiated. The method of building specific professional registers in this way is not (yet) supported by Ampersand, so it has to be deployed all in once (11).

<sup>21</sup>Appx. E func.req.:Dealing with shared components such as Concepts, Relations or Patterns. This both within a project and across the projects.

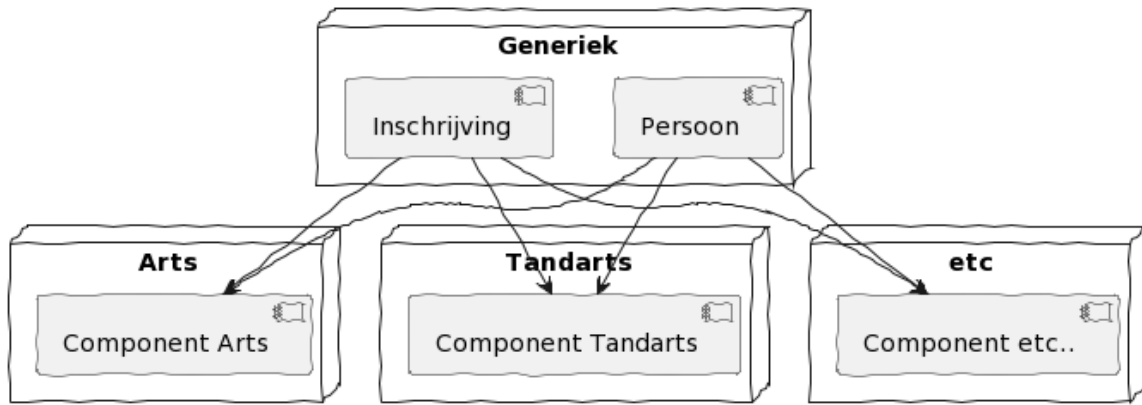


Figure 11: Deployment in once

If this does not work for this law, it will also not work for links with other registers where data and the associated management software must be shared. (Ref. to [5.3.2 Common objects \(Ampersand as tool\)](#))

Every organization has an authorization mechanism for the software. The CIBG uses a JWT mechanism. In the research I encountered an authorization mechanism when creating the Rules and when building the Interface. I have not found out whether it is possible to integrate this with the organization's own authorization mechanism. (Ref. to [5.3.3 Crud \(Ampersand as tool\)](#))

To be able to use Ampersand correctly, knowledge of relation algebra is required. You use this algebra as a Software engineer to build rules and relationships. The knowledge of multiplicity is indispensable when creating rules and relationships. The naming convention is partly present within Ampersand. Some elements, such as pattern, for example, are capitalized. Concepts always start with a capital letter. For example, there are a number of fixed agreements. There are no agreements regarding the use of, for example, relation names. For matters for which Ampersand has no constraint, a set of best practices could be set up. (Ref. to [5.2.1 Naming \(Ampersand as method\)](#), [5.2.2 Multiplicity \(Ampersand as method\)](#), [5.2.3 Rules \(Ampersand as method\)](#))

The Software engineer needs limited knowledge of Latex and HTML to influence the **conceptual analysis**. In many cases this is not necessary because Ampersand handles this excellently. However, there are opportunities to intervene and provide direction here. (Ref. to [5.4.2 Conceptual analysis \(Design\)](#))

The Software engineer must know how to control Ampersand in terms of the use of includes. These includes control the **conceptual analysis**, but are also used when building the application. If the includes are not specified where they are needed, the build will fail and when it is specified where not necessary, the build goes well and the **software engineer** has the option to send the **conceptual analysis** on content. The development of extra functions that have not yet been included within Ampersand is done using PHP. The Software engineer therefore needs knowledge of PHP to develop these functions and of course Ampersand to be able to actually deploy these functions. (Ref. to [5.3.1 Includes \(Ampersand as tool\)](#), [5.3.4 PhP](#))



(Ampersand as tool)

The team that Ampersand maintains and wants to expand is very active. The involvement is also apparent from the rapid resolution of issues that occurred. (Ref. to 5.1.1 Setup (Useful))

## 6.2. Ampersand core in wet BIG

With sub-question "What are the Concepts, Relationships and Rules in the Wet-BIG" we want to know what is in the Conceptual analysis. For this appendix D is included. The Conceptual analysis contains the Concepts, Relations and Rules (see figure 12). The Conceptual analysis is not completed because not all articles and

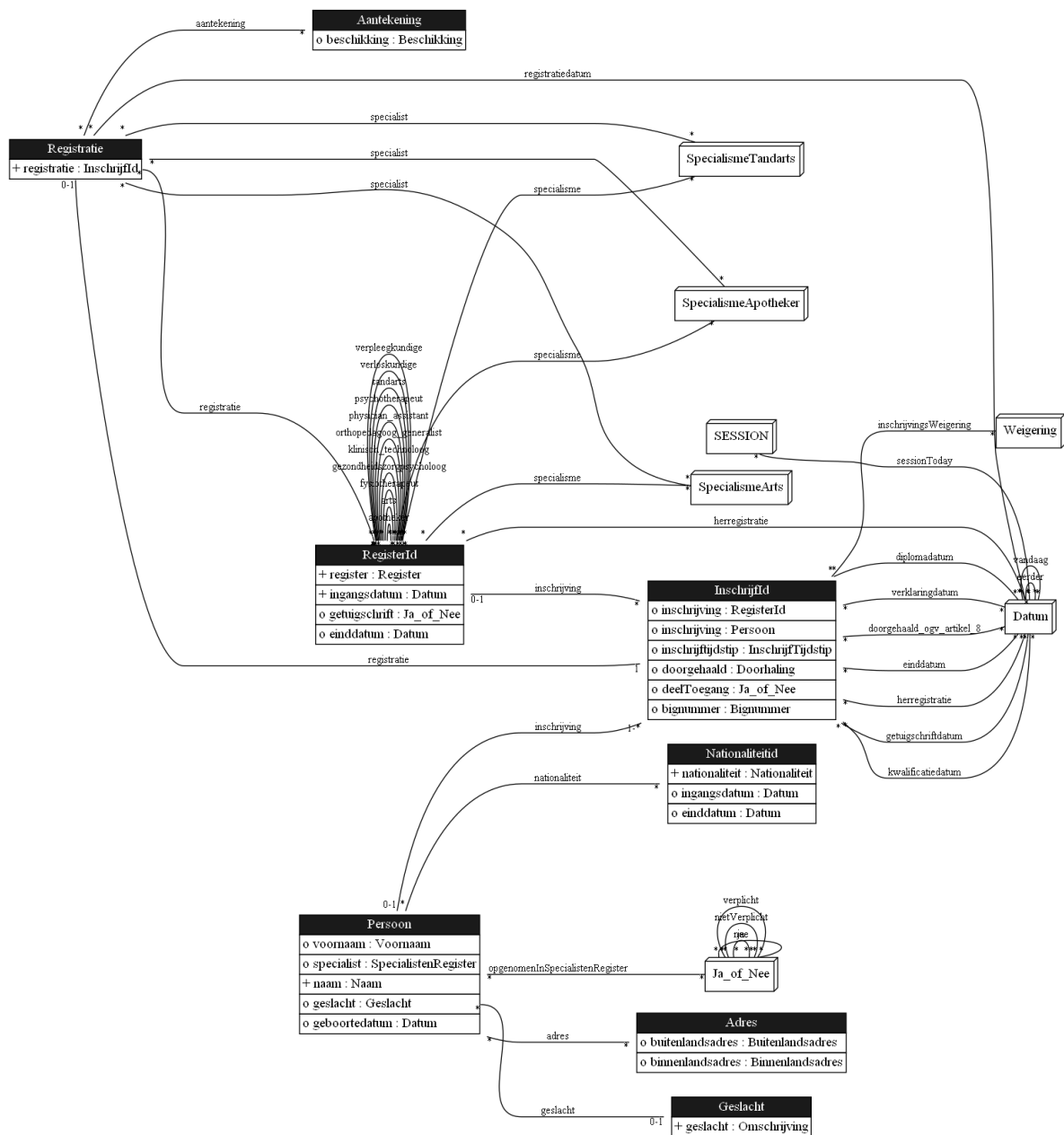


Figure 12: LogicalDataModel from the Conceptual analysis

related regulations has been analyzed.

### 6.3. Setup law for Ampersand

Reading and understanding the legal texts requires special skills. For example, article 13 paragraph 1, here it reads: *"Indien bij besluit van Onze Minister inschrijving in een register is geweigerd, de afgifte van een verklaring van vakbekwaamheid wordt geweigerd of een beroepsbeoefenaar de bevoegdheid zijn beroep uit te oefenen heeft verloren omdat hij de aanvraag tot inschrijving of tot afgifte van een verklaring gebaseerd heeft op valse kwalificaties, kan Onze Minister besluiten, onverminderd de hoofdstuk V van de Algemene verordening gegevensbescherming, de bevoegde autoriteiten van andere staten dan de staten bedoeld in artikel 31a, eerste lid, van de Algemene wet erkenning EU-beroepskwalificaties, daarvan in kennis stellen."* Due to the length of the sentences and the many parentheses, the analysis of a piece of legal text can only be read properly by a person with experience in reading legal documents.

The sub-question **"How are the laws and regulations set up so that they can be used in a useful way for the Ampersand method"** deals with the law, in the case of the **Wet-BIG**, and the way in which it can be analyzed and processed using Ampersand. To answer this question we can use the next results: **5.1.5 Data add (Useful)**, **5.4.3 Lifecycle law (Design)**, **5.4.4 Register unbundling (Design)**, **5.6.1 Environment (Analysing law)**, **5.6.2 law (Analysing law)**, **5.6.3 Parts (Analysing law)**, **5.6.4 Tools (Analysing law)**, **5.6.5 Suitability of the law (Analysing law)**, **5.4.6 Brp (Design)**

Interviews paint a picture of a law that originated in the 19th century. Although this has been adapted to the current times, the structure is not equipped for a one-to-one translation to an information system. It has been indicated that there are new laws that are much better suited for translation, such as, for example, "Regeling bewijsstukken sociale hygiëne Drink- en Horecawet 2015". Since we have not analyzed any other laws, it cannot be determined whether this is the case. But the **Wet-BIG** is a large and complex law, according to a lawyer at the **CIBG**. (Ref. to **5.6.5 Suitability of the law (Analysing law)**)

Analyzing the law requires legal knowledge. Reading the legal texts also requires the necessary experience. Analyzing the law is usually not the domain of a business analyst. At the start of the analysis, a team should be set up that should include at least an analyst and a lawyer. The analyst for building and managing the script and the lawyer for the translation of the law into Concepts and Relationships. This ensures consistency and completeness of the analysis. It has been found that even a lawyer can understand the concepts and the relationships of conceptual analysis. As a result, the cooperation on this point will run smoothly. (Ref. to **5.6.2 law (Analysing law)**)

By starting with the analysis of the law with the help of a lawyer, an overview can be obtained at an early stage of the content and structure of the law. By looking at the structure, the analyst can better understand what the law is about. The structure can also help to determine the structure of the patterns. It's certainly

not the case that every chapter is a separate pattern, but it certainly influences the setup of the **Conceptual analysis** and thereby help to gain an overview of the law and, on the other hand, of the analysis to be performed. (Ref. to **5.6.4 Tools (Analysing law)**, **5.6.3 Parts (Analysing law)**)

In order to extract the correct data and understandable data from the source text, experience is required in reading and interpreting the legal texts. Some laws lend themselves to this better than others. In addition to the understandable law, a law analyst is also needed. (Ref. to **5.1.5 Data add (Useful)**, **5.1.6 Deviation (Useful)**)

The set-up of the **Wet-BIG** is limited in nature. The limitation is that it does not include lifecycle management. The law deals with how a person can register and deregister. The law also specifies the requirements that the person must meet in order to remain registered. This per is partly general and partly per register. The missing lifecycle management relates to the management of the registers themselves. The law states that they are there, in decrees more are added. But nowhere is it written what should be done when cleaning one or more registers. (Ref. to **5.4.3 Lifecycle law (Design)**, **5.4.4 Register unbundling (Design)**)

In the initial analysis of an Ampersand assignment, the scope will be determined. This scope is often more than the law itself. In the case of **Wet-BIG** there is a list of rules and decisions (see **C**). In addition to the immediately findable legislation and regulations, there are also overarching regulations that play a role. In some cases it is legislation and regulations that influence the scope, but it is rules that are determined by another source. Think of the NORA architecture rule, the formatting rules of addresses by BRP. So a set of people are needed to determine the scope. (Ref. to **5.6.1 Environment (Analysing law)**, **5.4.6 Brp (Design)**)

#### **6.4. Ampersand for government organization**

The sub-question "**What are the strengths and weaknesses (SWOT) in using Ampersand for registry systems for a government organization**" focuses on the use of Ampersand within the **CIBG** organization. The information systems that are not based on legislation and regulations and which aim to monitor data quality are often the registers. **CIBG** builds, manages and monitors this data through registration systems.

In conversations with an architect of the **CIBG**, maintenance of the Ampersand model is discussed. When a model is set up, this results in a certain version of the model. The model consists of a database model and the other software and an **conceptual analysis**. This model can be implemented by a development team. Legislation will certainly be amended during the software's life cycle. By including these changes in the model, a new model is created. Ampersand does not provide any resources to guide the conversion from the old model to the new model. The development team will therefore have to make an analysis of the old and the new situation and have to develop conversion software for that. This is a method that is different from usual. Usually when changing the software, the changes in the database are taken into account immediately. The advantage of a new model is that

the software does not have to do with legacy. It is therefore always a state-of-the-art model. The downside is that the conversion is likely to be complex. Data that was previously valid may be invalid in a subsequent model. (Ref. to [5.1.9 Model maintenance \(Useful\)](#), [5.4.8 User experience \(Design\)](#))

Ampersand is declarative and reactive, so the Ampersand implementation always responds to the current situation through validations. The execution of management processes is left to the **registerkern**, which supports the process handling. (Ref. to [5.5.1 Registerkern \(Registry systems\)](#))

Although Ampersand is intended as a design and prototyping tool, it does have APIs at its disposal. This can only be obtained from log lines and apparently not intended as a means of communication from external systems. This is also apparent from the fact that no API description is made in, for example, Swagger<sup>22</sup>. But it can work that way. It is possible to communicate with the Ampersand core from an external source. The return actions from the called APIs are not provided with a code but text. As proof of concept, calls were made from Postman (see figure 13) to Ampersand and that worked as expected, see figure 7. (Ref. to [5.1.8 Api \(Useful\)](#))

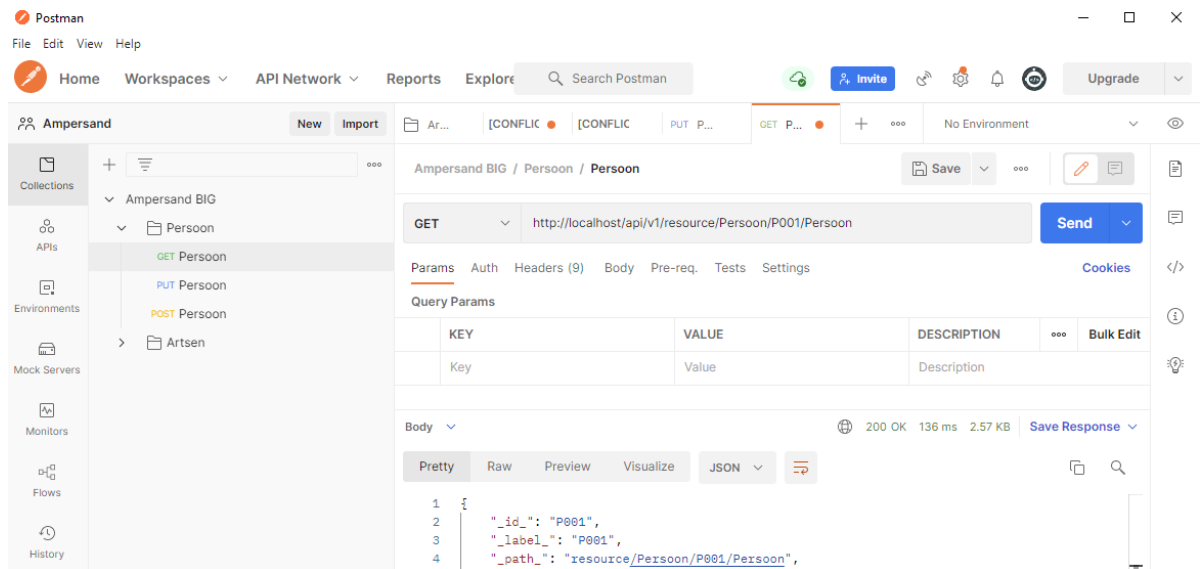


Figure 13: Postman GET Person

```

1 {
2   "_id_": "P001",
3   "_label_": "P001",
4   "_path_": "resource/Persoon/P001/Persoon",
5   "_view_": [],
6   "Persoon": {
7     "_id_": "P001",
8     "_label_": "P001",
9     "_path_": "resource/Persoon/P001/Persoon/Persoon",
10    "_view_": [],
11    "_ifcs_": []
12  },
13  "Naam": "Edelaar",
14  "Voorna_40_a_41_m_40_en_41_": "Gerard",

```

<sup>22</sup><https://swagger.io>

```

15 "Geslacht": {
16     "_id_": "M",
17     "_label_": "M",
18     "_path_": "resource/Persoon/P001/Persoon/Geslacht/M",
19     "_view_": [],
20     "Code": {
21         "_id_": "M",
22         "_label_": "M",
23         "_path_": "resource/Persoon/P001/Persoon/Geslacht/M/Code",
24         "_view_": [],
25         "_ifcs_": []
26     },
27     "Omschrijving": {
28         "_id_": "Man",
29         "_label_": "Man",
30         "_path_": "resource/Persoon/P001/Persoon/Geslacht/M/Omschrijving/Man",
31         "_view_": [],
32         "_ifcs_": []
33     },
34     "_ifcs_": []
35 },
36 "Adres": [
37     {
38         "_id_": "adres1",
39         "_label_": "adres1",
40         "_path_": "resource/Persoon/P001/Persoon/Adres/adres1",
41         "_view_": [],
42         "_ifcs_": [
43             {
44                 "id": "Adres",
45                 "label": "Adres"
46             }
47         ]
48     }
49 ],
50 "Geboortedatum": "2000-01-01",
51 "Nationaliteit": [
52     {
53         "_id_": "0001",
54         "_label_": "Nederlandse",
55         "_path_": "resource/Persoon/P001/Persoon/Nationaliteit/0001",
56         "_view_": {
57             "nationaliteit": "Nederlandse"
58         },
59         "_ifcs_": []
60     }
61 ],
62 "Inschrijving": [
63     {
64         "_id_": "I001",
65         "_label_": "I001",
66         "_path_": "resource/Persoon/P001/Persoon/Inschrijving/I001",
67         "_view_": [],
68         "_EMPTY_": {
69             "_id_": "I001",
70             "_label_": "I001",
71             "_path_": "resource/Persoon/P001/Persoon/Inschrijving/I001/_EMPTY_",
72             "_view_": [],
73             "_ifcs_": [
74                 {
75                     "id": "Inschrijving",
76                     "label": "Inschrijving"
77                 }
78             ]
79         },
80         "_ifcs_": []
81     }

```

```
82 ],
83   "_ifcs_": []
84 }
```

Listing 7: Postman output from GET Person

By using Ampersand as a design tool, a prototype is available at an early stage. This prototype can be converted into a website with the appearance of a **CIBG** website by means of HTML additions and CSS adjustments. Test cases can already be developed at an early stage on the basis of this prototype and the functions of the prototype, by using the APIs, can be used as a stub in the development of the system. (Ref. to [5.2.6 Prototype use \(Ampersand as method\)](#))

An organized ICT organization such as the **CIBG** has an architecture that new software must comply with. One of the developments in the **CIBG** is the set-up of the **registerkern** (see interview developer Appendix B). **Registerkern** its terminology includes "zaken" and "producten". Every service, read implementation of a law, we call a product. There are default items that always appear in every registry. These are pre-modeled in **registerkern**. This includes a foundation for each registry and can be expanded to meet the needs of the registry. The basis is the minimum common denominator of the registers. Extendable to specific elements arising from the law. There is certainly overlap in the data obtained from the analysis of the great law and the **registerkern**. About 80% of the **registerkern** is generic and the other 20% is custom. All new registers therefore have the same basic principles and largely run on the same software. However, a mapping still has to take place from the found Concepts from the law to **registerkern**. The overlap in this is not always immediately visible. Within the **registerkern** the term product is used, within the **Wet-BIG** this product is a register or possibly even all registers. The latter depends on the implementation. This mapping must be made explicit and has not been taken into account in this study. The developer has indicated that full integration of the Ampersand model is not possible, due to the aforementioned overlap of the Concepts. However, the modified part of the **registerkern** can be used for the Ampersand model implementation. Calculations are not standard in Ampersand, but this can be solved by writing external functions or even solving it in **registerkern**. (Ref. to [5.5.2 Demarcation \(Registry systems\)](#), [5.1.7 Architecture and registerkern \(Useful\)](#), [5.4.1 Architectural fit \(Design\)](#))

In RAP there is a tool called Atlas, it shows the context and the patterns. In addition, all Concepts, Rules, Properties and Relations, with hyperlinks to the components. Via the hyperlink details of the item inclusion a relationship diagram is shown. Very nicely executed and very useful when working in RAP. This tool is a viewer on the information and it is not possible to also edit the *information in Atlas*<sup>23</sup>. However, the case study was so large that the RAP environment was not sufficient. RAP does not support Includes and has been used extensively. Unfortunately, *Atlas availability*<sup>24</sup> cannot be implemented outside of RAP. (Ref. to [5.3.5 Model maintenance \(Ampersand as tool\)](#))

<sup>23</sup>Appx. E func.req.:Being able to edit the Atlas information from Atlas

<sup>24</sup>Appx. E func.req.:Make Atlas available outside the RAP environment.

As a novice user of Ampersand it takes a while to master Relation algebra. That is why an excel sheet has been made to make the Relations visible with the associated multiplicity. This is a method that is easy to use initially. The disadvantage of this approach is the consistent transfer of Concepts and Relations. We need to double track this information and redundancy in the field of data will certainly go wrong. When Concepts disappear, they must also be removed from Excel or Relations that do change due to new insights must be adjusted here. In short, this does work for small, well-arranged projects, but for larger ones, gaps will quickly arise and this no longer represents reality. The result was that the excel sheet was used a lot in the beginning and not anymore later on. (Ref. to [5.2.2 Multiplicity \(Ampersand as method\)](#))

To be able to perform the analysis properly, it is not enough to have one person perform it, as in the study. Due to inexperience with the use of Ampersand, the initial appointment set was not created. Ampersand's knowledge is only really gained during the execution. In addition, the amount of legal texts is so large that it cannot be passed through within a reasonable period of time. In addition to IT knowledge, legal knowledge is also required, on the one hand to be able to read the law and on the other hand to find the implicitly related laws and regulations. Depending on the size of the legislation and regulations to be analyzed and the lead time that one wants to use, a team size will be determined. A team consists of at least a lawyer and an (Ampersand) experienced business analyst. A third person to validate the data. (Ref. to [5.2.5 Team \(Ampersand as method\)](#))

Ampersand is a completely new method for the CIBG organization. People have never heard of it and unfortunately there is not much to be found about it. This means that people are not positive about it in advance, calling NIH <sup>25</sup>[[Antons et al., 2017](#)]. It is expected that the Ampersand method will take more time than the current method (see interview [26](#)) and is wary of anything new. It is clear that the advantages of the method are not yet understood. Benefits such as working directly on the source, generating a prototype from there (see appendix [G](#)) with all validations and full conceptual specifications (see appendix [I](#)). Having a prototype makes it possible to build test scenarios at an early stage and with the [Conceptual analysis](#) you can start building immediately. (Ref. to [5.4.5 Test scenario \(Design\)](#), [5.2.6 Prototype use \(Ampersand as method\)](#), [5.1.11 Law effective \(Useful\)](#))

Although Ampersand is new to the CIBG organization, one of the interviewees pointed out (see interview analyst [B](#)) that documentation in the form of a design should be made with each new register. According to the analyst, it should not matter which tool is used for this. The advantage of Ampersand is that it generates a model from the analysis instead of the usual model-to-text approach. A model is made of each pattern. See for example the Pattern for Person (see script [8](#)). This results in the model of figure [14](#). (Ref. to [5.1.10 Ampersand design method \(Useful\)](#), [5.2.7 Organisation Ampersand use \(Ampersand as method\)](#), [5.4.7 Total design \(Design\)](#))

---

<sup>25</sup>not invented here

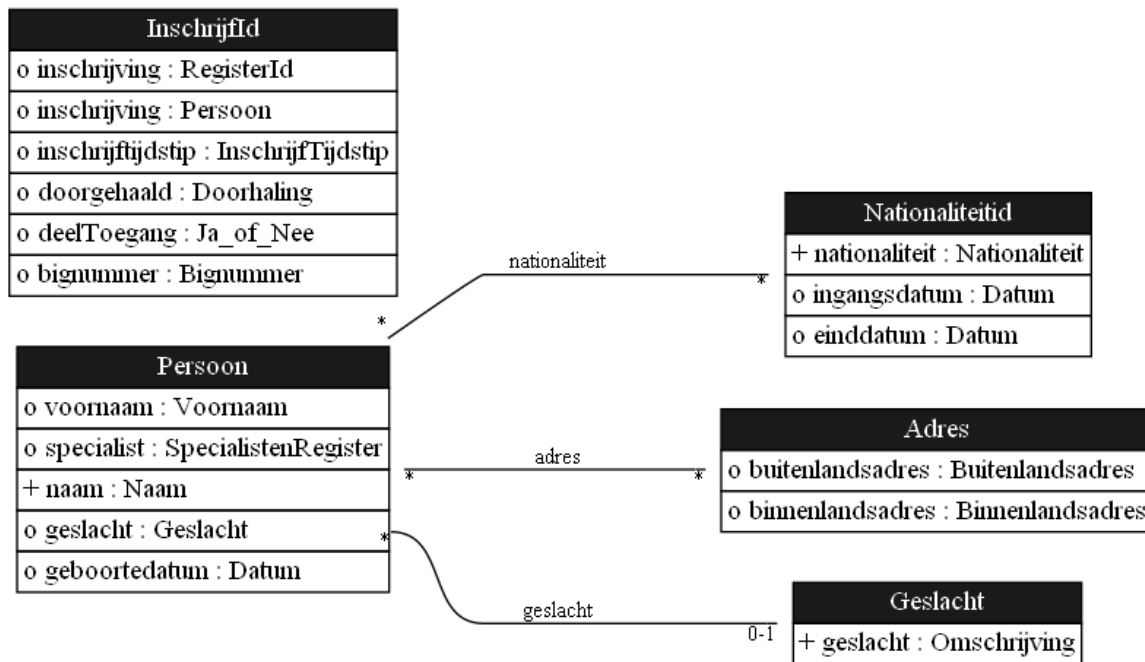


Figure 14: Pattern Persoon

The design of registry systems has no specific points of attention. Part of the research question was about designing for registry systems. Other than the source specific link, the **Wet-BIG**, no specifics were found for registry systems. The translation of this law into an information system results in a register. The requirements of the register are laid down in law. This concerned, among other things, the identifying data of a registration. In article 3, paragraph 1<sup>26</sup> it says it's about registers. This also explains why there are not much observations regarding registry systems. The register system is therefore not an isolated one, but a consequence of the fact that the source is a law.

## 6.5. Limitations

The focus of the research was on executing the process to construct at a **conceptual analysis** and a prototype. The time for this is in principle 2 quarters. We have moved a little over time and it turns out that it is not possible to fully analyze just a law like **Wet-BIG**. The limitation we encountered is that there is a shortage of time. This has to do with an optimistic estimate of what work can be done. The start with Ampersand is more difficult than it seems. The **Wet-BIG** is much larger and more complex than it first appears. Reading the law is also an art. The law has many references to other laws. This resulted in an **conceptual analysis** which is not complete. The process did provide enough material to make several statements (see section 7.1)

<sup>26</sup><https://wetten.overheid.nl/jci1.3:c:BWBR0006251&hoofdstuk=II&paragraaf=1&artikel=3&z=2022-04-01&g=2022-04-01>



## 7. Conclusions and Recommendation

### 7.1. Conclusions

In this research we investigated how useful Ampersand is for designing registry systems by analyzing legislation and regulations. We did this in the form of **action research**. The case used is the **Wet-BIG**.

The Ampersand method was used to analyze part of the law and to process it via scripting (see appx. **F**) into a prototype (see appx. **G**) and **Conceptual analysis** ( see appx. **I**). During the analysis phase, observations (see appx. **A**) were made. These are recorded with date and time stamp. In addition to the analysis, there were interviews (see appx. **B**) with a number of people from the **CIBG** organization. During the interviews the Ampersand approach was discussed and the **Conceptual analysis** and the prototype were discussed. The collected data from observations and interview has been input for the content analysis (see appx. **D**).

In addition to the main question "**How useful is Ampersand for designing registry systems by analysing public health legislation and regulations, in particular the Wet-BIG**" sub-questions have been defined. The sub-questions contribute to answering the main question. The parts of this question are discussed in subsection **6.1**.

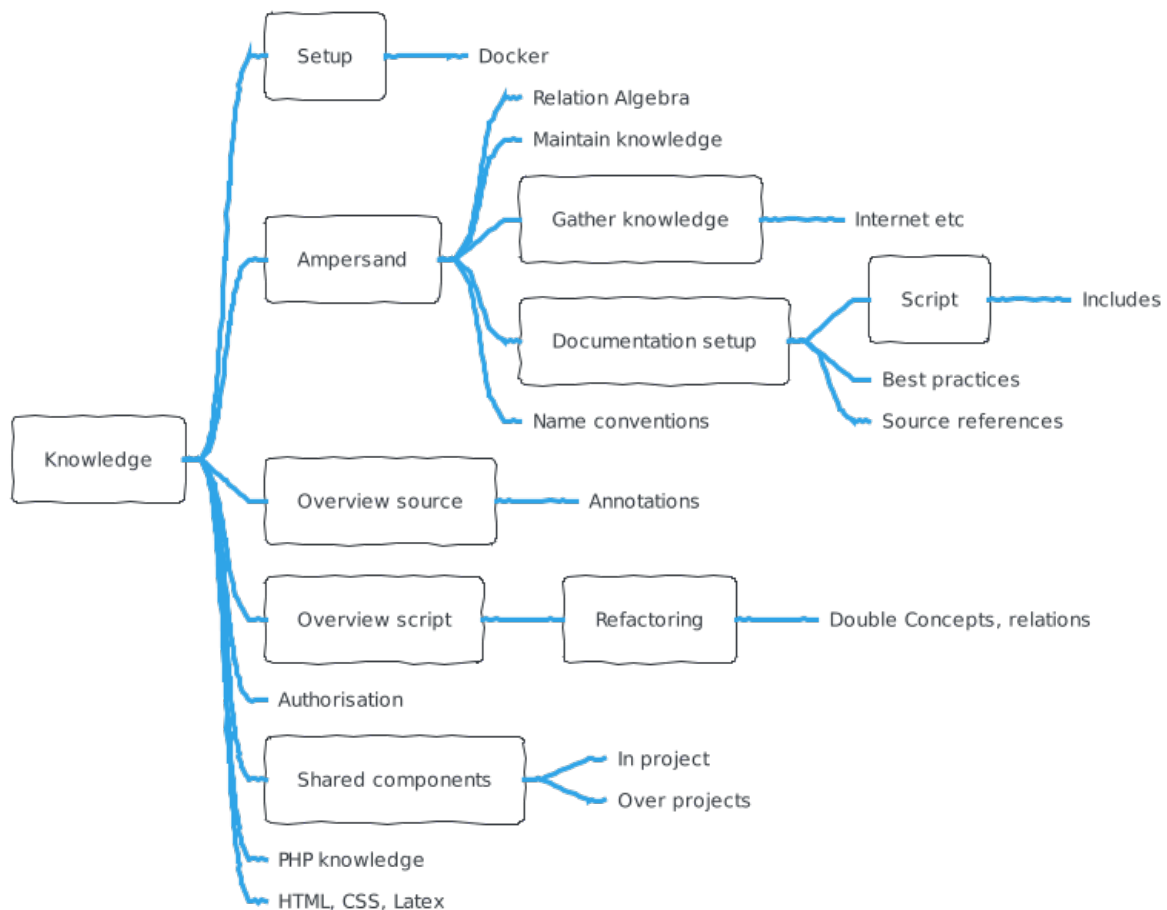


Figure 15: Mindmap knowledge

The knowledge that the software engineer needs to be able to work with Ampersand is not just limited to the knowledge of Ampersand. Based on the section 6 we have compiled a mindmap(see fig 15 in which the matters related to the required knowledge are displayed. The setup of Ampersand was finally set up in Docker, after we experimented with a RAP environment on the server of [Open Universiteit](#) and later a local environment in [XAMPP](#). We need knowledge about the design, use and operation of Docker. Relationship Algebra is used to create relationships between Concepts and to compose the Rules. Knowledge of Relation Algebra in combination with Ampersand can be acquired by following the Rule-Based Design course of the [OU](#), or at least by reading the accompanying book. When Ampersand's knowledge has been acquired, through theory and practice, it is important to keep this knowledge up to date. Ampersand's application knowledge should be available on the Internet today, at sites like <https://stackoverflow.com> and other reliable information sites. Due to a small community and, as noted earlier, low usage, the application knowledge on the internet is very limited. To build a readable [Conceptual analysis](#) we use the scripts of Ampersand. For the [Conceptual analysis](#) we use the Include statements to control build. Best practices should be collected to simplify the start of an Ampersand project. In these best practices things like naming (upper and lower case, CamelCase, etc) are included and proposals regarding the use of source texts. When we are performing the analysis, there is a need for overview. On the one hand, an overview of the treatment of the source document, ie which parts of the legal texts have already been processed and which have not yet been processed. For this you need an annotation tool, which helps you to record the processing and helps you to keep an overview. On the other hand, an overview is also needed during the creation of the script to be able to refactor things and avoid duplication's. There is a tool for this, called Atlas, but it is only available in the RAP environment and not in the local setup. Ampersand has built in a form of authorization that works over Rules and at the Interfaces. With this authorization, a distinction can be made between user roles and the applications, in the prototype, that may be executed. Knowledge is also required about dealing with shared Concepts. Sharing can relate to Concepts within one project, whereby, as in the case, generic patterns with associated Concepts are shared or reused. This form of sharing works when all components are deployed simultaneously, but it is then not possible to run different non-generic components side by side (see figure 9, 10 and 11). The foregoing concerns sharing Concepts within a project. Connection must be made with existing Concepts, which are not always called Concepts, within the organization. The mapping between the Concepts found and the existing one, with which the Ampersand implementation has a relationship, must be performed. Another form of sharing Concepts involves projects. Defined Concepts included in Patterns will be reused by other projects. The prototype is an HTML website in combination with CSS. When changes are made to this, knowledge of HTML and CSS is required. The extra functions that are not (yet) included in Ampersand can be made in PHP. So when using this, knowledge of PHP is required.

The question regarding Concepts, Relations and Rules, which appear in the **Wet-BIG**, can be referred to the appendix I. Here we find an overview of all these elements. A finding that emerges here concerns the embedding in the software architecture of the ICT organization. In the software architecture, the software components are managed and there is an overview of the relationships between these components. We can see the subsystems or patterns found as software components. It then appears that there is a certain degree of overlap of Concepts and Relations in the existing architecture and the model that Ampersand has made. Very careful analysis is needed to discover this overlap. The name of a Concept or Relation does not have to match, but the meaning does. It is also possible that the naming matches, but the meaning does not. In short, the existing software landscape needs to be carefully examined to determine which parts of the Ampersand model can be implemented. In the future, when multiple laws are analyzed to build registry systems, there may be pattern reuse. For the legal registers, no agreements in the form of data may be shared, so no data reuse. For example, the customer in the Donor Register may never be linked to a BIG registration with the aim of reusing customer data.

The question regarding the usefulness of the law for the Ampersand method has the following aspects.

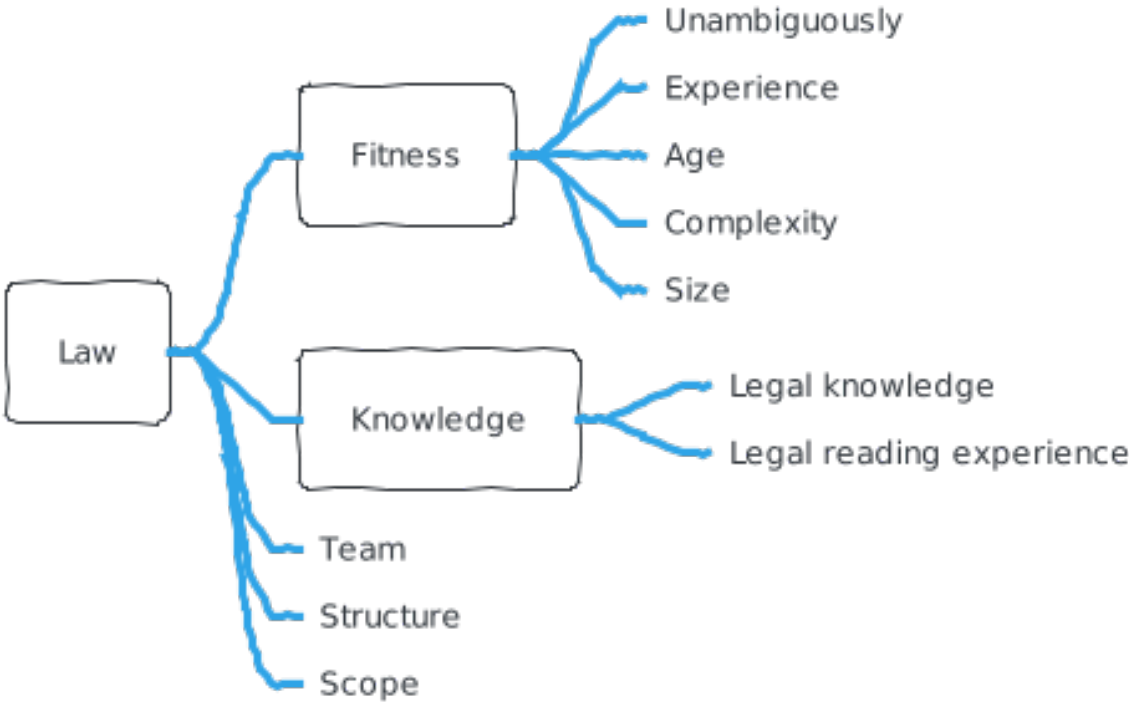


Figure 16: Mindmap law

We chose the **Wet-BIG** to analyze it with the Ampersand method. This law was chosen because there was a need from **CIBG** to redesign and rebuild the system that supports the law. With Ampersand we can do the redesign. The choice for

the law was made because this law met the requirement of redesign. During the analysis phase, we encountered a number of issues that do not support the choice and that a later choice of law should preferably comply with. For example, the law appears to be ambiguous on some points, according to the lawyer. Experience with the law is necessary to be able to properly analyze the law. This is especially true if the law, such as **Wet-BIG**, has options for interpretation. The antiquity of the original law may be a cause of interpretation. The complexity and scope of the law makes the analysis less straightforward. When we start with the legal analysis, start with a team. The team consists of at least one lawyer and two analysts. This guarantees legal knowledge and experience in reading and interpreting laws and regulations. The analysts have to keep each other on their toes when making the **Conceptual analysis**. At the start, we map out all relevant legislation and regulations and determine which legislation is included in the analysis. After the step, the structure is determined for each part and we probably have an idea how the system can look like. We assume here that the structure of the analysis will follow the structure of the law. The conclusion is not that when a law is ambiguous, complex, old and large, it cannot be analyzed by Ampersand, but it does make the trajectory difficult. One way to take is to be closer to the giver so that he is already aware when writing the law and considers the translation of the law into a registry system. One idea is that law would already be designed directly in relation to algebra. Then any ambiguity is gone. This just goes to show why it is necessary to team up with a lawyer. The lawyer can interpret the law and knows how to navigate the law.

What are the strengths and weaknesses of using Ampersand for registry systems in a government organization.

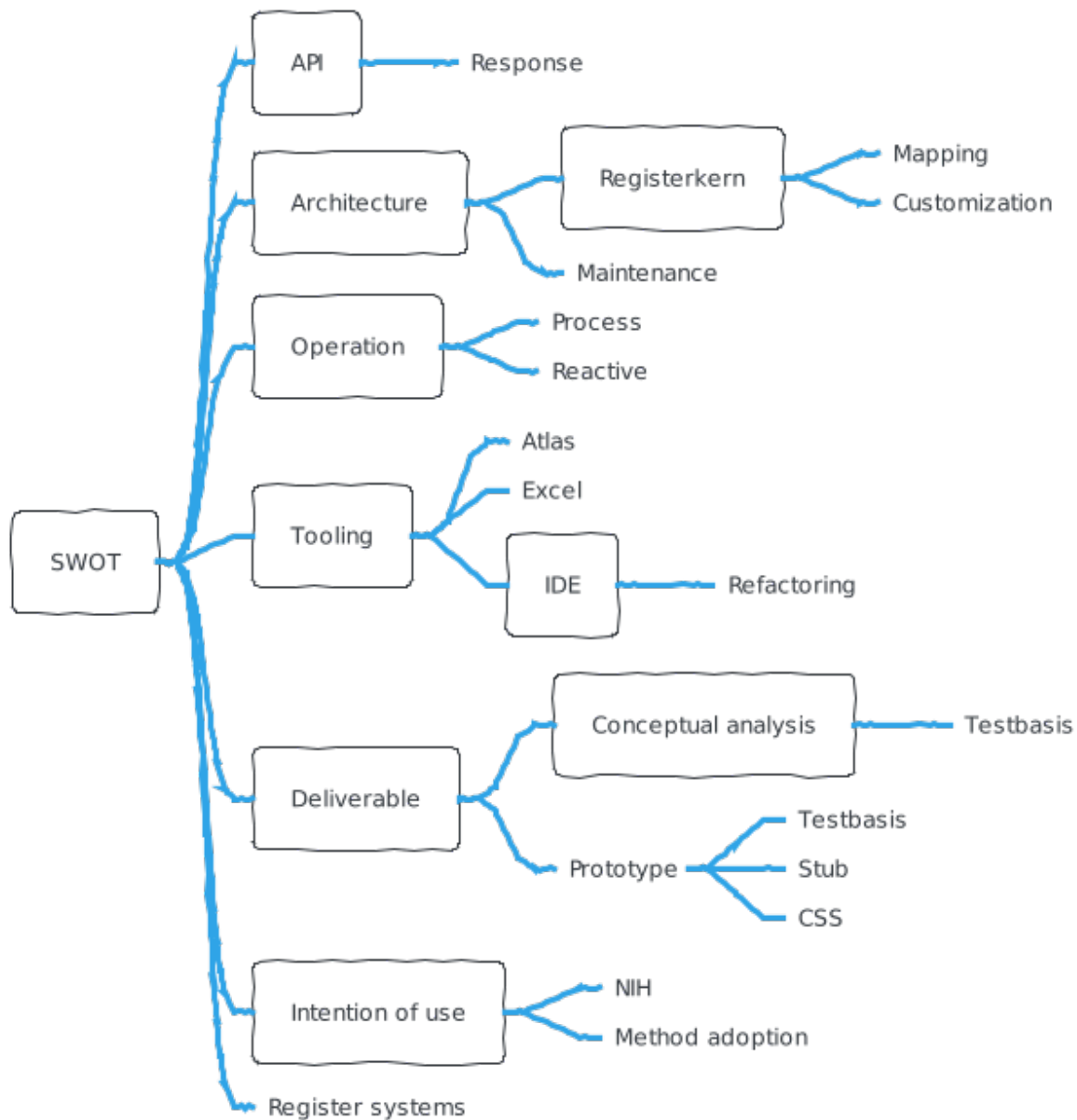


Figure 17: Mindmap swot

We can say that the analysis of a law can lead to a register system and because it is a register system, which is derived from the law, it will always be placed with a government organization. We also concluded that there are not many observations and comments about registry systems. Then it remains to map the strengths and weaknesses of Ampersand for a government organization and then specifically for the **CIBG**. From the perspective of weakness and strength we will go through all parts on the basis of figure 17.

API availability within Ampersand at the prototype stage and many systems use APIs to communicate with the source. The description of the APIs are missing and can be retrieved from the log. Pushing the description of the APIs to Swagger, for example, makes it easier to use the APIs. Adapting response from the API to

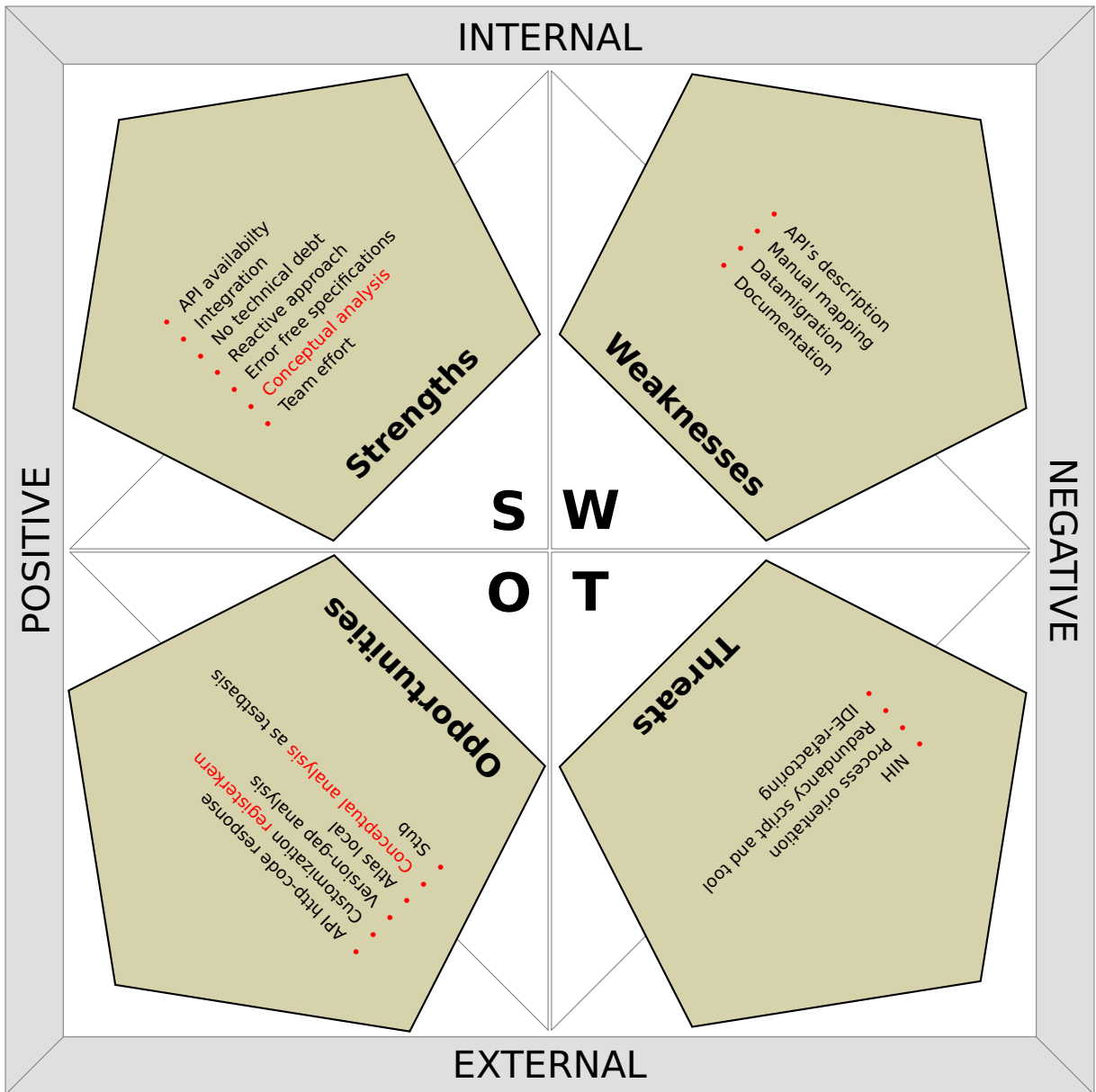


Figure 18: SWOT Ampersand for government organization

the calling system would be an improvement.

The mapping from Ampersand Concepts to **registerkern** is performed so that Ampersand analysis connects to **registerkern**, thereby integration takes place. This is a manual operation and can cause errors such that incorrect mappings take place or mapping does not take place.

The **registerkern** has a customization that makes it possible to place register values in the editable part. The mapping and customization will bring Ampersand and **registerkern** closer together.

The issue of maintenance on the Ampersand model has been discussed before. A strong point here is that after every maintenance a completely new model is created and no technical is introduced. But by always setting up a completely new system, it is now not possible to migrate the data. Ampersand systems are not used live, so the data conversion is only needed for the prototype environment.

Ampersand is a reactive designed system. The business rules actually define the process. The tool generates error-free specifications to support the business process. The **CIBG** is a strong process oriented organization.

The analyst needs an overview when managing the Concepts, Relations and Rules. Within RAP, the Atlas tool is available for this, but it is not available for the local environment. We worked with an excel sheet during the research, but this means that things are kept up to date twice and that is of course asking for problems. Within the IDE, the programming languages have refactoring tools at their disposal. We have been working with **VSC**, here the refactoring was not present and it happened that this caused inconsistency and the compile no longer ran correctly.

The **Conceptual analysis** is created as deliverable. This is used as a design for the implementation and because it is available early in the process, it can also be used as a validation tool and to base tests on. The prototype can also be used as a test basis and real tests can be performed on this. In combination with the API, the prototype can act in whole or in part as a stub.

For an organisation, a new method can be experienced as threatening. It is therefore possible that one reacts with an NIH action. To deal with this, it is wise to conduct an extensive POC and actively inform the parties. Assemble a team and use them as promoters. Non-ICT professionals can also be deployed as Ampersand modellers within the team, provided they have knowledge of Relation Algebra.

Ampersand should get a little more exposure than it is now just in the scientific environment. Then it will become more known and will be used more, making it more famous again. Now there is a certain reticence and that has the basis in the obscurity of Ampersand.

Overall conclusion is that Ampersand is a useful product for translating the law. The output products are very useful. Not all laws are equally suitable and the application of Ampersand in the development process must be incorporated. The latter still requires some mission work because it is different from what people are used to and it is very unknown. An organization will have to focus on using this and the organization is not very change-oriented.

## 7.2. Recommendation

Now that the research has been completed and the results and conclusions have been described, it is worth considering what further research could take place. The conclusions revealed that there are omissions in the area of maintaining an overview. In future research, attention could be given to the way in which this overview can be maintained. This may move in the direction of annotation tools. The follow-up research could focus on selecting and implementing tools within Ampersand for the purpose of maintaining overview. An overview is also needed on the Concepts management side. This seems to be possible by porting Atlas from RAP to the local environment.

Another conclusion that has been drawn is that the **Wet-BIG** is not the most suitable law to analyze it via the Ampersand method. This is not Ampersand's fault, but the law. It is interesting to map out which requirements the law must meet in order to fit in well with the method and the follow-up is to examine how we can shape future laws that people would like to be supported by (register) systems so that they can be quickly analyzed by Ampersand. Early participation in the legislative process could save a lot of time and money. How much that could be, is of course fodder for a research project.

One of the interviewees feared that the Ampersand approach would take more time in the design phase than the regular approach. The regular approach includes a more or less agile approach, in which the design is made in outline. After which the system is cut into pieces and these are created agile. One could research the design of two comparable systems or possibly even the same system, with one done the Ampersand way and the other the regular way. It is then interesting to see which is faster, more complete and more workable for the follow-up process.

Another comment made during the interviews relates to the size of the system. The hypothesis was that the system size of an Ampersand project will be smaller than the size of a system from a regular trajectory. This could be related to the fact that Ampersand is directly on the source and does not want to include all kinds of peripheral matters.

In the context of maintaining the overview, it has been suggested to make use of the addition of XML in the source document. So enriching the source document with the annotation XML. The big advantage of this would be that it is then possible to generate the Ampersand script. Especially after changes in the source document, where the existing annotations can be inserted in the new version. This adjustment would be even more beneficial once there is an existing national base of Concepts and Relations. The link between annotations and the national base could result in an enormous acceleration in development. This is worth investigating, but will have to be split into several studies.

During the research we were regularly confronted with the **registerkern**. It is worth investigating how exactly this link should be established. Where are the similarities and where are the differences? In this context we again come across the issue of the common Concepts.



## 8. Reflection

The conclusions have been drawn and recommendations have been made on the basis of these. Now is the time to reflect on the quality of the research and the validity of the conclusions. My research has yielded a lot of observations made from my perspective. From my IT perspective I have looked at the Ampersand tool and method. At the start of the research, I was very focused on how Ampersand works, so I started looking at the use of Ampersand and that's what the first observations are about. My observations therefore have the character of a testing Ampersand. Anything missing or differs from what I would expect is noted.

At a later stage, when the first pieces of the prototype were running, I focused on the **Conceptual analysis**. There I was discovering how this had to be built and how I had to deal with the scripts to get a logical story from it. Building on the **Conceptual analysis** naturally also affects the prototype.

In order to make **Conceptual analysis** the law had to be analyzed as well. I got a lot less far with this than I would have liked. That's because I had to get through the first two phases if I wanted to be able to perform this phase. And then it turns out that the law is difficult to understand. The danger I noticed was that I suffered from bias. Because of the approach in the form of **action research** I was in the middle of the research. Partly due to the text that is difficult to comprehend and the knowledge of **BIG-registration System**, the legal text was not always looked closely enough and sometimes a shape was given to it based on our own knowledge. We failed to correct this in all cases in the **Conceptual analysis**.

After building part of the prototype and also having a version of the **Conceptual analysis**, I started talking with some colleagues. The prototype was not always well understood, but the **Conceptual analysis** was perceived as recognizable. The prototype was in the different form than people were used to and the terminology is not based on the **Wet-BIG** at all. These are sometimes different terms than are used within the register systems.

The research focused on the usefulness of Ampersand for use as a design tool for registry systems. I have sufficiently demonstrated that the usefulness of Ampersand as a method and tool is fine. Whether it is also useful for the organization depends on the will to use it.

## 9. Acknowledgements

This thesis was made possible with the help of many people. First of all I would like to thank Stef Joosten and the Ampersand team. Stef guided me through this phase. I found in another thesis in which Stef was described as a fantastic supervisor, knowledgeable, supportive, enthusiastic, and very approachable. I wholeheartedly agree with this. When I didn't know what to do anymore, he knows how to motivate me and shows me the way to continue. I also wish other students such a pleasant and knowledgeable supervisor.

I would also like to thank Jasper Kuijten for guidance from CIBG and for pointing out the research direction and the regular discussions about this. With Stef and Jasper I got through the research period and here is the product.

Of course I would like to thank all the people who were bothered by the fact that I needed so much time for my studies. I took time off from work very regularly and part of my task fell on the shoulders of colleagues.

I certainly couldn't have done this without the support of my own family. My wife, Fieke, who always gave me space and encouraged me to continue. Sophie, Philip and Sarah who also always had to listen to me about the study and research. Thank you for reading the thesis. I really appreciate the space and mental support that has been given.

Gerard Edelaar, Voorhout - April 2022

## Bibliography

- David Antons, Mathieu Declerck, Kathleen Diener, Iring Koch, and Frank T. Piller. Assessing the not-invented-here syndrome: Development and validation of implicit and explicit measurements. *Journal of Organizational Behavior*, 38(8):1227–1245, 2017. ISSN 1099-1379. doi: 10.1002/job.2199. URL <http://onlinelibrary.wiley.com/doi/abs/10.1002/job.2199>. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/job.2199>.
- Engineer Bainomugisha, Andoni Lombide Carreton, Tom van Cutsem, Stijn Mostinckx, and Wolfgang de Meuter. A survey on reactive programming. *ACM Comput. Surv.*, 45(4):52:1–52:34, August 2013. ISSN 0360-0300. doi: 10.1145/2501654.2501666. URL <http://doi.org/10.1145/2501654.2501666>.
- Inger Johanne Bakken, Anja M. S. Ariansen, Gun Peggy Knudsen, Knut Ivar Johansen, and Stein Emil Vollset. The norwegian patient registry and the norwegian registry for primary health care: Research potential of two nationwide health-care registries:. *Scandinavian Journal of Public Health*, 2019. doi: 10.1177/1403494819859737. URL <http://journals.sagepub.com/doi/10.1177/1403494819859737>. Publisher: SAGE PublicationsSage UK: London, England.
- Nancy Carter, Denise Bryant-Lukosius, Alba DiCenso, Jennifer Blythe, and Alan J. Neville. The use of triangulation in qualitative research. *Oncol Nurs Forum*, 41(5):545–547, September 2014. ISSN 1538-0688. doi: 10.1188/14.ONF.545-547.
- E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970. ISSN 0001-0782. doi: 10.1145/362384.362685. URL <http://doi.org/10.1145/362384.362685>.
- Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting Empirical Methods for Software Engineering Research. In Forrest Shull, Janice Singer, and Dag I. K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer London, London, 2008. ISBN 978-1-84800-043-8 978-1-84800-044-5. doi: 10.1007/978-1-84800-044-5\_11. URL [http://link.springer.com/10.1007/978-1-84800-044-5\\_11](http://link.springer.com/10.1007/978-1-84800-044-5_11).
- Conal Elliott and Paul Hudak. Functional reactive animation. *SIGPLAN Not.*, 32(8):263–273, August 1997. ISSN 0362-1340. doi: 10.1145/258949.258973. URL <http://doi.org/10.1145/258949.258973>.
- Jillian Farquhar, Nicolette Michels, and Julie Robson. Triangulation in industrial qualitative case study research: Widening the scope. *Industrial Marketing Management*, 87:160–170, May 2020. ISSN 0019-8501. doi: 10.1016/j.indmarman.2020.02.001. URL <https://www.sciencedirect.com/science/article/pii/S0019850118306916>.

- Kasper Hornbæk. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64(2):79–102, 2006. ISSN 1071-5819. doi: <https://doi.org/10.1016/j.ijhcs.2005.06.002>. URL <https://www.sciencedirect.com/science/article/pii/S1071581905001138>.
- Hsiu-Fang Hsieh and Sarah E. Shannon. Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9):1277–1288, 2005. ISSN 1049-7323. doi: [10.1177/1049732305276687](https://doi.org/10.1177/1049732305276687). URL <https://doi.org/10.1177/1049732305276687>. Publisher: SAGE Publications Inc.
- Stef Joosten. Software Development in Relation Algebra with Ampersand. In Peter Höfner, Damien Pous, and Georg Struth, editors, *Relational and Algebraic Methods in Computer Science*, pages 177–192, Cham, 2017. Springer International Publishing. ISBN 978-3-319-57418-9.
- Stef Joosten. Relation Algebra as programming language using the Ampersand compiler. *Journal of Logical and Algebraic Methods in Programming*, 100:113–129, November 2018. ISSN 23522208. doi: [10.1016/j.jlamp.2018.04.002](https://doi.org/10.1016/j.jlamp.2018.04.002). URL <https://linkinghub.elsevier.com/retrieve/pii/S2352220817301499>.
- Florian Kohlbacher. The use of qualitative content analysis in case study research. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 7(1), 2006. ISSN 1438-5627. doi: [10.17169/fqs-7.1.75](https://doi.org/10.17169/fqs-7.1.75). URL <https://www.qualitative-research.net/index.php/fqs/article/view/75>. Number: 1.
- Vinay Kulkarni and Sreedhar Reddy. An abstraction for reusable MDD components: model-based generation of model-based code generators. In *Proceedings of the 7th international conference on Generative programming and component engineering - GPCE '08*, page 181. ACM Press, 2008. ISBN 978-1-60558-267-2. doi: [10.1145/1449913.1449940](https://doi.org/10.1145/1449913.1449940). URL <http://portal.acm.org/citation.cfm?doid=1449913.1449940>.
- Roger D. Maddux. Bibliography. In Roger D. Maddux, editor, *Relation Algebras*, volume 150 of *Studies in Logic and the Foundations of Mathematics*, pages 713–722. Elsevier, 2006. doi: [https://doi.org/10.1016/S0049-237X\(06\)80031-5](https://doi.org/10.1016/S0049-237X(06)80031-5). URL <https://www.sciencedirect.com/science/article/pii/S0049237X06800315>. ISSN: 0049-237X.
- Philipp Mayring. Qualitative content analysis. *Forum: Qualitative Social Research*, 1(2), 2000. ISSN 1438-5627. URL <http://www.qualitative-research.net/index.php/fqs/article/view/1089>. Publisher: FQS.
- Philipp Mayring. Qualitative inhaltsanalyse – abgrenzungen, spielarten, weiterentwicklungen. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, Vol 20:No 3 (2019): Qualitative Content Analysis I, 2019.

doi: 10.17169/FQS-20.3.3343. URL <http://www.qualitative-research.net/index.php/fqs/article/view/3343>. Publisher: Forum Qualitative Sozialforschung / Forum: Qualitative Social Research.

Gerard Michels, Sebastiaan Joosten, Jaap van der Woude, and Stef Joosten. Amper-sand: Applying relation algebra in practice. In Harrie de Swart, editor, *Relational and Algebraic Methods in Computer Science*, volume 6663, pages 280–293. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-21069-3 978-3-642-21070-9. doi: 10.1007/978-3-642-21070-9\_21. URL [http://link.springer.com/10.1007/978-3-642-21070-9\\_21](http://link.springer.com/10.1007/978-3-642-21070-9_21). Series Title: Lecture Notes in Computer Science.

Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empir Software Eng*, 14(2):131, December 2008. ISSN 1573-7616. doi: 10.1007/s10664-008-9102-8. URL <https://doi.org/10.1007/s10664-008-9102-8>.

Morten Schmidt, Sigrun Alba Johannesdottir Schmidt, Jakob Lynge Sandegaard, Vera Ehrenstein, Lars Pedersen, and Henrik Toft Sørensen. The danish national patient registry: a review of content, data quality, and research potential. *Clinical Epidemiology*, 7:449, 2015. doi: 10.2147/CLEP.S91125. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4655913/>. Publisher: Dove Press.

Brian Shackel. Usability – context, framework, definition, design and evaluation. *Interacting with Computers*, 21(5):339–346, 2009. ISSN 0953-5438. doi: 10.1016/j.intcom.2009.04.007. URL <https://doi.org/10.1016/j.intcom.2009.04.007>. \_eprint: <https://academic.oup.com/iwc/article-pdf/21/5-6/339/2247629/iwc21-0339.pdf>.

## Non-published Readings

Wet op de beroepen in de individuele gezondheidszorg. <https://wetten.overheid.nl/1993-2021>. URL <https://wetten.overheid.nl/BWBR0006251/2021-07-01>.

The Reactive Manifesto, September 2014. URL <https://www.reactivemanifesto.org/>.

Anouschka Ausems, John Bulles, and Mariette Lokin. *Wetsanalyse: een aanpak voor effectieve en uitlegbare toepassing van wetgeving*. 2021. OCLC: 1255476721.

Elleke Baecke. Argument assistance software in legal reasoning.pdf, March 2018.

Belastingdienst. webinar Wetsanalyse, June 2021. URL <https://www.onlineseminar.nl/belastingdienst/webinar/41348/webinar-wetsanalyse>.

Pim Bos. Bedrijfsregels in verschillende vormen, 2013.

Jet Bussemaker. de-b-van-bekwaam-naar-een-toekomstbestendige-wet-big.pdf, 2019.

Francis de Kok, Stef van Dyck, Gerard Edelaar, and Marco Zwolve. Analyse rapport Applicatie Lifecycle Management BIG-register, May 2019.

S Joosten and R Joosten. Will Rule based BPM obliterate Process Models? page 14, 01 2007.

Tim koopman. ADEQUATE COMPLETENESS OF DATA MARTS COMPARED TO INFORMATION REQUESTS.pdf, June 2014.

L. Wedemeijer, S.M.M. Joosten, G. Michels, and J.L.C. Arkenbout. Werkboek ontwerpen met bedrijfsregels.pdf, 2013a.

Lex Wedemeijer, Stef Joosten, and Jaap van der Woude. Rule-Based Design.pdf, 2013b.

## A. Observations

### Ampersand

**observation 1** (rq1-13:17-10: The setup of Ampersand in local environment is specific and not self-explanatory. Help is needed here to get this working. Attempts to get the process working in localhost were unsuccessful. The manual on the Ampersand site showed how to do this. But it still didn't work). **obs:** *the documentation gives an indication of how to configure ampersand and make it work locally. Using XAMPP, but this is not going to work. Not clear why and it finally did work in a Docker environment.*

**observation 2** (rq1-18 Can not find an example on the internet, only in the repo of Ampersand itself. That is difficult to find.). **obs:** *Little to be found about Ampersand except in its own repos.*

**observation 3** (rq1-45:24-10: Overview within an Ampersand script is difficult to obtain.). **obs:** *The need for overview is there as the script grows.*

**observation 4** (rq1-2 Ampersand has no annotation option, therefore requires a separate action or document to keep track of what has been passed.). **obs:** *There is a need to maintain an overview. Hence the annotation option.*

**observation 5** (rq1-47:27-10: Detecting a bug. Placing these in github issues at the Ampersand repository will get a response within a day and resolve it. In this case it was a bug in Ampersand that was quickly fixed with a new version.). **obs:** *Quick fix of a bug in Ampersand by the development team.*

**observation 6** (rq1-60:9-11: Training and education is required to write an Ampersand script.). **obs:** *There should always be someone with experience in the background or in the collaboration.*

**observation 7** (rq1-96:30-12: Skill in scripting within Ampersand is quickly lost if you don't do this frequently.). **obs:** *Practice a lot and keep using it.*

**observation 8** (rq2-19:16-11: Ampersand returns constraints and no executable). **obs:** *It is not an executable file, but a collection of database constraints that is the core of Ampersand.*

**observation 9** (rq4-1 Ampersand cannot calculate. But since Ampersand is static, process data can be monitored in other ways.). **obs:** *Ampersand cannot calculate.*

**observation 10** (rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)). **obs:** *When a new model is created within Ampersand, the data structure is reloaded. There are no provisions for preserving the data that has already been entered.*

**observation 11** (rq4-8:22-11: The team behind Ampersand is very dedicated.). **obs:** *Calls are resolved quickly. Example was the error message on wrong multiplicity (notification <https://github.com/AmpersandTarski/RAP/issues/128>) Reloading a new version is not easy.*

## Api

**observation 12** (rq1-8:14-11: No swagger is created for the api;). **obs:** *If you want to use an external input, API descriptions are very relevant. These are not generated automatically.*

**observation 13** (rq1-70:14-11: Postman works with api/v1/resource, e.g. GET <localhost/api/v1/resource/Person/P001/Person>, retrieves that of an existing person. So the validation structure of ampersand can be used from outside Ampersand by means of api.). **obs:** *Ampersand is more open than it first appears.*

**observation 14** (rq1-72:14-11: Besides the GET(get), the POST(append) and PUT (mutate) also work). **obs:** *Using Postman, the api features were tested.*

**observation 15** (rq1-73:14-11: Ampersand can be used from other applications through APIs, but the return values are next to the requested information also messages and not message codes. These codes could be included in the reports, but now remain "unstructured" data.). **obs:** *Here you are missing the structure of the responses. So Ampersand is apparently not intended to be used in this way. See also note on swagger(rq1-8).*

**observation 16** (rq1-74:16-11: Link between an external front-end and an Ampersand back-end (Ampersandapi). A change in the back-end, so an Ampersand change, then the front end almost certainly has to change with it.). **obs:** *Forced maintenance of the external front-end due to changes within Ampersand APIs.*

**observation 17** (rq4-2 The api link works fine, but entire messages return. These should actually get codes.). **obs:** *An api returned a text. Calling applications usually don't handle that very well. It is usual to return a code and sometimes with text. Think of http response codes.*

**observation 18** (rq4-5 Postman used for api link with Ampersand.). **obs:** *As a test it is possible to use Postman for the link. So it is not necessary to build an application for this.*

## Architecture

**observation 19** (rq1-62:10-11: There has be the architecture link between the law core and the register core). **obs:** *Ampersand analysis must fit the architecture of the organization and the way of working.*

**observation 20** (rq4-3 Embedding in architecture, the core of the law with shared concepts and processes. The core of law is specific law. Shared concepts are also part of the law but also occur elsewhere. This is part of embedding in architecture.). **obs:** *Embedding into the existing architecture is important for usability, in the form of acceptance.*



## Classify

**observation 21** (rq1-86:30-11: Classify is a specialization of a concept. No experience has been gained with this.). **obs:** *There was no place for this in the research.*

## Concept

**observation 22** (rq1-16 Notation method of Concept and Relations and Rules are defined for a very small part. Only the first position is uppercase or lowercase. There is no rule about other spelling. So using CamelCase or underscore or hyphen.). **obs:** *You are not forced to work in any particular structure. There is no need for coercion in this area, but advice is practical for novice users.*

**observation 23** (rq1-79:20-11: Once a concept for a date or other element is defined, it can be used anywhere in the context. The question then is how to deal with shared Concepts and how to manage them.). **obs:** *Within the context, a concept is reused. The operation of shared concept within other contexts is not self-evident.*

**observation 24** (rq1-30:12-9: Defining the meaning and definition of the concept is free of rules. There is no fixed pattern for documentation.). **obs:** *Defining the meaning and definition is very free.*

**observation 25** (rq1-42:19-10: Immediately add the description when recording a concept and relation. Later it is difficult to find out why the recording took place.). **obs:** *To avoid rework, the definition and meaning and purpose should be defined immediately when defining concepts and relationships.*

**observation 26** (rq1-48:27-10: The concept current date is solved very complicated. But eventually it works. Current time does not seem to have developed yet. Although the example scripts seem to say something different.). **obs:** *A frequently used element like date and time is not easily solved in Ampersand.*

**observation 27** (rq1-46:24-10: There is no find able relationship between the relation and the concept in the script.). **obs:** *The overview where a concept is used is difficult to obtain. The IDE used also does not provide any tooling to obtain this overview.*

**observation 28** (rq1-80:20-11: A consistent naming of a concept is necessary.). **obs:** *A once defined concept could just be redefined (due to lack of overview). With just a different format or definition.*

**observation 29** (rq1-84:30-11: A concept is immutable. for example a person is concept, not doctor. It must be an intrinsic property, which cannot be changed.). **obs:** *The important property of concept within Ampersand.*

**observation 30** (rq1-89:7-12: Items named as common concepts.). **obs:** *There are common elements across registers. The question is how to address this commonality.*

**observation 31** (rq1-91:14-12: A concept and a relation can be defined several times within your own patterns. So that the patterns can stand on their own.). **obs:** *Dangerous because it allows the same concepts to have different definitions. This behaviour shows up in the **conceptual analysis**.*

**observation 32** (rq2-7:4-10: A concept Person is not equal to BIG-number. A big number is an attribute of the registration. A person can have multiple BIG-numbers.). **obs:** *It seems in the text that a BIG-number is equated with a person.*

**observation 33** (rq3-2 By reading the law, a structure becomes clear. The concept Person, Registration and Registration with management and Discipline(Discipline with measures.). **obs:** *The main lines of the law seem clear to a non-lawyer.*

**observation 34** (rq1-95:29-12: The format of a concept big number is not included in the law). **obs:** *Should there be requirements for the big number now I think it is 8 digits, but there should be. A guid may not be very useful for user-friendliness.*

**observation 35** (rq2-17:10-11: A dutch person has an concept address that must conform to the BRP format (should be a standard building block for it!). A foreign address is unclear what to do with this.). **obs:** *It is unclear how to handle addresses.*

**observation 36** (rq3-14:19-10: A concept person and a big number are very different things. Person is immutable, big number is not. They do have a relation with each other.). **obs:** *In speech, these are sometimes used interchangeably, so that it seems that a person is equivalent to a big number.*

**observation 37** (rq2-1 Substantively includes **Wet-BIG** also includes disciplinary law (tuchtrecht), which is another branch of sport.). **obs:** *Disciplinary law is not easy to capture in concept and relations. Disciplinary law consists more of processes and procedures.*

## **Conceptual analysis**

**observation 38** (rq1-1 Formatting in Ampersand (patterns) has consequences for the Conceptual analysis.). **obs:** ***\*\*welke dan ?? nog even over nadenken\*\****

**observation 39** (rq1-43:23-10: The order of the data in the Conceptual analysis is a bit strange. First the definition is shown, then the name of the relation and below that the meaning again.). **obs:** *The layout of the Conceptual design doesn't seem quite logical and is therefore confusing.*

**observation 40** (rq1-44:23-10: In the Conceptual analysis enters must be taken into account in the texts. These come back directly in the documents and then yield broken sentences.). **obs:** *Break enters in the **IDE** also produce extra newlines in the output. This causes the formatting to go wrong.*

**observation 41** (rq1-76:20-11: The "disclaimer" does not appear in the Conceptual analysis.). **obs:** *The "disclaimer" does not appear in the Conceptual analysis.*

**observation 42** (rq1-99:6-1: when generating a Conceptual analysis the doc gets the name of the first concept.). **obs:** *The name of the generated document will be the name of the first draft contained in the document.*

**observation 43** (rq1-35:14-9: The law has been drawn up in Dutch, which means that the Conceptual analysis can also be done in Dutch.). **obs:** *The starting point is to make the Conceptual analysis in Dutch.*

**observation 44** (rq1-51:2-11: Discussing the Conceptual analysis should be done theme by theme.). **obs:** *Where a theme equals pattern.*

**observation 45** (rq2-4:30-9: Which agreements must be made regarding the structure of the descriptions for Conceptual analysis. Do agreements have to be made about it or leave it unstructured?). **obs:** *To prevent the description from becoming a mess, agreements (implicit or explicit) must be made about the way of describing and name the references.*

**observation 46** (rq2-18:16-11: Good to realize that the meaning you write down also ends up in the Conceptual analysis. So looking at the way of writing it down can form a story in the analysis.). **obs:** *The meaning must be worded in such a way that all these meanings form a story.*

## **Docker**

**observation 47** (rq1-11 Implementation in Docker with RAP creates new directories all the time.). **obs:** *The Docker environment is polluted by adding new directories all the time. This makes analysis difficult because it is not clear which directory is used.*

**observation 48** (rq1-6:21-10: Docker is also another thing to learn. There should also be an introductory course to quickly understand Docker usage for Ampersand. A waste of time to have to look this up yourself or it is preconditions to be able to use Ampersand.). **obs:** *Docker knowledge (limited) is required*

## **Documentation**

**observation 49** (rq1-78:20-11: The documentation generated in HTML loaded in firefox and no PNG's are visible. Chrome is doing well.). **obs:** *Firefox does not show the generated models.*

**observation 50** (rq1-41:19-10: The "wettenbank" website contains a persistent hyperlink, which can be used in the documentation as reference.). **obs:** *References to persistent links can be included. But is the output still pleasant to read because of the continuous references.*

**observation 51** (rq1-54:2-11: The documentation can be written in different ways. This can be done using mark down, html and latex.). **obs:** *You will encounter this in usage, even though the documentation states this as well.*

**observation 52** (rq1-97:30-12: By puzzling with Ampersand people quickly forget to make correct documentation. Often you are happy that something works.). **obs:** *Too much trying and figuring distracts from documenting.*

**observation 53** (rq1-75:20-11: Some more experimentation with the documentation in the prototype. When describing the purpose of the context, it takes a while to figure out how this text can be properly conveyed. An <h1> results in an extra chapter in H4 and H4 then becomes H5 and H5 has then become a meaningless piece. With an <h2> and <h3> it works well.). **obs:** *Interfering with structure can have unexpected consequences.*

**observation 54** (rq3-7 Adding documentation with the correct description to a concept and relation is not so easy. Easy to stray and add your own interpretation.). **obs:** *While drafting concepts and relationships, a description of the position where the element comes from must be immediately included. This doesn't always happen because the scripting language keeps you so busy (a lot of messing around) that you forget to add the text.*

## **Flexible**

**observation 55** (rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses.). **obs:** *Ampersand is very flexible. Define a Concept and relationship and it is realized. Second observation is that in the case of the address it is not immediately clear what this should look like. But there are other sources for that. It takes some searching and making assumptions.*

## **Include**

**observation 56** (rq1-81:20-11: Compilation error due to a include that no longer existed. Observation here is that an adl has been renamed or moved or deleted. The tool **Visual Studio Code** does not support a refactoring stroke on said changes.). **obs:** *Refactoring is not supported with **Visual Studio Code**.*

**observation 57** (rq1-15:4-10 With include statements the order of the contents of the document is determined. The expectation was that includes are needed to link parts of code together but includes are not everywhere necessary to get the code working.). **obs:** *Includes are not only to run the scripts completely, but also to send the documentation.*

**observation 58** (rq1-82:20-11: include don't always seem necessary on compilation. It is not entirely clear when this is necessary or not. Another function of includes is to format the analysis.). **obs:** *Includes are useful and necessary, but it's not always clear how to use them.*

**observation 59** (rq1-90:14-12: Collection model of regulations than by means of includes keep it small and therefore clear. This is for the reusability of the script. One module per feature.). **obs:** *Small modules with reusability in mind.*

## Interface

**observation 60** (rq1-12: At the start it is not clear when a capital letter or small letter should be used with the crud in the interface.). **obs:** *It is in the manual<sup>27</sup>, but you have to find out by trial and error how it really works.*

**observation 61** (rq1-40:10-10: The concepts used in the interface must be of type "object" (represent). The concept may therefore not be alpha or integer.). **obs:** *Interface did not start correctly. This was caused by the interface concept not being of type "object".*

**observation 62** (rq1-53:2-11: The crud (**Create, Read, Use, Delete**) and **CRUD** in the interface don't always work as it should be. There is no full validation on usage. So an on/off does not make sense everywhere. rq1-37:3-10: CRUD/crud options also need some study before they can be applied properly.). **obs:** *No (full) validation on the use of crud. It is possible to apply variations that have no impact.*

**observation 63** (rq1-58:8-11: Per interface max one multiplicity, otherwise you won't get data stored.). **obs:** *Within an interface, multiple total constraints were included in the relationships. The result was that no more data could be added within the prototype.*

**observation 64** (rq1-71:14-11: The interface also belongs to the design and not just to the prototype. Changing the **Create, Read, Use, Delete** changes the behavior of the API.). **obs:** *API behavior changes by changing **CRUD**.*

**observation 65** (rq1-83:27-11: Experiment with HTML view within the interface fails. Documentation of this is not conclusive. The examples are not enough). **obs:** *This part was not made to work.*

**observation 66** (rq1-98:30-12: When using linkto in the interface as last element in the interface and the signature occurs more often than a dropdown to all subinterfaces (of the same signature) appears.). **obs:** *Unexpected behavior of the **LINKTO**.*

---

<sup>27</sup><https://ampersandtarski.gitbook.io/documentation/the-language-ampersand/services/crud>

**observation 67** (rq1-5:30-10: The browser is holding data from the interface and periodically the cache needs to be cleared for customization to work.). **obs:** *It looks like the changes made to the script don't affect operation. It is caused by the browser's cache not being emptied automatically. There are browser extensions to still do this manually.*

**observation 68** (rq1-10 The function HTML href with target blank does not work within the interface rq1-77:20-11: In HTML mode the <a href="x" target=\_blank> is not supported. The target is removed in the compilation.). **obs:** *The expectation was that the target \_blank would open a new tab in the HTML text, but that does not happen.*

**observation 69** (rq2-12:19-10: TOT has the property that this must be entered in the interface because otherwise the data will not be saved. A variant of this is an rule with this property As a result, the other items are stored in the database, but a notification of incompleteness continues to appear.). **obs:** *so there are several ways to deal with a TOT. Therefore, the use of this resource must also be considered.*

**observation 70** (rq1-21:7-11: TOT is usually overcome by a tot-rule, it turns out that a TOT causes something to be saved when entered, while a tot-rule allows a save to occur while the notification remains open to stand.). **obs:** *so there are several ways to deal with a TOT. Therefore, the use of this resource must also be considered.*

**observation 71** (rq2-15:19-10: In the interface a FOR can also be used. This populates user roles.). **obs:** *So be authorized can be arranged here. The question is how this works in, for example, a combination of api with FOR.*

**observation 72** (rq4-4 The interface produces many messages and these remain.). **obs:** *Prototype screens fill up with messages when they are not resolved.*

## Latex

**observation 73** (rq1-33:9-1: VSC does not support the latex environment well. My PC often hangs on this. rq1-87:3-12: Latex can also be written in VSC. Apparently it is a different version, because the import does not immediately succeed. Does not work really well and the result is poor.). **obs:** *Visual Studio Code also supports the TEX environment through add-ons. But this add-on completely hangs my system. I got a 100% cpu load for a long time.*

## Law

**observation 74** (rq3-3:12-9 There are parts of the law that are no longer valid, they are not included). **obs:** *The law is quite complex, and it is possible to go back in time. The choice that has been made is not to go back in time within this scope.*

**observation 75** (rq3-4:12-9 There are more laws involved than just the **Wet-BIG**. rq3-6 12-9 In addition to the law, decisions are also important.). **obs:** *The law website contains references to other laws and regulations.*

**observation 76** (rq1-42:21-10: It is easy to deviate from the legal texts. Because they are so hard to read. Some knowledge of the law or the process means that your own interpretation is quickly made. Action research also means that you quickly fall into this trap.). **obs:** *Due to complex texts, there is a danger that knowledge is trusted on your own background.*

**observation 77** (rq1-29:12-9: Not all law- and regulations using **Wet-BIG** can be found under the search term "big"). **obs:** *There is more than just **Wet-BIG**.*

**observation 78** (rq1-26:12-9: Also the laws and the regulations can still have references to other laws and regulations. Because they can be based on these laws or extend it.). **obs:** *Scoping is important.*

**observation 79** (rq1-27:12-9: There are also laws and regulations that are not included in this particular law, but are valid from a higher law (implicit references). In case of **Wet-BIG** this could be eg the Archives Act or the Time Limits Act and Criminal Law.). **obs:** *To get a complete overview of laws and regulations, the help of a lawyer is needed.*

**observation 80** (rq1-23:24-10: law Reading is a skill.). **obs:** *The law consists of jargon and you need a lawyer for that.*

**observation 81** (rq3-16:24-10: For the Netherlands, we have a country table from the RvIG. These are nationally established and maintained tables. No maintenance function is therefore required.). **obs:** *There is more than the law.*

**observation 82** (rq3-15:24-10: In the law the nationality is mentioned, it also refers to the EU and non-eu residents. It is not recognized that the nationality definition is defined per country.). **obs:** *This is a limitation of the law.*

**observation 83** (rq3-10:12-10: Formatting of the name is not stated literally in the law, but must conform to BRP standards.). **obs:** *Is it relevant that this is not in there. This could be enforced elsewhere than in Ampersand. Input validations at a front-end system.*

**observation 84** (rq3-11:12-10: Matters such as authorization decisions that allow an information system to retrieve BRP data are not found in the law.). **obs:** *The law does not focus on the translation to ICT.*

**observation 85** (rq1-25:12-9: First make overview of all laws and regulations). **obs:** *Scoping is important.*

## **Multiplicity**

**observation 86** (rq1-66:10-11: XLSX files format is created partly on the basis of multiplicity. one on n relation produces its own tab.). **obs:** *The Excel file is a reflection of the database structure so that insight can be obtained in the database structure.*

**observation 87** (rq2-5:2-10: Making the multiplicity explicit.). **obs:** *Since you don't always have a clear picture of how this works, it needs to be written out to make it workable.*

UNI	P->0-1 H	most
TOT	P->1-* H	least
INJ	H->1 P	one
SUR	H->1-* P	at least 1

**observation 88** (rq1-3 Created a separate excel to write out and discover the multiplicity of the relations.). **obs:** *As a method this is a clear way. Did notice that it is difficult (from a management perspective) to keep the Excel document in sync with the scripts.*

**observation 89** (rq2-2 Only UNI, TOT, INJ and SUR are used.). **obs:** *Although there are more forms of multiplicity, in practice (also in the examples) UNI and TOT are mainly used. To a lesser extent INJ and SUR.*

**observation 90** (rq2-13:19-10: What applies to multiplicity TOT, also applies to SUR. ). **obs:** *There are several ways to deal with a SUR. Therefore, the use of this resource must also be considered.*

## Obsedian

**observation 91** (rq1-88:5-12: Tried the tool Obsidian as a new tool. But here too I do not get an immediate overview and it is digital. Apparently writing in a log is more convenient for me). **obs:** *Also tried a new tool while writing the logs. Either this one does not work for me or I need to be more patient.*

## Pattern

**observation 92** (rq1-49:30-10: Isolating a pattern or subsystem for testing does not work. This has to do with setting up Docker and possible ignorance on my part.). **obs:** *The goal was to put a part of the system on its own so that only that part could be tested. But due to the Docker setup, this doesn't seem possible. Or I don't have enough knowledge of Docker to make this possible.*

**observation 93** (rq1-33:14-9: The use of patterns within Ampersand is important. These are the subsystems of the information system. The question is whether this should be classified in advance or whether it builds up on its own.). **obs:** *Use of patterns is necessary for the subsystem layout.*

**observation 94** (rq1-34:14-9: The spelling of a pattern is capitalized and the pattern ends with an end-pattern. Multiple patterns are possible within one script.). **obs:** *It's in the documentation, but you read about it. It must happen to you.*

**observation 95** (rq1-38:3-10: Should the subsystems be mapped in advance.). *This is controlled via patterns.* **obs:** *It is not necessary to divide the analysis in advance into patterns or subsystems. It is possible, but then there must already be a good picture of the text.*



## PDF, RTF, XML, JSON

**observation 96** (rq1-24:12-9 XML download from wetBig seems like a logical step for the analysis and processing, but it is too complex. This also applies to the JSON structure. Both structures are not pleasant to read. The thought that comes to mind here is why SDU doesn't directly annotate the concepts and relationships.).

**obs:** *Both structures have been downloaded to add the annotations to them. To later use a program to extract the annotation (xml/json annotations) together with the definitions and meaning and to generate a script or part of a script with this.*

**observation 97** (rq1-31:14-9: Besides XML and JSON, RTF and PDF are also an option. In rtf (doc) you can add items in the margins via "comments". With a PDF, annotations and color highlighting can be given this feature.). **obs:** *To get the overview and to keep track of what has already been processed.*

## Php

**observation 98** (rq1-9 Adding pieces of php code in the script is possible, but it is not clear how). **obs:** *Information is missing on how to do this.*

**observation 99** (rq1-57-2:7-11: Parts like next big number or now() and today() are better solved in a dev language, like php.). **obs:** *A development language like php because it's in the Ampersand software stack.*

**observation 100** (rq2-9:7-10: Subscription time is added automatically. This is done by means of a rule.). **obs:** *Despite not being able to add php functions, it appears to be possible to add a date-time automatically. That failed in previous attempts. This only worked with support.*

**observation 101** (rq2-16:19-10/11-11: Ampersand has a hard time determining a period. Ampersand cannot calculate out of the box. This requires the php functions, which are also not easy to allocate.). **obs:** *Ampersand cannot calculate out-of-the-box. Is this actually a problem or do you have to solve these types of elements at Ampersand.*

## Prototype

**observation 102** (rq1-36:29-9: Failed to run prototype under localhost in Windows-10. The service would not start in localhost. We did manage to get the service running within Docker. There was an error in the installation documentation. Turns out that is was not the installation directory RapInstall, but the directory RAP.). **obs:** *Unable to run the service in localhost, but within Docker.*

**observation 103** (rq1-69:14-11: Postman application installed and works with the prototype.). **obs:** *An external resource (Postman<sup>28</sup>) that can perform tests using api.*

---

<sup>28</sup><https://www.postman.com/product/what-is-postman/>

**observation 104** (rq1-36:3-10: What about prototype test scenarios.). **obs:** *Apparently lacking testing tools. Generic tools such as Selenium may need to be used.*

## Register

**observation 105** (rq1-85:30-11: A new structure where the registers can operate independently of each other, with only the generic elements as common items.). **obs:** *This does not work due to multicontext issue.*

**observation 106** (rq1-92:14-12: Basically trying to create its own container per register. Multi-context problem. This makes it impossible to isolate these containers.). **obs:** *It is not possible in the current setup of Ampersand to create your own container per registry.*

**observation 107** (rq1-93:19-12: Implementation choice for separate registers has an impact on the whole. How to deal with shared modules. How to deal with shared data (such as person). Should the choice be made to only share the concepts and relationships and not implementation.). **obs:** *solution could be to provide each register with its own db and a shared db for eg people Port usage is therefore an issue. Can something be arranged in the .env. Elaboration of the own containers does not seem to work, the db structure is always overwritten by the new registry.*

**observation 108** (rq3-9:19-9: The structure of the register's is the same, registers are also called registrations). **obs:** *Commonality emerges here.*

**observation 109** (rq3-8:19-9: The law states that there are multiple registers. There is a register per profession. The scripts may also need to be formatted that way.). **obs:** *Multiple registers are mentioned in the text of the law. The current implementation of Zorro shows that only one register has been implemented, with different workflows for handling the professions (the actual registers).*

## Relation

**observation 110** (rq1-7:10-11: Each relation is part of a record structure.). **obs:** *Good to discover how the database structure is established. It is probably stated somewhere how this happens. But this can be determined through reversed engineering. Above all, it provides insight and makes it more tangible.*

**observation 111** (rq2-11:19-10: An relation that is univalent is a function. A one function there can only come out one thing. The description of UNI is therefore P ->0-1 H at most (see 2-5)). **obs:** *An relation that is univalent is a function.*

**observation 112** (rq2-10:19-10: The naming of a relation is usually assigned to the TRG attribute of the set. Such as [Persoon \* Voornaam] with relation name "voornaam"). **obs:** *Making agreements about processing is important. When there are agreements, things can also be found again.*

## Represent

**observation 113** (rq1-50:30-10: The represent statement makes the interface react differently. When using the represent statement, the append option ("+" disappears.). **obs:** *Unexpected behavior, it is not immediately clear why this is happening.*

**observation 114** (rq1-65:10-11: DATETIME (represent) field could not be converted to Excel. The compilation process hangs on this.). **obs:** *Crashing while building the application using DATETIME in the represent statement.*

**observation 115** (rq2-8:7-10/10-10: Date of birth must be formatted as date. The represent seems to have to fulfill that role. Represent defines a type of a concept, but DATETIME causes interface problems.). **obs:** *Type of elements can be sent, subject to conditions.*

## Rule

**observation 116** (rq1-4 Automatically executed rule are easy to describe, but implementation here also takes a lot of patience and trying.). **obs:** *Rules are not easy to create. To implement rules, knowledge of Ampersand is required and many examples must be used. It is usually not possible to immediately implement a rule. Many attempts are needed to realize this.*

**observation 117** (rq1-55:2-11: At the rule it is necessary to add a ROLE with a MAINTAINS, otherwise the rule will not work.). **obs:** *In the beginning this is not obvious. This becomes clear when studying examples.*

**observation 118** (rq1-59:9-11: Many messages remain open if not all rules are met.). **obs:** *When the input is handled easily, more and more messages appear. The messages are grouped by type. The workable screen is getting smaller and smaller.*

**observation 119** (rq1-17 Applying a rules takes a lot of patience and practice. This is quite a steep learning curve.). **obs:** *Implementing a rule requires knowledge of relation algebra and a lot of trying and looking at examples.*

**observation 120** (rq1-39:3-10: Do not forget to create delete rules in addition to append and edit rules in the rules in the context of the Lifecycle approach.). **obs:** *Completeness of functions on the relations.*

**observation 121** (rq1-67:11-11: If there is an automatic rule, should there still be a validation rule on it?). **obs:** *Yes, if there is a chance that the automatic rule is (accidentally) removed or changed. Use as an intrinsic control agent.*

**observation 122** (rq2-6:2-10/13-11: A rule is not easy to realize. There are tricks to realize this. rq1-17 Applying a rule requires a lot of patience and practice.). **obs:** *Rules require knowledge of Ampersand, but also many examples and they are not very available.*

**observation 123** (rq2-14:19-10: The role gives control to the user. A user is authorized for use. It indicates which user is allowed to use the function.). **obs:** *In the beginning I left this element out of consideration, assuming there was some kind of authorization. But this is necessary to get the rule working.*

**observation 124** (rq1-61:9-11: There should be a check on the draft date of birth(rule), so that someone must be at least 18. Sounds logical, but is a derived rule. This is already implicit in the training requirement. The duration of the training means that someone is at least 18 years old before the training is completed.). **obs:** *You don't have to think of anything yourself.*

## Specialism

**observation 125** (rq3-13:17-10: There is no list of specialties in **Wet-BIG**, where is it?). **obs:** *The law does refer to specialism.*

## Validation

**observation 126** (rq1-57-1:7-11: Using Ampersand for validation.). **obs:** *Building an Ampersand script delivers a core on all defined validations and can be used immediately.*

## Visual Studio Code

**observation 127** (rq1-22 The tool VSC also doesn't have a generic search option across the ads.). **obs:** *Not being able to search globally is inconvenient when looking for usage of concepts and relationships or when refactoring them. To promote reuse, findability is necessary. Now tools outside of **VSC** must be used, within the OS being used, to search within files.*

**observation 128** (rq1-32:14-9: The tool VSC has an Ampersand extension. It hangs once in a while.). **obs:** *Must be my system, but it's annoying. My PC often hangs on this. rq1-87:3-12: Latex can also be written in VSC. Apparently it is a different version, because the import does not immediately succeed. Does not work really well and the result is poor. **obs:** **Visual Studio Code** also supports the TEX environment through add-ons. But this add-on completely hangs my system. I got a 100% cpu load for a long time.*

## B. Interviews

### Lawyer

**int 1** (I-3.1, For a lawyer, the IT environment is an unfamiliar environment. The lawyer actually wants to form a picture of the system. In particular, what it looks like and what it can do. This while we want to involve the lawyer at the beginning of the process. Especially when we don't have the system yet.).

**int 2** (I-3.2, A draft of the conceptual analysis is available and this is experienced as trusted by the lawyer. In fact, these are recognizable texts because they have been taken directly from the law.).

**int 3** (I-3.3, The **Wet-BIG** offers a lot of room for interpretation. This interpretation possibility means that the law may lend itself less to an Ampersand translation than a recent law would. The new laws have therefore been drafted more carefully. The law provides a framework and the question is how far one should go with recording. This law gives the freedom to fill in matters yourself.).

**int 4** (I-3.4, The aim should not be to record everything that is stated in the law in an ICT system. That makes it very rigid. Make sure that 80% of the situations are supported and leave the rest to the employees. Ampersand is very suitable for this, precisely because it has a reactive approach and therefore does not prescribe how the practitioners should act.).

**int 5** (I-3.5, The aim of an ICT system should be to do as little manual work as possible. And when a new law is being developed, an ICT representative should be present.).

**int 6** (I-3.6, Because the law was drafted some time ago, the definitions are not always unambiguous. And because of the aforementioned interpretation possibility, the legislator can interpret the law slightly differently through jurisprudence.).

**int 7** (I-3.7, The law consists of the following parts. Going through the law should be a first step for the conceptual analysis.).

- Artikel 1 definities
- Artikel 3 welke beroepen ed
- Artikel 4 uitbreiding Artikel 3
- Artikel 5 grondslagen voor regelgeving
- Artikel 6 weigergronden
- Artikel 7 doorhalen - 7a hardheidsclausules
- Artikel 8 basis voor herregistratie, technische artikelen

- Artikel 9 tuchtgebeuren; wat we aantekenen op het register (inschrijving = registratie) maatregelen, doorgehaalde reden bepaald of je zichtbaar bent
- Artikel 10 beschikking
- Artikel 11 aanmelden beschikking (staatscourant)
- Artikel 12 openbaarmaking big-registratie - staat wat er gemeld mag worden
- Artikel 13 privacy + delen van info; grondslagen per doelgroep
- Artikel 14 beroepsverenigingen - wordt aangetekend in big-register
- Artikel 15.16.17 specialisten registers
- H3- Artikel 18 eisen per beroep oa opleiding tm Artikel 33
- Artikel 34 geen beroepstitel, maar wel behandeld als a3-beroep; ook opleiding is bepalend. Geen eigen register
- Artikel 35 voorbehouden handelingen
- Artikel 36a+b tijdelijke registers bv mondhygenistes
- h5 tav buitenlandse gediplomeerden; erkenning process (EU, overige buitenland)
- Artikel 45 als Artikel 34
- h6/7 tucht
- etc

**Product Owner** TOGO

### **Developer**

**int 8** (I-2.1,CIBG's architecture for new registers consists largely of **Registerkern**. This was introduced not so long ago and is still being expanded.).

**int 9** (I-2.2,Ampersand has APIs and that is interesting to be able to link with. Whether that can also be linked with **Registerkern** is not clear at the moment.).

**int 10** (I-2.3,Nice that Ampersand is an open source product. There is not much to be found. Only the github repository can be found.).

**int 11** (I-2.4,Does Ampersand support databases other than just MariaDB? Not at the moment, but it is to be expected that this will be possible.).

**int 12** (I-2.5,When maintenance takes place on the model, how do we get from one model to another. So how does the IST go to SOLL situation. Ampersand is always creating a new model. So when the law is changed and a new model is needed as a result, Ampersand will produce a completely new model. As a result, no technical debt will remain in the model. It is always a new model. However, the challenge will be in the data migration from the old to the new model.).

**int 13** (I-2.6,Registerkern its terminology includes things and products. Every service, read implementation of a law, we call a product. There are standard parts that always appear in every register. These are pre-modeled within Registerkern. This includes a base for each registry and can be expanded according to the needs of the registry. The basis is the minimum common denominator of the registers. Extendable to specific elements arising from the law. There is certainly overlap in the data obtained from the analysis of the big law and the Registerkern. About 80% of the Registerkern is generic and the other 20% is customised. So all new registers have the same basic principles and for the most part run on the same software.).

**int 14** (I-2.7,Another aspect of the terminology is that items with the same definition are named differently within the law and within the Registerkern. In Registerkern we are talking about business and products, while the law is big about registrations, applications and professional registers. A mapping of the terms used will have to take place.).

**int 15** (I-2.8,Due to the overlap between Registerkern and the Conceptual analysis of Big, it is difficult to find the demarcation line between the two systems. Ampersand is state oriented and the Registerkern is process oriented. The link and cooperation must be sought.).

**int 16** (I-2.9,The usual procedure within a register is the application process for a registration. The Registerkern has a wizard for this, which includes a diploma check, for example. This diploma check is also part of the current implementation of the Wet-BIG.).

**int 17** (I-2.10,Ampersand's approach is in line with the Registerkern, but not at the implementation level. It doesn't seem possible to implement Ampersand directly, but the analysis seems quite useful for extending Registerkern. Where generality is discovered and for the specific parts of the law. Then we are talking about a conceptual link and not a technical one.).

**int 18** (I-2.11,An addition of Ampersand is that a prototype is made that can also be tested. This allows the entire system to be tested because this combination must comply with validation from the law.).

## Architecture

**int 19** (I-1.1,The assumption made by the interviewee that Ampersand is a tool that performs an interpretation on the law itself, is not correct. A manual stroke has to be done over the text of the law to recognize the concepts and relationships. This is seen as an intensive action.).

**int 20** (I-1.2,Ampersand can be interesting, because it will be able to clear conflicting matters from the law. By performing the analysis, these will show up in the analysis. This makes it a resource to use before the law is enacted.).

**int 21** (I-1.3,The **Registerkern**, an architecture model of CIBG, uses shared concepts. With this it has similarities with Ampersand. There is also an overlap of Ampersand with **Registerkern**. Within Ampersand are concepts that are also in **Registerkern**. **Registerkern** is a defining part of the architecture. Other parts will have to conform to this architecture.).

**int 22** (I-1.4,The danger of using legislation and regulations is that there is a possible incomplete picture of the concepts. This by adopting the rules one-on-one, without the interpretations. More laws are also used in an analysis than just the **Wet-BIG**. The question is how far is the analysis of the various laws going.).

**int 23** (I-1.5,To be able to use Ampersand it would be useful to avoid having to write code in C#.).

**int 24** (I-1.6,In addition, the implementation must be such that the effective dates of the specific amendments to the law are also taken into account. For example, at the time of an application, it is decisive whether the processing will take place in accordance with the old situation or the new situation.).

**int 25** (I-1.7,Ampersand does not support a maintenance cycle. There must be a solution for this.).

**int 26** (I-1.8,Ampersand's possible positioning is to use it as an interpreter of legislation and regulations. Then maintain the current analysis and development process and use the prototype to validate the analysis. The question is whether this approach will not result in additional work compared to the current working method. There is a certain skepticism towards Ampersand.).

**int 27** (I-1.9,Ampersand relies on facts and not on processes. While a practitioner is strongly process oriented. For example, the law does indicate that a diploma is required and also which type, but not exactly which diploma. So the law tells you what to do, but in most cases not how.).

**int 28** (I-1.10,In addition, the practitioner's usual working method is that he works from overviews and lists. Ampersand will have to be designed for this with user requirements, because these things are not mentioned in the law. The law does not support a method and approach. This will have to be a so-called co-creation between IT and business.).



**int 29** (I-1.11,The question is whether the wet-big is very suitable for this approach. The original law dates from 1993 and it is based on the legislation of 1865.).

**int 30** (I-1.12,The terms case, submission and application are strongly represented in the handling of the registers. These terms do not appear in the Ampersand analysis. The term “case” is not mentioned at all in the law-big. Because the process part is missing, this is considered a weakness of Ampersand. It is clear that Ampersand state is oriented and reactive and not process oriented.).

## **Analyst**

**int 31** (I-4.1,The **Wet-BIG** is big and also old. Ampersand could help detect inconsistencies in the law.).

**int 32** (I-4.2,The prototype shown is not easy for the user to understand. The user not only looks at the functionality, but also at the design. The current design does not comply with the national government web guidelines. The question is whether the user will be able to see through this. It was not part of the research, but it was stated that adjusting the CSS could bring closer to the web guidelines.).

**int 33** (I-4.3,Ampersand’s deployment could be applied to new tasks. These have no history and can be built from scratch using the Ampersand method.).

**int 34** (I-4.4,A use case can also be devised for the use of rebuilding existing systems. Through the analysis with the help of Ampersand, a system can be rebuilt in which the waste has been cut away. The question is how much this waste would be. Worth a try.).

**int 35** (I-4.5,For use, the question is how quickly a base is set up. It may be difficult to get to a 100% model. It may also be okay if this covers an 80% charge. LCSH as new project could be a good candidate.).

**int 36** (I-4.6,Ampersand method is a way of writing things down. That is not necessarily better or worse than any other method. So when something is being written down, so analysis is being done, why not with this. More is possible with it than with a Word document. The output is good to use and the structure too.).

**int 37** (I-4.7,In the current trend, validations are usually located in the business layer. Is that also the case with Ampersand? The validations are spread over the database and surrounding code.).

**int 38** (I-4.8,How is the maintenance of the system? A new model is always made with the help of Ampersand. The data will have to be migrated itself. Ampersand does not support that. Usually the data structure is taken into account in advance so that as little conversion as possible has to take place. This means that a system is getting bigger and less manageable. So the strength of Ampersand is that this is prevented because a new core system is always being built and the effort is in the data conversion and the connection of adjacent systems.).

**int 39** (I-4.9,The learning curve doesn't seem that big. Even less technical people can work with this. With the adjustment in the styling, a prototype can be quickly made with which a working system can be demonstrated. On the other hand, only the conceptual analysis can be used. Based on this analysis, test scenarios can be devised and executed.).

**int 40** (I-4.10,The Ampersand approach is different from most products. Most workbenches work from a drawn model and from there generate code from the documentation or possibly. Ampersand does this from a script and generates the models and documentation itself.).

**int 41** (I-4.11,The question is whether the system will only work for simple registers or whether we can also use it to tackle complex registers. The **Wet-BIG** is complex, but not fully analyzed either.).

**int 42** (I-4.12,A follow-up study could be to make a comparison between a system built traditionally and a system built on the Ampersand method. It is expected that due to code generation and being closer to the law, the amount of code will be a lot less. And with that also a better SIG qualification.).

**int 43** (I-4.13,To start with, a team should be set up to deal with this. This team of lawyers and analysts should be doing the analysis of a law and have it built.).

**int 44** (I-4.14,One could also only use the output of the analysis to build a system. Multiple scenarios are possible.).

## C. Associated laws and regulations

- Wet op de beroepen in de individuele gezondheidszorg  
<https://wetten.overheid.nl/BWBR0006251/2021-07-01>
- Algemene wet bestuursrecht  
<https://wetten.overheid.nl/BWBR0005537/2022-03-02>
- Besluit periodieke registratie Wet BIG  
<https://wetten.overheid.nl/BWBR0024841/2020-07-01>
- Registratiebesluit BIG  
<https://wetten.overheid.nl/BWBR0007648/2021-01-01>
- Tuchtrectbesluit BIG  
<https://wetten.overheid.nl/BWBR0008688/2021-04-01>
- Besluit gezondheidszorgpsycholoog  
<https://wetten.overheid.nl/BWBR0009467/2016-10-06>
- Regeling periodieke registratie Wet BIG  
<https://wetten.overheid.nl/BWBR0025605/2020-12-15>
- Regeling tarieven registratie beroepsbeoefenaren Wet BIG  
<https://wetten.overheid.nl/BWBR0031720/2014-02-01>
- Algemene wet erkenning EU-beroepskwalificaties  
<https://wetten.overheid.nl/BWBR0023066/2021-08-26>
- Besluit buitenslands gediplomeerden volksgezondheid  
<https://wetten.overheid.nl/BWBR0007397/2020-10-01>
- Regeling erkenning EU-beroepskwalificaties beroepen in de individuele gezondheidszorg  
<https://wetten.overheid.nl/BWBR0024755/2018-09-08>

## D. Content Analysis

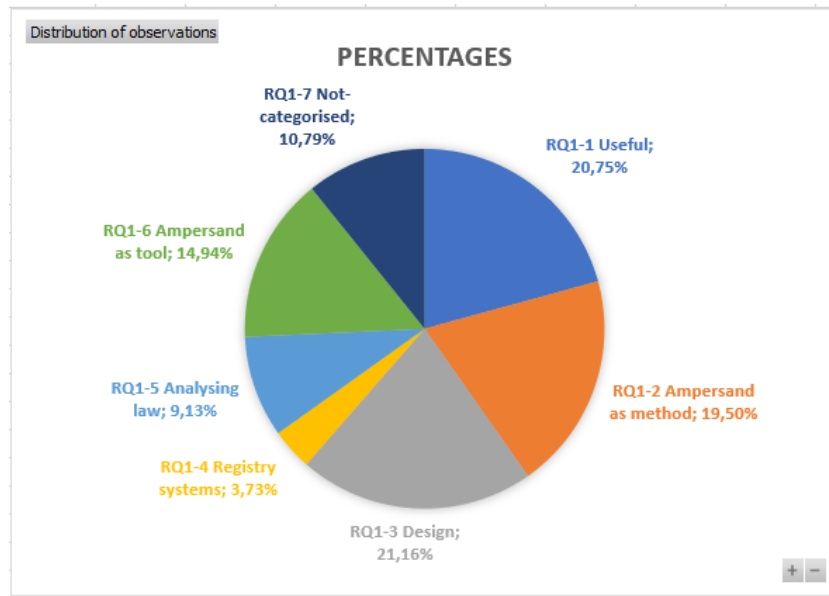


Figure 19: Content observations Distribution

Table 9: List of observations **RQ1-1, Category:Useful, Reference to observation/interview**

<b>RQ1-1, Category:Useful, Reference to observation/interview</b>
int.35: I-4.5,For use, the question is how quickly a base is set up. It may be difficult to get to a 100% model. It may also be okay if this covers an 80% charge. LCSH as new project could be a good candidate
obs.47: rq1-11 Implementation in Docker with RAP creates new directories all the time
obs.2: rq1-18 Can not find an example on the internet, only in the repo of Ampersand itself. That is difficult to find
obs.1: rq1-13:17-10: The setup of Ampersand in local environment is specific and not self-explanatory. Help is needed here to get this working. Attempts to get the process working in localhost were unsuccessful. The manual on the Ampersand site showed how to do this. But it still didn't work
obs.102: rq1-36:29-9: Failed to run prototype under localhost in Windows10. The service would not start in localhost. We did manage to get the service running within Docker. There was an error in the installation documentation. Turns out that is was not the installation directory RapInstall, but the directory RAP
obs.5: rq1-47:27-10: Detecting a bug. Placing these in github issues at the Ampersand repository will get a response within a day and resolve it. In this case it was a bug in Ampersand that was quickly fixed with a new version
obs.48: rq1-6:21-10: Docker is also another thing to learn. There should also be an introductory course to quickly understand Docker usage for Ampersand. A waste of time to have to look this up yourself or it is preconditions to be able to use Ampersand
obs.11: rq4-8:22-11: The team behind Ampersand is very dedicated
obs.119: rq1-17 Applying a rules takes a lot of patience and practice. This is quite a steep learning curve
Continued on next page

**Table 9 – continued from previous page**

**RQ1-1, Category:Useful, Reference to observation/interview**

- obs.6: rq1-60:9-11: Training and education is required to write an Ampersand script
- obs.7: rq1-96:30-12: Skill in scripting within Ampersand is quickly lost if you don't do this frequently
- int.7: I-3.7,The law consists of the following parts. Going through the law should be a first step for the conceptual analysis
- obs.4: rq1-2 Ampersand has no annotation option, therefore requires a separate action or document to keep track of what has been passed
- obs.85: rq1-25:12-9: First make overview of all laws and regulations
- obs.97: rq1-31:14-9: Besides XML and JSON, RTF and PDF are also an option. In rtf (doc) you can add items in the margins via "comments". With a PDF, annotations and color highlighting can be given this feature
- obs.25: rq1-42:19-10: Immediately add the description when recording a concept and relation. Later it is difficult to find out why the recording took place
- obs.3: rq1-45:24-10: Overview within an Ampersand script is difficult to obtain
- obs.27: rq1-46:24-10: There is no find able relationship between the relation and the concept in the script
- obs.52: rq1-97:30-12: By puzzling with Ampersand people quickly forget to make correct documentation. Often you are happy that something works
- int.2: I-3.2,A draft of the conceptual analysis is available and this is experienced as trusted by the lawyer. In fact, these are recognizable texts because they have been taken directly from the law
- obs.54: rq3-7 Adding documentation with the correct description to a concept and relation is not so easy. Easy to stray and add your own interpretation
- obs.76: rq1-42:21-10: It is easy to deviate from the legal texts. Because they are so hard to read. Some knowledge of the law or the process means that your own interpretation is quickly made. Action research also means that you quickly fall into this trap
- int.21: I-1.3,The Registerkern, an architecture model of CIBG, uses shared concepts. With this it has similarities with Ampersand. There is also an overlap of Ampersand with Registerkern. Within Ampersand are concepts that are also in Registerkern. Registerkern is a defining part of the architecture. Other parts will have to conform to this architecture
- int.26: I-1.8,Ampersand's possible positioning is to use it as an interpreter of legislation and regulations. Then maintain the current analysis and development process and use the prototype to validate the analysis. The question is whether this approach will not result in additional work compared to the current working method. There is a certain skepticism towards Ampersand
- int.17: I-2.10,Ampersand's approach is in line with the Registerkern, but not at the implementation level. It doesn't seem possible to implement Ampersand directly, but the analysis seems quite useful for extending Registerkern. Where generality is discovered and for the specific parts of the law. Then we are talking about a conceptual link and not a technical one
- obs.19: rq1-62:10-11: There has be the architecture link between the law core and the register core
- int.9: I-2.2,Ampersand has APIs and that is interesting to be able to link with. Whether that can also be linked with Registerkern is not clear at the moment
- obs.17: rq4-2 The api link works fine, but entire messages return. These should actually get codes
- obs.18: rq4-5 Postman used for api link with Ampersand

Continued on next page

**Table 9 – continued from previous page**

**RQ1-1, Category:Useful, Reference to observation/interview**

obs.13: rq1-70:14-11: Postman works with api/v1/resource, e.g. GET localhost/api/v1/resource/Person/P001/Person, retrieves that of an existing person. So the validation structure of ampersand can be used from outside Ampersand by means of api

obs.14: rq1-72:14-11: Besides the GET(get), the POST(append) and PUT (mutate) also work

obs.15: rq1-73:14-11: Ampersand can be used from other applications through APIs, but the return values are next to the requested information also messages and not message codes. These codes could be included in the reports, but now remain "unstructured" data

obs.16: rq1-74:16-11: Link between an external front-end and an Ampersand back-end (Ampersandapi). A change in the back-end, so an Ampersand change, then the front end almost certainly has to change with it

obs.12: rq1-8:14-11: No swagger is created for the api;

int.12: I-2.5,When maintenance takes place on the model, how do we get from one model to another. So how does the IST go to SOLL situation. Ampersand is always creating a new model. So when the law is changed and a new model is needed as a result, Ampersand will produce a completely new model. As a result, no technical debt will remain in the model. It is always a new model. However, the challenge will be in the data migration from the old to the new model

int.38: I-4.8,How is the maintenance of the system? A new model is always made with the help of Ampersand. The data will have to be migrated itself. Ampersand does not support that. Usually the data structure is taken into account in advance so that as little conversion as possible has to take place. This means that a system is getting bigger and less manageable. So the strength of Ampersand is that this is prevented because a new core system is always being built and the effort is in the data conversion and the connection of adjacent systems

obs.10: rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)

obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses

int.36: I-4.6,Ampersand method is a way of writing things down. That is not necessarily better or worse than any other method. So when something is being written down, so analysis is being done, why not with this. More is possible with it than with a Word document. The output is good to use and the structure too

obs.46: rq2-18:16-11: Good to realize that the meaning you write down also ends up in the Conceptual analysis. So looking at the way of writing it down can form a story in the analysis

int.20: I-1.2,Ampersand can be interesting, because it will be able to clear conflicting matters from the law. By performing the analysis, these will show up in the analysis. This makes it a resource to use before the law is enacted

int.24: I-1.6,In addition, the implementation must be such that the effective dates of the specific amendments to the law are also taken into account. For example, at the time of an application, it is decisive whether the processing will take place in accordance with the old situation or the new situation

int.41: I-4.11,The question is whether the system will only work for simple registers or whether we can also use it to tackle complex registers. The Wet-BIG is complex, but not fully analyzed either

Continued on next page

**Table 9 – continued from previous page**

**RQ1-1, Category:Useful, Reference to observation/interview**

- int.42: I-4.12,A follow-up study could be to make a comparison between a system built traditionally and a system built on the Ampersand method. It is expected that due to code generation and being closer to the law, the amount of code will be a lot less. And with that also a better SIG qualification
- int.33: I-4.3,Ampersand’s deployment could be applied to new tasks. These have no history and can be built from scratch using the Ampersand method
- int.39: I-4.9,The learning curve doesn’t seem that big. Even less technical people can work with this. With the adjustment in the styling, a prototype can be quickly made with which a working system can be demonstrated. On the other hand, only the conceptual analysis can be used. Based on this analysis, test scenarios can be devised and executed
- obs.120: rq1-39:3-10: Do not forget to create delete rules in addition to append and edit rules in the rules in the context of the Lifecycle approach
- obs.110: rq1-7:10-11: Each relation is part of a record structure
- obs.101: rq2-16:19-10/11-11: Ampersand has a hard time determining a period. Ampersand cannot calculate out of the box. This requires the php functions, which are also not easy to allocate
- obs.81: rq3-16:24-10: For the Netherlands, we have a country table from the RvIG. These are nationally established and maintained tables. No maintenance function is therefore required

Table 10: List of observations **RQ1-2, Category:Ampersand as method, Reference to observation/interview**

**RQ1-2, Category:Ampersand as method, Reference to observation/interview**

- obs.24: rq1-30:12-9: Defining the meaning and definition of the concept is free of rules. There is no fixed pattern for documentation
- obs.93: rq1-33:14-9: The use of patterns within Ampersand is important. These are the subsystems of the information system. The question is whether this should be classified in advance or whether it builds up on its own
- obs.95: rq1-38:3-10: Should the subsystems be mapped in advance
- obs.27: rq1-46:24-10: There is no find able relationship between the relation and the concept in the script
- obs.28: rq1-80:20-11: A consistent naming of a concept is necessary
- obs.45: rq2-4:30-9: Which agreements must be made regarding the structure of the descriptions for Conceptual analysis. Do agreements have to be made about it or leave it unstructured?
- obs.88: rq1-3 Created a separate excel to write out and discover the multiplicity of the relations
- obs.89: rq2-2 Only UNI, TOT, INJ and SUR are used
- obs.70: rq1-21:7-11: TOT is usually overcome by a tot-rule, it turns out that a TOT causes something to be saved when entered, while a tot-rule allows a save to occur while the notification remains open to stand
- obs.63: rq1-58:8-11: Per interface max one multiplicity, otherwise you won’t get data stored
- obs.111: rq2-11:19-10: An relation that is univalent is a function. A one function there can only come out one thing. The description of UNI is therefore P ->0-1 H at most (see 2-5)
- obs.69: rq2-12:19-10: TOT has the property that this must be entered in the interface because otherwise the data will not be saved. A variant of this is an rule with this property As a result, the other items are stored in the database, but a notification of incompleteness continues to appear

Continued on next page

**Table 10 – continued from previous page**

**RQ1-2, Category:Ampersand as method, Reference to observation/interview**

- obs.90: rq2-13:19-10: What applies to multiplicity TOT, also applies to SUR
- obs.87: rq2-5:2-10: Making the multiplicity explicit
- obs.116: rq1-4 Automatically executed rule are easy to describe, but implementation here also takes a lot of patience and trying
- obs.124: rq1-61:9-11: There should be a check on the draft date of birth(rule), so that someone must be at least 18. Sounds logical, but is a derived rule. This is already implicit in the training requirement. The duration of the training means that someone is at least 18 years old before the training is completed
- obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses
- obs.121: rq1-67:11-11: If there is an automatic rule, should there still be a validation rule on it?
- obs.8: rq2-19:16-11: Ampersand returns constraints and no executable
- obs.122: rq2-6:2-10/13-11: A rule is not easy to realize. There are tricks to realize this. rq1-17 Applying a rule requires a lot of patience and practice
- obs.23: rq1-79:20-11: Once a concept for a date or other element is defined, it can be used anywhere in the context. The question then is how to deal with shared Concepts and how to manage them
- obs.31: rq1-91:14-12: A concept and a relation can be defined several times within your own patterns. So that the patterns can stand on their own
- int.22: I-1.4,The danger of using legislation and regulations is that there is a possible incomplete picture of the concepts. This by adopting the rules one-on-one, without the interpretations. More laws are also used in an analysis than just the Wet-BIG. The question is how far is the analysis of the various laws going
- int.7: I-3.7,The law consists of the following parts. Going through the law should be a first step for the conceptual analysis
- int.43: I-4.13,To start with, a team should be set up to deal with this. This team of lawyers and analysts should be doing the analysis of a law and have it built
- obs.25: rq1-42:19-10: Immediately add the description when recording a concept and relation. Later it is difficult to find out why the recording took place
- obs.44: rq1-51:2-11: Discussing the Conceptual analysis should be done theme by theme
- obs.29: rq1-84:30-11: A concept is immutable. for example a person is concept, not doctor. It must be an intrinsic property, which cannot be changed
- int.26: I-1.8,Ampersand's possible positioning is to use it as an interpreter of legislation and regulations. Then maintain the current analysis and development process and use the prototype to validate the analysis. The question is whether this approach will not result in additional work compared to the current working method. There is a certain skepticism towards Ampersand
- int.18: I-2.11,An addition of Ampersand is that a prototype is made that can also be tested. This allows the entire system to be tested because this combination must comply with validation from the law
- int.1: I-3.1,For a lawyer, the IT environment is an unfamiliar environment. The lawyer actually wants to form a picture of the system. In particular, what it looks like and what it can do. This while we want to involve the lawyer at the beginning of the process. Especially when we don't have the system yet
- int.2: I-3.2,A draft of the conceptual analysis is available and this is experienced as trusted by the lawyer. In fact, these are recognizable texts because they have been taken directly from the law

Continued on next page



**Table 10 – continued from previous page**

**RQ1-2, Category:Ampersand as method, Reference to observation/interview**

int.32: I-4.2,The prototype shown is not easy for the user to understand. The user not only looks at the functionality, but also at the design. The current design does not comply with the national government web guidelines. The question is whether the user will be able to see through this. It was not part of the research, but it was stated that adjusting the CSS could bring closer to the web guidelines

obs.126: rq1-57-1:7-11: Using Ampersand for validation

int.19: I-1.1,The assumption made by the interviewee that Ampersand is a tool that performs an interpretation on the law itself, is not correct. A manual stroke has to be done over the text of the law to recognize the concepts and relationships. This is seen as an intensive action

int.27: I-1.9,Ampersand relies on facts and not on processes. While a practitioner is strongly process oriented. For example, the law does indicate that a diploma is required and also which type, but not exactly which diploma. So the law tells you what to do, but in most cases not how

int.40: I-4.10,The Ampersand approach is different from most products. Most workbenches work from a drawn model and from there generate code from the documentation or possibly. Ampersand does this from a script and generates the models and documentation itself

int.44: I-4.14,One could also only use the output of the analysis to build a system. Multiple scenarios are possible

int.35: I-4.5,For use, the question is how quickly a base is set up. It may be difficult to get to a 100% model. It may also be okay if this covers an 80% charge. LCSH as new project could be a good candidate

int.39: I-4.9,The learning curve doesn't seem that big. Even less technical people can work with this. With the adjustment in the styling, a prototype can be quickly made with which a working system can be demonstrated. On the other hand, only the conceptual analysis can be used. Based on this analysis, test scenarios can be devised and executed

obs.117: rq1-55:2-11: At the rule it is necessary to add a ROLE with a MAINTAINS, otherwise the rule will not work

obs.110: rq1-7:10-11: Each relation is part of a record structure

obs.112: rq2-10:19-10: The naming of a relation is usually assigned to the TRG attribute of the set. Such as [*Persoon \* Voornaam*] with relation name "voornaam"

obs.123: rq2-14:19-10: The role gives control to the user. A user is authorized for use. It indicates which user is allowed to use the function

obs.115: rq2-8:7-10/10-10: Date of birth must be formatted as date. The represent seems to have to fulfill that role. Represent defines a type of a concept, but DATETIME causes interface problems

Table 11: List of observations **RQ1-3, Category:Design, Reference to observation/interview**

**RQ1-3, Category:Design, Reference to observation/interview**

int.30: I-1.12,The terms case, submission and application are strongly represented in the handling of the registers. These terms do not appear in the Ampersand analysis. The term "case" is not mentioned at all in the law-big. Because the process part is missing, this is considered a weakness of Ampersand. It is clear that Ampersand state is oriented and reactive and not process oriented

int.21: I-1.3,The Registerkern, an architecture model of CIBG, uses shared concepts. With this it has similarities with Ampersand. There is also an overlap of Ampersand with Registerkern. Within Ampersand are concepts that are also in Registerkern. Registerkern is a defining part of the architecture. Other parts will have to conform to this architecture

Continued on next page

**Table 11 – continued from previous page**

**RQ1-3, Category:Design, Reference to observation/interview**

int.8: I-2.1,CIBG's architecture for new registers consists largely of Registerkern. This was introduced not so long ago and is still being expanded

int.13: I-2.6,Registerkern its terminology includes things and products. Every service, read implementation of a law, we call a product. There are standard parts that always appear in every register. These are pre-modeled within Registerkern. This includes a base for each registry and can be expanded according to the needs of the registry. The basis is the minimum common denominator of the registers. Extendable to specific elements arising from the law. There is certainly overlap in the data obtained from the analysis of the big law and the Registerkern. About 80% of the Registerkern is generic and the other 20% is customised. So all new registers have the same basic principles and for the most part run on the same software

int.14: I-2.7,Another aspect of the terminology is that items with the same definition are named differently within the law and within the Registerkern. In Registerkern we are talking about business and products, while the law is big about registrations, applications and professional registers. A mapping of the terms used will have to take place

int.15: I-2.8,Due to the overlap between Registerkern and the Conceptual analysis of Big, it is difficult to find the demarcation line between the two systems. Ampersand is state oriented and the Registerkern is process oriented. The link and cooperation must be sought

obs.9: rq4-1 Ampersand cannot calculate. But since Ampersand is static, process data can be monitored in other ways

obs.20: rq4-3 Embedding in architecture, the core of the law with shared concepts and processes. The core of law is specific law. Shared concepts are also part of the law but also occur elsewhere. This is part of embedding in architecture

obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses

obs.30: rq1-89:7-12: Items named as common concepts

obs.38: rq1-1 Formatting in Ampersand (patterns) has consequences for the Conceptual analysis

obs.53: rq1-75:20-11: Some more experimentation with the documentation in the prototype. When describing the purpose of the context, it takes a while to figure out how this text can be properly conveyed. An <h1> results in an extra chapter in H4 and H4 then becomes H5 and H5 has then become a meaningless piece. With an <h2> and <h3> it works well

obs.49: rq1-78:20-11: The documentation generated in HTML loaded in firefox and no PNG's are visible. Chrome is doing well

obs.42: rq1-99:6-1: when generating a Conceptual analysis the doc gets the name of the first concept

obs.46: rq2-18:16-11: Good to realize that the meaning you write down also ends up in the Conceptual analysis. So looking at the way of writing it down can form a story in the analysis

obs.120: rq1-39:3-10: Do not forget to create delete rules in addition to append and edit rules in the rules in the context of the Lifecycle approach

obs.34: rq1-95:29-12: The format of a concept big number is not included in the law

obs.106: rq1-92:14-12: Basically trying to create its own container per register. Multi-context problem. This makes it impossible to isolate these containers

obs.107: rq1-93:19-12: Implementation choice for separate registers has an impact on the whole. How to deal with shared modules. How to deal with shared data (such as person). Should the choice be made to only share the concepts and relationships and not implementation

Continued on next page

**Table 11 – continued from previous page**

**RQ1-3, Category:Design, Reference to observation/interview**

- obs.109: rq3-8:19-9: The law states that there are multiple registers. There is a register per profession. The scripts may also need to be formatted that way
- int.32: I-4.2,The prototype shown is not easy for the user to understand. The user not only looks at the functionality, but also at the design. The current design does not comply with the national government web guidelines. The question is whether the user will be able to see through this. It was not part of the research, but it was stated that adjusting the CSS could bring closer to the web guidelines
- obs.104: rq1-36:3-10: What about prototype test scenarios
- obs.103: rq1-69:14-11: Postman application installed and works with the prototype
- int.37: I-4.7,In the current trend, validations are usually located in the business layer. Is that also the case with Ampersand? The validations are spread over the database and surrounding code
- obs.35: rq2-17:10-11: A dutch person has an concept address that must conform to the BRP format (should be a standard building block for it!). A foreign address is unclear what to do with this
- obs.83: rq3-10:12-10: Formatting of the name is not stated literally in the law, but must conform to BRP standards
- obs.84: rq3-11:12-10: Matters such as authorization decisions that allow an information system to retrieve BRP data are not found in the law
- obs.64: rq1-71:14-11: The interface also belongs to the design and not just to the prototype. Changing the Create, Read, Use, Delete changes the behavior of the API
- int.28: I-1.10,In addition, the practitioner’s usual working method is that he works from overviews and lists. Ampersand will have to be designed for this with user requirements, because these things are not mentioned in the law. The law does not support a method and approach. This will have to be a so-called co-creation between IT and business
- int.4: I-3.4,The aim should not be to record everything that is stated in the law in an ICT system. That makes it very rigid. Make sure that 80% of the situations are supported and leave the rest to the employees. Ampersand is very suitable for this, precisely because it has a reactive approach and therefore does not prescribe how the practitioners should act
- int.5: I-3.5,The aim of an ICT system should be to do as little manual work as possible. And when a new law is being developed, an ICT representative should be present
- int.38: I-4.8,How is the maintenance of the system? A new model is always made with the help of Ampersand. The data will have to be migrated itself. Ampersand does not support that. Usually the data structure is taken into account in advance so that as little conversion as possible has to take place. This means that a system is getting bigger and less manageable. So the strength of Ampersand is that this is prevented because a new core system is always being built and the effort is in the data conversion and the connection of adjacent systems
- int.34: I-4.4,A use case can also be devised for the use of rebuilding existing systems. Through the analysis with the help of Ampersand, a system can be rebuilt in which the waste has been cut away. The question is how much this waste would be. Worth a try
- obs.17: rq4-2 The api link works fine, but entire messages return. These should actually get codes
- obs.72: rq4-4 The interface produces many messages and these remain
- obs.10: rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)
- obs.123: rq2-14:19-10: The role gives control to the user. A user is authorized for use. It indicates which user is allowed to use the function

Continued on next page

**Table 11 – continued from previous page****RQ1-3, Category:Design, Reference to observation/interview**

obs.71: rq2-15:19-10: In the interface a FOR can also be used. This populates user roles

obs.100: rq2-9:7-10: Subscription time is added automatically. This is done by means of a rule

Table 12: List of observations **RQ1-4, Category:Registry systems, Reference to observation/interview**

**RQ1-4, Category:Registry systems, Reference to observation/interview**

int.17: I-2.10,Ampersand’s approach is in line with the Registerkern, but not at the implementation level. It doesn’t seem possible to implement Ampersand directly, but the analysis seems quite useful for extending Registerkern. Where generality is discovered and for the specific parts of the law. Then we are talking about a conceptual link and not a technical one

int.13: I-2.6,Registerkern its terminology includes things and products. Every service, read implementation of a law, we call a product. There are standard parts that always appear in every register. These are pre-modeled within Registerkern. This includes a base for each registry and can be expanded according to the needs of the registry. The basis is the minimum common denominator of the registers. Extendable to specific elements arising from the law. There is certainly overlap in the data obtained from the analysis of the big law and the Registerkern. About 80% of the Registerkern is generic and the other 20% is customised. So all new registers have the same basic principles and for the most part run on the same software

int.14: I-2.7,Another aspect of the terminology is that items with the same definition are named differently within the law and within the Registerkern. In Registerkern we are talking about business and products, while the law is big about registrations, applications and professional registers. A mapping of the terms used will have to take place

int.16: I-2.9,The usual procedure within a register is the application process for a registration. The Registerkern has a wizard for this, which includes a diploma check, for example. This diploma check is also part of the current implementation of the Wet-BIG

obs.105: rq1-85:30-11: A new structure where the registers can operate independently of each other, with only the generic elements as common items

obs.106: rq1-92:14-12: Basically trying to create its own container per register. Multi-context problem. This makes it impossible to isolate these containers

obs.109: rq3-8:19-9: The law states that there are multiple registers. There is a register per profession. The scripts may also need to be formatted that way

obs.108: rq3-9:19-9: The structure of the register’s is the same, registers are also called registrations

Table 13: List of observations **RQ1-5, Category:Analysing law, Reference to observation/interview**

**RQ1-5, Category:Analysing law, Reference to observation/interview**

obs.75: rq3-4:12-9 There are more laws involved than just the Wet-BIG. rq3-6 12-9 In addition to the law, decisions are also important

obs.78: rq1-26:12-9: Also the laws and the regulations can still have references to other laws and regulations. Because they can be based on these laws or extend it

Continued on next page

**Table 13 – continued from previous page**

**RQ1-5, Category:Analysing law, Reference to observation/interview**

obs.79: rq1-27:12-9: There are also laws and regulations that are not included in this particular law, but are valid from a higher law (implicit references). In case of Wet-BIG this could be eg the Archives Act or the Time Limits Act and Criminal Law

obs.77: rq1-29:12-9: Not all law- and regulations using Wet-BIG can be found under the search term "big"

obs.82: rq3-15:24-10: In the law the nationality is mentioned, it also refers to the EU and non-eu residents. It is not recognized that the nationality definition is defined per country

obs.80: rq1-23:24-10: law Reading is a skill

obs.43: rq1-35:14-9: The law has been drawn up in Dutch, which means that the Conceptual analysis can also be done in Dutch

obs.50: rq1-41:19-10: The "wettenbank" website contains a persistent hyperlink, which can be used in the documentation as reference

obs.76: rq1-42:21-10: It is easy to deviate from the legal texts. Because they are so hard to read. Some knowledge of the law or the process means that your own interpretation is quickly made. Action research also means that you quickly fall into this trap

obs.37: rq2-1 Substantively includes Wet-BIG also includes disciplinary law (tuchtrecht), which is another branch of sport

obs.33: rq3-2 By reading the law, a structure becomes clear. The concept Person, Registration and Registration with management and Discipline(Discipline) with measures

obs.96: rq1-24:12-9 XML download from wetBig seems like a logical step for the analysis and processing, but it is too complex. This also applies to the JSON structure. Both structures are not pleasant to read. The thought that comes to mind here is why SDU doesn't directly annotate the concepts and relationships

obs.85: rq1-25:12-9: First make overview of all laws and regulations

obs.97: rq1-31:14-9: Besides XML and JSON, RTF and PDF are also an option. In rtf (doc) you can add items in the margins via "comments". With a PDF, annotations and color highlighting can be given this feature

int.29: I-1.11,The question is whether the wet-big is very suitable for this approach. The original law dates from 1993 and it is based on the legislation of 1865

int.26: I-1.8,Ampersand's possible positioning is to use it as an interpreter of legislation and regulations. Then maintain the current analysis and development process and use the prototype to validate the analysis. The question is whether this approach will not result in additional work compared to the current working method. There is a certain skepticism towards Ampersand

int.3: I-3.3,The Wet-BIG offers a lot of room for interpretation. This interpretation possibility means that the law may lend itself less to an Ampersand translation than a recent law would. The new laws have therefore been drafted more carefully. The law provides a framework and the question is how far one should go with recording. This law gives the freedom to fill in matters yourself

int.6: I-3.6,Because the law was drafted some time ago, the definitions are not always unambiguous. And because of the aforementioned interpretation possibility, the legislator can interpret the law slightly differently through jurisprudence

int.31: I-4.1,The Wet-BIG is big and also old. Ampersand could help detect inconsistencies in the law

obs.107: rq1-93:19-12: Implementation choice for separate registers has an impact on the whole. How to deal with shared modules. How to deal with shared data (such as person). Should the choice be made to only share the concepts and relationships and not implementation

Continued on next page

**Table 13 – continued from previous page****RQ1-5, Category:Analysing law, Reference to observation/interview**

obs.34: rq1-95:29-12: The format of a concept big number is not included in the law

obs.125: rq3-13:17-10: There is no list of specialties in Wet-BIG, where is it?

Table 14: List of observations **RQ1-6, Category:Amperсанд as tool, Reference to observation/interview****RQ1-6, Category:Amperсанд as tool, Reference to observation/interview**

obs.57: rq1-15:4-10 With include statements the order of the contents of the document is determined. The expectation was that includes are needed to link parts of code together but includes are not everywhere necessary to get the code working

obs.56: rq1-81:20-11: Compilation error due to a include that no longer existed. Observation here is that an adl has been renamed or moved or deleted. The tool Visual Studio Code does not support a refactoring stroke on said changes

obs.58: rq1-82:20-11: include don't always seem necessary on compilation. It is not entirely clear when this is necessary or not. Another function of includes is to format the analysis

obs.59: rq1-90:14-12: Collection model of regulations than by means of includes keep it small and therefore clear. This is for the reusability of the script. One module per feature

obs.92: rq1-49:30-10: Isolating a pattern or subsystem for testing does not work. This has to do with setting up Docker and possible ignorance on my part

obs.55: rq1-63:10-11: Ampersand is flexible by extension concepts and relationships. Such as dividing an address into street name, house number and addition is quickly realized. Actual address formatting is not in the law. The usual method within the government is to conform to BRP use of addresses

obs.105: rq1-85:30-11: A new structure where the registers can operate independently of each other, with only the generic elements as common items

obs.30: rq1-89:7-12: Items named as common concepts

obs.106: rq1-92:14-12: Basically trying to create its own container per register. Multi-context problem. This makes it impossible to isolate these containers

obs.60: rq1-12: At the start it is not clear when a capital letter or small letter should be used with the crud in the interface

obs.113: rq1-50:30-10: The represent statement makes the interface react differently. When using the represent statement, the append option ("+") disappears

obs.72: rq4-4 The interface produces many messages and these remain

obs.61: rq1-40:10-10: The concepts used in the interface must be of type "object" (represent). The concept may therefore not be alpha or integer

obs.62: rq1-53:2-11: The crud (Create, Read, Use, Delete) and CRUD in the interface don't always work as it should be. There is no full validation on usage. So an on/off does not make sense everywhere. rq1-37:3-10: CRUD/crud options also need some study before they can be applied properly

obs.65: rq1-83:27-11: Experiment with HTML view within the interface fails. Documentation of this is not conclusive. The examples are not enough

obs.98: rq1-9 Adding pieces of php code in the script is possible, but it is not clear how

obs.26: rq1-48:27-10: The concept current date is solved very complicated. But eventually it works. Current time does not seem to have developed yet. Although the example scripts seem to say something different

Continued on next page

**Table 14 – continued from previous page**

**RQ1-6, Category:Ampersand as tool, Reference to observation/interview**

- obs.101: rq2-16:19-10/11-11: Ampersand has a hard time determining a period. Ampersand cannot calculate out of the box. This requires the php functions, which are also not easy to allocate
- int.23: I-1.5,To be able to use Ampersand it would be useful to avoid having to write code in C#
- int.25: I-1.7,Ampersand does not support a maintenance cycle. There must be a solution for this
- int.11: I-2.4,Does Ampersand support databases other than just MariaDB? Not at the moment, but it is to be expected that this will be possible
- int.12: I-2.5,When maintenance takes place on the model, how do we get from one model to another. So how does the IST go to SOLL situation. Ampersand is always creating a new model. So when the law is changed and a new model is needed as a result, Ampersand will produce a completely new model. As a result, no technical debt will remain in the model. It is always a new model. However, the challenge will be in the data migration from the old to the new model
- obs.10: rq4-7 What happens if Ampersand is implemented and there are changes in the structure (normal for software)
- int.34: I-4.4,A use case can also be devised for the use of rebuilding existing systems. Through the analysis with the help of Ampersand, a system can be rebuilt in which the waste has been cut away. The question is how much this waste would be. Worth a try
- obs.68: rq1-10 The function HTML href with target blank does not work within the interface rq1-77:20-11: In HTML mode the <a href="x" target=\_blank> is not supported. The target is removed in the compilation
- obs.114: rq1-65:10-11: DATETIME (represent) field could not be converted to Excel. The compilation process hangs on this
- obs.86: rq1-66:10-11: XLSX files format is created partly on the basis of multiplicity. one on n relation produces its own tab
- obs.31: rq1-91:14-12: A concept and a relation can be defined several times within your own patterns. So that the patterns can stand on their own
- obs.66: rq1-98:30-12: When using linkto in the interface as last element in the interface and the signature occurs more often than a dropdown to all subinterfaces (of the same signature) appears
- obs.100: rq2-9:7-10: Subscription time is added automatically. This is done by means of a rule

Table 15: List of observations **RQ1-7, Category:Not-categorised, Reference to observation/interview**

**RQ1-7, Category:Not-categorised, Reference to observation/interview**

- int.10: I-2.3,Nice that Ampersand is an open source product. There is not much to be found. Only the github repository can be found
- int.7: I-3.7,The law consists of the following parts. Going through the law should be a first step for the conceptual analysis
- obs.47: rq1-11 Implementation in Docker with RAP creates new directories all the time
- obs.22: rq1-16 Notation method of Concept and Relations and Rules are defined for a very small part. Only the first position is uppercase or lowercase. There is no rule about other spelling. So using CamelCase or underscore or hyphen
- obs.127: rq1-22 The tool VSC also doesn't have a generic search option across the adls
- obs.74: rq3-3:12-9 There are parts of the law that are no longer valid, they are not included

Continued on next page

**Table 15 – continued from previous page**

**RQ1-7, Category:Not-categorised, Reference to observation/interview**

<p>obs.128: rq1-32:14-9: The tool VSC has an Ampersand extension. It hangs once in a while</p> <p>obs.73: rq1-33:9-1: VSC does not support the latex environment well. My PC often hangs on this.</p> <p>rq1-87:3-12: Latex can also be written in VSC. Apparently it is a different version, because the import does not immediately succeed. Does not work really well and the result is poor</p> <p>obs.94: rq1-34:14-9: The spelling of a pattern is capitalized and the pattern ends with an end-pattern. Multiple patterns are possible within one script</p> <p>obs.39: rq1-43:23-10: The order of the data in the Conceptual analysis is a bit strange. First the definition is shown, then the name of the relation and below that the meaning again</p> <p>obs.40: rq1-44:23-10: In the Conceptual analysis enters must be taken into account in the texts. These come back directly in the documents and then yield broken sentences</p> <p>obs.67: rq1-5:30-10: The browser is holding data from the interface and periodically the cache needs to be cleared for customization to work</p> <p>obs.51: rq1-54:2-11: The documentation can be written in different ways. This can be done using mark down, html and latex</p> <p>obs.99: rq1-57-2:7-11: Parts like next big number or now() and today() are better solved in a dev language, like php</p> <p>obs.41: rq1-76:20-11: The "disclaimer" does not appear in the Conceptual analysis</p> <p>obs.21: rq1-86:30-11: Classify is a specialization of a concept. No experience has been gained with this</p> <p>obs.91: rq1-88:5-12: Tried the tool Obsidian as a new tool. But here too I do not get an immediate overview and it is digital. Apparently writing in a log is more convenient for me</p> <p>obs.32: rq2-7:4-10: A concept Person is not equal to BIG-number. A big number is an attribute of the registration. A person can have multiple BIG-numbers</p> <p>obs.36: rq3-14:19-10: A concept person and a big number are very different things. Person is immutable, big number is not. They do have a relation with each other</p>
--



## E. List of Functional Requirements

Annotation, An annotation tool is required to maintain an overview of the text to be processed. This prevents things from being processed twice or not., 49

Atlas availability, Make Atlas available outside the RAP environment., 57

information in Atlas, Being able to edit the Atlas information from Atlas, 57

refactor, There is a need to enable refactoring within an IDE. We can then prevent issues when removing, for example, Includes. Changing naming or viewing where a Concept or Relation occurs is highly desirable., 49

shared components, Dealing with shared components such as Concepts, Relations or Patterns. This both within a project and across the projects., 50

## F. Adl scripts

```
1 CONTEXT Persoon IN DUTCH
2
3 PATTERN Persoon
4 -- The concepts
5 -- Persoon
6 CONCEPT Persoon "Een Persoon representeerd een persoons-id dat opgenomen is in het BIG-register."
7 REPRESENT Persoon TYPE OBJECT
8 PURPOSE CONCEPT Persoon
9 {+
10 In artikel 3 lid 2 wordt aangegeven dat bij elke inschrijving in het register de naam, voornamen, geslacht,
    geboortedatum, nationaliteit en adres van de betrokkene en het nummer en het tijdstip van inschrijving wordt vermeld. Bij
    ministeriële regeling kunnen gegevens worden aangewezen die ten behoeve van het identificeren van beroepsbeoefenaren bij de
    inschrijving worden vermeld.
11 Deze beroepsbeoefenaren zijn personen.
12 +}
13 POPULATION Persoon CONTAINS [ "P001", "P002", "P003" ]
14
15
16 --naam
17 CONCEPT Naam "Aanduiding van de familienaam zoals vastgelegd in de BRP."
18 REPRESENT Naam TYPE ALPHANUMERIC
19 PURPOSE CONCEPT Naam
20 {+
21 In artikel 3 lid 2 is aangegeven dat de naam een onderdeel is van de identificatie van de zorgverlener.
22 +}
23 {-zoeken naar de data binnen de BRP-}
24 POPULATION Naam CONTAINS
25 [
26 "Edelaar",
27 "Jansen",
28 "Pietersen" ]
29
30
31 --voornaam
32 CONCEPT Voornaam "Alle voornamen van de Persoon zoals dit is vastgelegd binnen de BRP."
33 REPRESENT Voornaam TYPE ALPHANUMERIC
34 PURPOSE CONCEPT Voornaam
35 {+
36 In artikel 3 lid 2 is aangegeven dat de voorna(m)en een onderdeel is/zijn van de identificatie van de zorgverlener.
37 +}
38
39 ---- relaties
40 RELATION naam [Persoon*Naam][UNI,TOT,SUR]
41 PRAGMA "De persoon met het id " " wordt " " genoemd."
42 MEANING "Elke ingeschrevene moet een naam hebben en een naam kan bij meerdere personen behoren."
43 POPULATION naam [Persoon*Naam] CONTAINS
44 [
45 ("P001", "Edelaar"),
46 ("P002", "Jansen"),
47 ("P003", "Pietersen" ]
48 --IDENT "Persoon" : Persoon(naam[Persoon*Naam])
49 ROLE USER MAINTAINS TotNaam
50 RULE TotNaam : I[Persoon] |- naam[Persoon*Naam];naam[Persoon*Naam]-
51 MEANING "meaning"
52 MESSAGE "Er moet een naam ingevuld worden."
53 VIOLATION ( TXT "Voor persoon ", SRC I , TXT " is geen naam ingevuld." )
54
55
56 RELATION voornaam [Persoon*Voornaam][UNI]
57 PRAGMA "De persoon met het id " " heeft " " als voornaam."
58 MEANING "Elke ingeschrevene moet een voornaam hebben."
59 POPULATION voornaam [Persoon*Voornaam] CONTAINS
60 [ ("P001","Gerard"),
61 ("P002","Jan"),
62 ("P003","Piet" ) ]
63 ROLE USER MAINTAINS TotVoornaam
64 RULE TotVoornaam : I[Persoon] |- voornaam[Persoon*Voornaam];voornaam[Persoon*Voornaam]-
65 MEANING "meaning"
66 MESSAGE "Er moet een voornaam ingevuld worden."
67 VIOLATION ( TXT "Voor persoon ", SRC I , TXT " is geen voornaam ingevuld." )
68
69
70
71 RELATION geboortedatum [Persoon*Datum][UNI]
72 MEANING "Elke ingeschrevene heeft een geboortedatum"
73 POPULATION geboortedatum [Persoon*Datum] CONTAINS
74 [ ("P001",2000-01-01),
75 ("P002",1999-01-01),
76 ("P003",1970-12-13) ]
77 ROLE USER MAINTAINS TotGeboortedatum
78 RULE TotGeboortedatum : I[Persoon] |- geboortedatum[Persoon*Datum];geboortedatum[Persoon*Datum]-
79 MEANING "meaning"
80 MESSAGE "De geboortedatum ingevuld worden."
81 VIOLATION ( TXT "Voor persoon ", SRC I, TXT " is geen geboortedatum ingevuld." )
82
83
84 RELATION geslacht [Persoon*Geslacht][UNI]
85 MEANING "Elke ingeschrevene behoort tot een geslacht"
```

```

86     POPULATION geslacht [Persoon*Geslacht] CONTAINS [
87         ("P001", "M"),
88         ("P002", "V"),
89         ("P003", "X") ]
90     ROLE USER MAINTAINS TotGeslacht
91     RULE TotGeslacht : I[Persoon] |- geslacht[Persoon*Geslacht];geslacht[Persoon*Geslacht]-
92     MEANING "meaning"
93     MESSAGE "Het geslacht moet ingevuld worden."
94     VIOLATION ( TXT "Voor persoon ", SRC I, TXT " is geen geslacht ingevuld.")
95
96
97     RELATION nationaliteit [Persoon*Nationaliteitid]
98     MEANING "Elke ingeschrevene heeft een nationaliteit"
99     POPULATION nationaliteit [Persoon * Nationaliteitid] CONTAINS
100    [ ("P001", "0001"),
101      ("P002", "0001"),
102      ("P003", "0052") ]
103     ROLE USER MAINTAINS "Elke persoon heeft een nationaliteit"
104     RULE "Elke persoon heeft een nationaliteit" : I[Persoon] |- nationaliteit[Persoon*Nationaliteitid];nationaliteit[Persoon*
Nationaliteitid]-
105     MEANING "meaning"
106     MESSAGE "De nationaliteit moet ingevuld worden."
107     VIOLATION ( TXT "Voor persoon ", SRC I, TXT " is geen nationaliteit ingevuld.")
108
109
110
111     RELATION adres [Persoon*Adres]
112     MEANING "Elke ingeschrevene heeft een adres"
113     POPULATION adres [Persoon * Adres] CONTAINS
114     [ ("P001", "adres1"),
115       ("P002", "adres2"),
116       ("P003", "adres3") ]
117     ROLE USER MAINTAINS TotAdres
118     RULE TotAdres : I[Persoon] |- adres[Persoon*Adres];adres[Persoon*Adres]-
119     MEANING "meaning"
120     MESSAGE "Het adres moet ingevuld worden."
121     VIOLATION ( TXT "Voor persoon ", SRC I, TXT " met naam ", SRC naam[Persoon*Naam], TXT " is geen adres ingevuld.")
122
123     PURPOSE RULE "Create Inschrijving"
124     {+Nieuw persoon moet ingeschreven worden.+}
125     ROLE ExecEngine MAINTAINS "Create Inschrijving"
126     RULE "Create Inschrijving" : I[Persoon] |- inschrijving[Persoon*InschrijfId];inschrijving[Persoon*InschrijfId]-
127     VIOLATION ( TXT "{EX} InsAtom;InschrijfId"
128                 , TXT "{EX} InsPair;inschrijving;Persoon;", SRC I, TXT ";InschrijfId;.NEW"
129                 )
130
131
132
133     ENDPATTERN
134
135
136     ENDCONTEXT

```

Listing 8: Persoon

```

1 CONTEXT Persoon IN DUTCH
2
3 INCLUDE "Geslacht.adl"
4 INCLUDE "Nationaliteit.adl"
5 INCLUDE "Adres.adl"
6 INCLUDE "Inschrijving.adl"
7 INCLUDE "Persoon.adl"
8
9 INTERFACE "Personen" FOR USER: V[SESSION*Persoon] CRud
10 BOX <TABLE sortable title="Persoon">
11 [ "Naam" : naam [Persoon*Naam] CRUD
12 , "Voorna(a)m(en)" : voornaam [Persoon*Voornaam] CRUD
13 , "Geslacht" : geslacht [Persoon*Geslacht] cRUd
14 , "Adres" : adres [Persoon*Adres] CRUD
15 BOX <TABLE>
16 [
17 "" : I LINKTO INTERFACE Adres
18 ]
19 , "Geboortedatum" : geboortedatum [Persoon*Datum] CRUD
20 , "Nationaliteit" : nationaliteit [Persoon*Nationaliteitid] cRUd
21 , "Inschrijving" : inschrijving [Persoon*InschrijfId] cRUd
22 BOX<TABLE>
23 [
24 "" : I LINKTO INTERFACE Inschrijving
25 ]
26 ]
27
28 INTERFACE "Persoon" FOR USER: I[Persoon] CRud
29 BOX
30 [ "Persoon" : I[Persoon] CRud
31 , "Naam" : naam [Persoon*Naam] CRUD
32 , "Voorna(a)m(en)" : voornaam [Persoon*Voornaam] CRUD
33 , "Geslacht" : geslacht [Persoon*Geslacht] cRUd
34 BOX <TABLE>
35 [ "Code" : I[Geslacht] cRUd
36 , "Omschrijving" : geslacht [Geslacht*Omschrijving] cRUd
37 ]
38 , "Adres" : adres [Persoon*Adres] CRUD
39 , "Geboortedatum" : geboortedatum [Persoon*Datum] CRUD
40 , "Nationaliteit" : nationaliteit [Persoon*Nationaliteitid] cRUd
41 , "Inschrijving" : inschrijving [Persoon*InschrijfId] cRUd
42 BOX<TABLE>
43 [
44 "" : I LINKTO INTERFACE Inschrijving
45 ]
46 ]
47
48
49
50 VIEW "Persoonsnaam": Naam
51 {
52 "Naam": naam~
53 }
54 HTML TEMPLATE "View-FILEOBJECT.html" ENDVIEW
55
56 VIEW "Adres": Adres
57 {
58 "Adres" : binnenlandsadres[Adres*Binnenlandsadres]
59 } HTML TEMPLATE "View-FILEOBJECT.html" ENDVIEW
60
61
62 VIEW "Inschrijving": Inschrijving
63 {
64 "Inschrijving": "Inschrijving"
65 }
66 ENDVIEW
67
68 ENDCONTEXT

```

Listing 9: PersoonInterface

```

1 CONTEXT Big IN DUTCH
2
3 INCLUDE "Generic.adl"
4 INCLUDE "Register.adl"
5 INCLUDE "Inschrijfduur.adl"
6 INCLUDE "Registratie.adl"
7 INCLUDE "PersoonI.adl"
8
9 PATTERN Inschrijving
10 PURPOSE PATTERN Inschrijving
11 {+
12   Inschrijving legt de relatie vast tussen Persoon en het Register.
13 +}
14
15 CONCEPT InschrijfId "De aanmelding van persoon in een register"
16 REPRESENT InschrijfId TYPE OBJECT
17 PURPOSE CONCEPT InschrijfId
18 {+In artikel 3 lid 2 wordt aangegeven dat bij elke inschrijving worden in het register vermeld de naam, voornamen,
    geslacht, geboortedatum, nationaliteit en adres van de betrokkene en het nummer en het tijdstip van inschrijving. Bij ministeriële
    regeling kunnen gegevens worden aangewezen die ten behoeve van het identificeren van beroepsbeoefenaren bij de inschrijving worden
    vermeld.
19 +}
20 RELATION inschrijving [Persoon*InschrijfId][TOT,INJ]
21 MEANING "Elk persoon die BIG geregistreerd wil zijn, moet zijn ingeschreven. Een persoon kan meerdere inschrijvingen hebben."
22 POPULATION inschrijving[Persoon*InschrijfId] CONTAINS
23   [{"P001", "I001"}
24   ,{"P002", "I002"}
25   ,{"P003", "I003"}]
26
27
28
29 --tijdstip van inschrijving
30 CONCEPT InschrijfTijdstip "Het inschrijftijdstip bevat de datum en tijdstip van inschrijving in Y-m-d h:i:s-formaat."
31 --REPRESENT InschrijfTijdstip TYPE DATETIME
32 PURPOSE CONCEPT InschrijfTijdstip
33 {+
34   In artikel 3 lid 2 is aangegeven dat de datum en het tijdstip van inschrijving een onderdeel is van de identificatie van
    de zorgverlener.
35 +}
36 RELATION inschrijftijdstip [InschrijfId*InschrijfTijdstip][UNI]
37 MEANING "Elke inschrijving vindt plaats op een tijdstip."
38 {- POPULATION inschrijftijdstip [Inschrijving*InschrijfTijdstip] CONTAINS
39   [ ("I001",2015-06-03T13:21:58Z),
40     ("I002",2016-06-03T13:21:58Z),
41     ("I003",2017-06-03T13:21:58Z) ]
42   -}
43 PURPOSE RULE "Voeg_inschrijftijd_toe_(automatisch)"
44 {+
45   Het tijdstip waarop de inschrijving wordt vastgelegd.
46 +}
47 ROLE "ExecEngine" MAINTAINS "Voeg_inschrijftijd_toe_(automatisch)"
48 RULE "Voeg_inschrijftijd_toe_(automatisch)" : I[InschrijfId] |- inschrijftijdstip [InschrijfId*InschrijfTijdstip];
    inschrijftijdstip [InschrijfId*InschrijfTijdstip]-
49 VIOLATION ( TXT "{EX} InsAtom;InschrijfTijdstip"
50             , TXT "{EX} InsPair;inschrijftijdstip;InschrijfId;" , SRC I , TXT ";InschrijfTijdstip;{php}date(DATE_ISO8601)" --
    Set the DateTime
51           )
52
53
54 {- het bepalen van het volgende BIG-nummer is beter op te lossen een programmeertaal en niet binnen AMPersand-}
55 CONCEPT Bignummer "Het identificatienummer van de BIG-ingeschrevene."
56 REPRESENT Bignummer TYPE OBJECT
57 PURPOSE CONCEPT Bignummer
58 {+
59   In artikel 3 lid 2 wordt aangegeven dat bij elke inschrijving in het register de naam, voornamen, geslacht,
    geboortedatum, nationaliteit en adres van de betrokkene en het nummer en het tijdstip van inschrijving wordt vermeld. Bij
    ministeriële regeling kunnen gegevens worden aangewezen die ten behoeve van het identificeren van beroepsbeoefenaren bij de
    inschrijving worden vermeld.
60   Het BIG-nummer identificeert de BIG-ingeschrevene.
61 +}
62
63 RELATION bignummer[InschrijfId*Bignummer][UNI,INJ]
64 PURPOSE RELATION bignummer
65 {+
66   De koppeling tussen een Persoon en een Bignummer. Dit is een één op één koppeling die automatisch wordt aangebracht.
67 +}
68 POPULATION bignummer CONTAINS
69   [ ("I001", "B001")
70     , ("I002", "B002")
71     , ("I003", "B003") ]
72 ROLE USER MAINTAINS TotInschrijvingBig
73 RULE TotInschrijvingBig : I[InschrijfId] |- bignummer[InschrijfId*Bignummer];bignummer[InschrijfId*Bignummer]-
74 MEANING "meaning"
75 MESSAGE "Er moet een bignummer ingevuld worden."
76 VIOLATION ( TXT "Voor inschrijving " , SRC I , TXT " is geen bignummer aangemaakt." )
77
78 PURPOSE RULE "Voeg_Bignummer_toe(automatisch)"
79 {+
80   Nieuw persoon krijgt een Bignummer.
81 +}
82 ROLE "ExecEngine" MAINTAINS "Voeg_Bignummer_toe(automatisch)"

```

```

83     RULE "Voeg_Bignummer_toe(automatisch)" : I[InschrijfId] |- bignummer[InschrijfId*Bignummer];bignummer[InschrijfId*Bignummer
]~
84     VIOLATION ( TXT "{EX} InsAtom;Bignummer"
85               , TXT "{EX} InsPair;bignummer;InschrijfId;", SRC I, TXT ";Bignummer;_NEW"
86               )
87
88     RELATION inschrijving[RegisterId*InschrijfId][INJ]
89     MEANING "Het vastleggen van de koppeling tussen het register en de inschrijving."
90     POPULATION inschrijving[RegisterId*InschrijfId] CONTAINS
91     [ ("1","I001")
92     , ("2","I002")
93     , ("1","I003" ) ]
94
95     VIEW RegisterId: RegisterId DEFAULT
96     { register : register
97     } ENDVIEW
98
99
100    RELATION deelToegang [InschrijfId*Ja_of_Nee][UNI]
101    PURPOSE RELATION deelToegang [InschrijfId*Ja_of_Nee]
102    {+
103    In Artikel 3, lid 4 van de Wet Big wordt aangegeven dat indien de inschrijving van een beroepsbeoefenaar plaatsvindt op
basis van een gedeeltelijke toegang als bedoeld in artikel 12, eerste lid, van de Algemene wet erkenning EU-beroepskwalificaties,
wordt dit bij de inschrijving aangetekend, waarbij wordt vermeld voor welke beroepswerkzaamheden de gedeeltelijke toegang geldt en
onder welke beroepstitel die beroepsbeoefenaar zijn beroepswerkzaamheden op grond van artikel 12, derde lid, uitvoert.
104    +)
105    PURPOSE RULE "Voeg_default_deeltoegang_toe.(automatisch)"
106    {+
107    De default waarde voor de deeltoegang is "nee"
108    +)
109    ROLE "ExecEngine" MAINTAINS "Voeg_default_deeltoegang_toe.(automatisch)"
110    RULE "Voeg_default_deeltoegang_toe.(automatisch)" : I[InschrijfId] |- deelToegang [InschrijfId*Ja_of_Nee];deelToegang [
InschrijfId*Ja_of_Nee]~
111    VIOLATION ( TXT "{EX} InsPair;deelToegang;InschrijfId;", SRC I, TXT ";Ja_of_Nee;Nee"
112              )
113
114
115    ENDPATTERN
116
117    INTERFACE Inschrijvingen FOR USER: V[SESSION*InschrijfId] CRud
118    BOX<TABS>
119    ["Inschrijving": I[InschrijfId] CRud
120    BOX <TABLE sortable title="Inschrijving" >
121    [
122    "Inschrijving" : I[InschrijfId] CRud
123    , "Persoon" : inschrijving[Persoon*InschrijfId]~ CRud
124    BOX<TABLE>
125    [ "Persoon" :I LINKTO INTERFACE Persoon
126    , "Naam" : naam[Persoon*Naam] cRud
127    ]
128    , "RegisterId" : inschrijving[RegisterId*InschrijfId]~ cRud
129    , "Big-nummer" : bignummer[InschrijfId*Bignummer] cRud
130    , "Inschrijvingstijd":inschrijftijdstip[InschrijfId*InschrijfTijdstip] cRud
131    , "Deeltoegang" : deelToegang[InschrijfId*Ja_of_Nee] cRud
132    , "Registratie" : registratie[Registratie*InschrijfId]~ cRud
133    ]
134    ,
135    "Diploma": I[InschrijfId]
136    BOX<TABLE>
137    ["Datums" : I[InschrijfId]
138    , "Diplomadatum" : diplomadatum[InschrijfId*Datum] CRud
139    , "verklaringdatum" : verklaringdatum[InschrijfId*Datum] CRud
140    , "getuigschriftdatum" : getuigschriftdatum[InschrijfId*Datum]CRud
141    , "kwalificatiedatum" : kwalificatiedatum[InschrijfId*Datum] cRud
142    ]
143    ,
144    "Weigering" : I[InschrijfId]
145    BOX <TABLE>
146    ["Weigering" : I[InschrijfId]
147    , "Weigeringsgrond" : inschrijvingsWeigering[InschrijfId*Weigering] cRud
148    ]
149    ]
150
151
152    INTERFACE Inschrijving FOR USER: I[InschrijfId] CRud
153    BOX<TABS>
154    ["Inschrijving": I[InschrijfId] CRud
155    BOX <TABLE sortable title="Inschrijving" >
156    [
157    "Inschrijving" : I[InschrijfId] CRud
158    , "Persoon" : inschrijving[Persoon*InschrijfId]~ CRud
159    BOX<TABLE>
160    [ "Persoon" :I LINKTO INTERFACE Persoon
161    , "Naam" : naam[Persoon*Naam] cRud
162    ]
163    , "RegisterId" : inschrijving[RegisterId*InschrijfId]~ cRud
164    , "Big-nummer" : bignummer[InschrijfId*Bignummer] cRud
165    , "Inschrijvingstijd":inschrijftijdstip[InschrijfId*InschrijfTijdstip] cRud
166    , "Deeltoegang" : deelToegang[InschrijfId*Ja_of_Nee] cRud
167    , "Registratie" : registratie[Registratie*InschrijfId]~ cRud
168    ]

```

```

169 ,
170 "Diploma": I[InschrijfId]
171   BOX<TABLE>
172     ["Datums" : I[InschrijfId]
173     , "Diplomadatum" : diplomadatum[InschrijfId*Datum] CRUd
174     , "verklaringdatum" : verklaringdatum[InschrijfId*Datum] CRUd
175     , "getuigschriftdatum" : getuigschriftdatum[InschrijfId*Datum]CRUd
176     , "kwalificatiedatum" : kwalificatiedatum[InschrijfId*Datum] cRUd
177     ]
178   ,
179 "Weigering" : I[InschrijfId]
180   BOX <TABLE>
181     ["Weigering" : I[InschrijfId]
182     , "Weigeringsgrond" : inschrijvingsWeigering[InschrijfId*Weigering] cRUd
183     ]
184   ,
185 "Doorhaling" : I[InschrijfId]
186   BOX <TABLE>
187     ["Doorhaling" : I[InschrijfId]
188     , "Doorhalinggrond" : doorgehaald [InschrijfId*Doorhaling] cRUd
189     ]
190   ]
191 ]
192
193
194
195
196 ENDCONTEXT

```

Listing 10: Inschrijving

```

1 CONTEXT Register IN DUTCH
2 INCLUDE "Generic.adl"
3 INCLUDE "../1_artsRegister/Main.adl"
4 INCLUDE "../2_tandartsRegister/Main.adl"
5 INCLUDE "../3_apothekerRegister/Main.adl"
6 INCLUDE "../4_gezondheidszorgpsycholoogRegister/Main.adl"
7 INCLUDE "../5_psychotherapeutRegister/Main.adl"
8 INCLUDE "../6_fysiotherapeutRegister/Main.adl"
9 INCLUDE "../7_verloskundigeRegister/Main.adl"
10 INCLUDE "../8_verpleegkundigeRegister/Main.adl"
11 INCLUDE "../9_physician_assistantRegister/Main.adl"
12 INCLUDE "../10_orthopedagoog_generalistRegister/Main.adl"
13 INCLUDE "../11_klinisch_technoloogRegister/Main.adl"
14
15
16 PATTERN Register
17 PURPOSE PATTERN Register
18 {+
19 Het BIG-register (Beroepen in de Individuele Gezondheidszorg) is een wettelijk, online en openbaar register. Alleen wie in het
BIG-register staat, mag een beschermde beroepstitel voeren en mag de bij het beroep horende voorbehouden handelingen zelfstandig
uitvoeren. Iedereen kan het register raadplegen. Het BIG-register verzorgt ook de erkenning van buitenlandse diplomas.
20 +}
21
22 --concepts
23 CONCEPT RegisterId "Technisch element voor het register."
24
25
26 CONCEPT Register "Een register is een officiële lijst van personen die aan de, door het register gestelde voorwaarden voldoen."
27 PURPOSE CONCEPT Register
28 {+
29 In artikel 1 lid 5 wordt aangegeven dat elk Register wordt ingesteld en beheerd door Onze Minister. In artikel 3 lid 6
wordt gesteld dat de registers worden ingesteld ten einde te kunnen voldoen aan een verzoek om informatie als bedoeld in artikel 12
en ten behoeve van het toezicht op de uitvoering van de artikelen 4 en 17.
30 +}
31 REPRESENT Register TYPE OBJECT
32 POPULATION Register CONTAINS
33 [ "arts", "tandarts", "apotheker", "gezondheidszorgpsycholoog", "psychotherapeut", "fysiotherapeut", "verloskundige", "
verpleegkundige", "physician assistant", "orthopedagoog-generalist", "klinisch technoloog" ]
34
35 RELATION register [RegisterId*Register][UNI,TOT,INJ]
36 POPULATION register[RegisterId*Register] CONTAINS
37 [
38 ("1","arts")
39 , ("2","tandarts")
40 , ("3","apotheker")
41 , ("4","gezondheidszorgpsycholoog")
42 , ("5","psychotherapeut")
43 , ("6","fysiotherapeut")
44 , ("7","verloskundige")
45 , ("8","verpleegkundige")
46 , ("9","physician assistant")
47 , ("10","orthopedagoog-generalist")
48 , ("11","klinisch technoloog" ]
49
50 RELATION ingangsdatum[RegisterId*Datum][UNI,TOT]
51 POPULATION ingangsdatum[RegisterId*Datum] CONTAINS
52 [
53 ("1",2000-01-01)
54 , ("2",2000-01-01)
55 , ("3",2000-01-01)
56 , ("4",2000-01-01)
57 , ("5",2000-01-01)
58 , ("6",2000-01-01)
59 , ("7",2000-01-01)
60 , ("8",2000-01-01)
61 , ("9",2000-01-01)
62 , ("10",2000-01-01)
63 , ("11",2000-01-01) ]
64
65
66 RELATION einddatum[RegisterId*Datum][UNI]
67
68 RELATION getuigschrift[RegisterId*Ja_of_Nee][UNI]
69 POPULATION getuigschrift[RegisterId*Ja_of_Nee] CONTAINS
70 [
71 ("1","Ja")
72 , ("2","Ja")
73 , ("3","Ja")
74 , ("4","Ja")
75 , ("5","Nee")
76 , ("6","Ja")
77 , ("7","Nee")
78 , ("8","Ja")
79 , ("9","Ja")
80 , ("10","Ja")
81 , ("11","Ja")
82 ]
83 ROLE USER MAINTAINS TotGetuigschrift
84 RULE TotGetuigschrift : I[RegisterId] |- getuigschrift[RegisterId*Ja_of_Nee];getuigschrift[RegisterId*Ja_of_Nee]-
85 MEANING "meaning"
86 MESSAGE "Het ja of nee moet ingevuld worden."

```



```

87      VIOLATION ( TXT "Voor register ", SRC I, TXT " is geen ja of nee ingevuld.")
88
89
90
91  ENDPATTERN
92
93  --*** interface
94  INTERFACE Register FOR Beheerder      : "_SESSION" cRud
95  BOX <TABS>
96  [
97      Register                          : V[SESSION+RegisterId] CRud
98      BOX <TABLE sortable title="Register">
99      [
100         "Register"                    : register[RegisterId*Register] CRud
101         , "Ingangsdatum"              : ingangsdatum[RegisterId*Datum] CRUd
102         , "Einddatum"                 : einddatum[RegisterId*Datum] CRUd
103         , "Getuigschrift nodig"       : getuigschrift[RegisterId*Ja_of_Nee] cRUd
104         ]
105         ,
106         Arts: V[SESSION+RegisterId];(I /\ arts[RegisterId*RegisterId] ) cRud
107         BOX <TABLE hideOnNoRecords sortable order="asc" showNavMenu>
108         [
109             "Inschrijvingen"          : inschrijving[RegisterId*InschrijfId] LINKTO INTERFACE Inschrijving
110             , "Persoon"                : inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~ LINKTO INTERFACE Persoon
111             , "Naam"                   : (inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~) ; naam[Persoon*Naam]
112             , "BIG-nummer"             : (inschrijving[RegisterId*InschrijfId];bignummer[InschrijfId*Bignummer]) cRud
113             ]
114             ,
115             Tandarts: V[SESSION+RegisterId];(I /\ tandarts[RegisterId*RegisterId] ) cRud
116             BOX <TABLE hideOnNoRecords sortable order="asc" showNavMenu>
117             [
118                 "Inschrijvingen"      : inschrijving[RegisterId*InschrijfId] LINKTO INTERFACE Inschrijving
119                 , "Persoon"            : inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~ LINKTO INTERFACE Persoon
120                 , "Naam"               : (inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~) ; naam[Persoon*Naam]
121                 , "BIG-nummer"         : (inschrijving[RegisterId*InschrijfId];bignummer[InschrijfId*Bignummer]) cRud
122                 ]
123                 ]
124                 ,
125                 Apotheker: V[SESSION+RegisterId];(I /\ apotheker[RegisterId*RegisterId] ) cRud
126                 BOX <TABLE hideOnNoRecords sortable order="asc" showNavMenu>
127                 [
128                     "Inschrijvingen"   : inschrijving[RegisterId*InschrijfId] LINKTO INTERFACE Inschrijving
129                     , "Persoon"         : inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~ LINKTO INTERFACE Persoon
130                     , "Naam"           : (inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~) ; naam[Persoon*Naam]
131                     , "BIG-nummer"     : (inschrijving[RegisterId*InschrijfId];bignummer[InschrijfId*Bignummer]) cRud
132                     ]
133                     ]
134                 ]
135             ]
136         ]
137     ]
138
139
140
141  ENDCONTEXT

```

Listing 11: Register

```

1 CONTEXT Arts IN DUTCH HTML
2 INCLUDE "../0_generiek/Main.adl"
3
4 PATTERN Arts
5   PURPOSE PATTERN Arts
6   {+
7   Het register voor arts bevat alle attributen van het register arts.
8   +}
9   -- dit mag niet gewijzigd worden
10  RELATION arts[RegisterId*RegisterId]
11    MEANING "Het identificerende nummer van de het artsenregister."
12  POPULATION arts[RegisterId*RegisterId] CONTAINS [{"1","1"}]
13  --
14
15  --specialismen
16  --CONCEPT Specialisme ""
17  --POPULATION Specialisme CONTAINS
18  --["Anesthesioloog","Arts klinische chemie","Arts maatschappij en gezondheid"]
19
20  RELATION specialisme [RegisterId*SpecialismeArts]
21  MEANING "Alle specialismes voor arts."
22  POPULATION specialisme [RegisterId*SpecialismeArts] CONTAINS
23  [
24    ("1","Anesthesioloog"),
25    ("1","Arts klinische chemie"),
26    ("1","Arts maatschappij en gezondheid"),
27    ("1","Arts-microbioloog"),
28    ("1","Bedrijfsarts"),
29    ("1","Cardioloog"),
30    ("1","Cardiothoracaal chirurg"),
31    ("1","Chirurg"),
32    ("1","Dermatoloog"),
33    ("1","Gynaecoloog"),
34    ("1","Huisarts"),
35    ("1","Internist"),
36    ("1","Keel-,neus-en oorarts"),
37    ("1","Kinderarts"),
38    ("1","Klinisch geneticus"),
39    ("1","Klinisch geriater"),
40    ("1","Longarts"),
41    ("1","Maag-,darm-en leverarts"),
42    ("1","Neurochirurg"),
43    ("1","Neuroloog"),
44    ("1","Nucleair geneeskundige"),
45    ("1","Oogarts"),
46    ("1","Orthopedisch chirurg"),
47    ("1","Patholoog"),
48    ("1","Plastisch chirurg"),
49    ("1","Psychiater"),
50    ("1","Radioloog"),
51    ("1","Radiotherapeut"),
52    ("1","Reumatoloog"),
53    ("1","Revalidatiearts"),
54    ("1","Specialist ouderen geneeskunde"),
55    ("1","Sportarts"),
56    ("1","Uroloog"),
57    ("1","Verzekeringsarts")
58  ]
59  RELATION specialist [Registratie*SpecialismeArts]
60  MEANING "De specialist is een uitbreiding van arts met een specialisme."
61
62
63
64  RELATION herregistratie[InschrijfId*Datum]
65  MEANING "Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf
jaar na datum van registratie."
66
67  ROLE USER MAINTAINS DefaultDeelToegang
68  RULE DefaultDeelToegang : I[InschrijfId] |- deelToegang [InschrijfId*Ja_of_Nee];deelToegang [InschrijfId*Ja_of_Nee]~
69  MEANING "meaning"
70  MESSAGE "Het ja of nee moet ingevuld worden."
71  VIOLATION ( TXT "Voor register ", SRC I, TXT " is geen ja of nee ingevuld.")
72
73  ENDPATTERN
74  INTERFACE Artsen FOR MEDEWERKER : V[SESSION*RegisterId];(I /\ arts[RegisterId*RegisterId] ) cRud
75  BOX <TABLE hideOnNoRecords title="Artsen" sortable >
76  [
77    "Inschrijvingen" : inschrijving[RegisterId*InschrijfId] LINKTO INTERFACE Inschrijving
78    , "Persoon" : inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~ LINKTO INTERFACE Persoon
79    , "Naam" : (inschrijving[RegisterId*InschrijfId];inschrijving [Persoon*InschrijfId]~) ; naam[Persoon*Naam]
80    , "BIG-nummer" : (inschrijving[RegisterId*InschrijfId];bignummer[InschrijfId*Bignummer]) cRud
81    , "Registratie" : (arts[RegisterId*RegisterId];inschrijving[RegisterId*InschrijfId]);registratie [Registratie*InschrijfId]
]~ LINKTO INTERFACE RegistratieArts
82
83  ]
84
85  INTERFACE RegistratieArts FOR MEDEWERKER : I[Registratie] cRud
86  BOX <FORM hideOnNoRecords title="Beroepregister" showNavMenu >
87  [ "Register" : registratie [Registratie*InschrijfId];inschrijving[RegisterId*InschrijfId]~ cRud
88  , "Registratie" : I[Registratie];(registratie [Registratie*InschrijfId];registratie [Registratie*InschrijfId]~) LINKTO
INTERFACE RegistratieArts

```

```

89 , "Inschrijving" : registratie [Registratie*InschrijfId] LINKTO INTERFACE Inschrijving
90 , "Deeltoegang" : registratie [Registratie*InschrijfId];deelToegang [InschrijfId*Ja_of_Nee] cRud
91 , "Persoon" : registratie [Registratie*InschrijfId];inschrijving [Persoon*InschrijfId]~ LINKTO INTERFACE Persoon
92 , "BIG-nummer" : registratie [Registratie*InschrijfId];bignummer [InschrijfId*Bignummer] cRud
93 , "Specialisme" : specialist [Registratie * SpecialismeArts] cRUd
94 , "Aantekening" : aantekening [Registratie*Aantekening] CRUD
95 ]
96
97 ENDCONTEXT

```

Listing 12: ArtsRegister

## G. Prototype

The screenshot shows a web application interface with a navigation bar at the top containing icons for 'Apotheker', 'Artsen', 'Beheer', 'Fysiotherapeut', 'Gezondheidszorgpsycholoog', 'Inschrijvingen', and 'Klinisch\_technoloog'. Below the navigation bar, the title 'Persoon' is displayed. The main content area shows a list of two people. Each person's record is a horizontal row with several fields: 'Naam' (Name), 'Voorna(am)(en)' (First name(s)), 'Geslacht' (Gender), 'Adres' (Address), 'Geboortedatum' (Date of birth), 'Nationaliteit' (Nationality), and 'Inschrijving' (Registration). The first person is 'Pietesen', 'Piet', gender 'X', address 'adres3', birth date '13-12-1970', nationality 'Belgische', and registration '1003'. The second person is 'Jansen', 'Jan', gender 'V', address 'adres2', birth date '01-01-1999', nationality 'Nederlandse', and registration '1002'. Each record has a red minus sign and a red square icon with a white 'X' next to the address field, and a green plus sign at the end of the row. Below each record is an 'Add Adres' button.

Figure 20: Personen

The screenshot shows a web application interface with a navigation bar at the top containing icons for 'Apotheker', 'Artsen', 'Beheer', 'Fysiotherapeut', 'Gezondheidszorgpsycholoog', 'Inschrijvingen', and 'Klinisch\_technoloog'. Below the navigation bar, the title 'Persoon' is displayed. The main content area shows a detailed view of a person. The fields are: 'Persoon' (ID: Persoon\_cf6add3f-4d89-4e1d-9e48-cbea2eb8c2b6), 'Naam' (Name: hendrikse), 'Voorna(am)(en)' (First name(s): hendrik), 'Geslacht' (Gender: Code M, Omschrijving Man), 'Adres' (Address: adres1), 'Geboortedatum' (Date of birth: 18-05-2001), 'Nationaliteit' (Nationality: Belgische), and 'Inschrijving' (Registration: Inschrijftid\_454db7fc-9024-4fa2-a013-983ac91d3f04). Each field has a corresponding input field or dropdown menu. There is a red minus sign and a red square icon with a white 'X' next to the address field.

Figure 21: Persoon

Home | Apotheeker | Artsen | Beheer | Fysiotherapeut | Gezondheidszorgpsycholoog | Inschrijvingen | Klinisch\_technoloog

### Inschrijving

Inschrijving | **Diploma** | Weigering | Doorhaling

#### Inschrijving

Inschrijving	Persoon	Registerid	Big-nummer	Inschrijvingstijd	Deeltoegang	Registratie
I003	<b>Persoon Naam</b> P003 Pietersen	arts	B003	2022-02-20T15:48:30+0100	Nee	Registratie_b0f161ce-2d98-4944-8b33-52188c5335bd

Figure 22: Inschrijving

Home | Apotheeker | Artsen | Beheer | Fysiotherapeut | Gezondheidszorgpsycholoog | Inschrijvingen | Klinisch\_technoloog

### Inschrijvingen

Inschrijving | **Diploma** | Weigering

Datums | Diplomadatum | verklaringdatum | getuigschriftdatum | kwalificatiedatum

Datums	Diplomadatum	verklaringdatum	getuigschriftdatum	kwalificatiedatum
I002	24-04-2022 dd-mm-yyyy	26-12-2021 dd-mm-yyyy	dd-mm-yyyy	26-12-2021 24-04-2022

Inschrijving | **Diploma** | Weigering

**Weigering Weigeringsgrond**

I003	Onder curatele gesteld Add Weigering
------	---

Inschrijving | **Diploma** | Weigering

#### Inschrijving

Inschrijving	Persoon	Registerid	Big-nummer	Inschrijvingstijd	Deeltoegang	Registratie
Inschrijfid_454db7fc-9024-4fa2-a013-983ac91d3f04	<b>Persoon Naam</b> Persoon_cf6add3f-4d89-4e1d-9e48-cbea2eb8c2b6 hendrikse	fysiotherapeut	Bignummer_39bfc16-356b-41bd-92b5-a6da12ddd066	2022-02-22T16:21:31+0100	Nee	Registratie_645c0897-7f38-4c06-aade-1bb743e77f5b

Figure 23: Inschrijving-diploma-en-weigergrond

Register	Persoon	Naam	Bignummer	Registratie
arts	P001	Edelaar Pietersen	B001	R001
	P003		B003	Registratie_b0f161ce-2d98-4944-8b33-52188c5335bd
orthopedagoog-generalist				
klinisch technoloog				
tandarts	P002	Jansen	B002	Registratie_c4423b39-f627-4392-900b-6e9d481556af
apotheker				
gezondheidszorgpsycholoog				
psychotherapeut				
fysiotherapeut	Persoon_cf6add3f-4d89-4e1d-9e48-cbea2eb8c2b6	hendriks e	Bignummer_39bfc16-356b-41bd-92b5-a6da12ddd66	Registratie_645c0897-7f38-4c06-aade-1bb743e77f5b
verloskundige				
verpleegkundige				
physician assistant				

Figure 24: Registratie

Persoon	Naam	Doorhalingreden
P003	Pietersen	Doorhalingreden
		I003 <input type="text" value="Add Doorhaling"/>
P002	Jansen	Doorhalingreden
		I002 <input type="text" value="Add Doorhaling"/>
P001	Edelaar	Doorhalingreden
		I001 <input type="text" value="Add Doorhaling"/>
Persoon_cf6add3f-4d89-4e1d-9e48-cbea2eb8c2b6	hendrikse	Doorhalingreden
		Inschrijfid_454db7fc-9024-4fa2-a013-983ac91d3f04 <input type="text" value="Add Doorhaling"/>
Persoon_eeb4c821-ddd1-4ca2-b78f-497825762ca4	jjj	Doorhalingreden
		Inschrijfid_50fc3e4a-7aa5-41ed-8277-d19cf7c153ed <input type="text" value="Add Doorhaling"/>

Figure 25: Registratie doorhaling

[Home](#)
[Apotheker](#)
[Artsen](#)
[Beheer](#)
[Fysiotherapeut](#)
[Gezondheidszorgpsycholoog](#)
[Inschrijvingen](#)
[Klinisch\\_technoloog](#)

### Registratie

[Register](#)
[Weigering](#)
[Doorhaling](#)

Persoon	Naam	Weigeringsgrond
P003	Pietersen	<b>Weigeringsgrond</b> I003 <b>Onder curatele gesteld</b> <input type="button" value="Add Weigering"/>
P002	Jansen	<b>Weigeringsgrond</b> I002 <input type="button" value="Add Weigering"/>
P001	Edelaar	<b>Weigeringsgrond</b> I001 <input type="button" value="Add Weigering"/>
Persoon_cf6add3f-4d89-4e1d-9e48-cbea2eb8c2b6	hendrikse	<b>Weigeringsgrond</b> Inschrijfid_454db7fc-9024-4fa2-a013-983ac91d3f04 <input type="button" value="Add Weigering"/>
Persoon_eeb4c821-ddd1-4ca2-b78f-497825762ca4	jjj	<b>Weigeringsgrond</b> Inschrijfid_50fc3e4a-7aa5-41ed-8277-d19c7c153ed <input type="button" value="Add Weigering"/>

Figure 26: Registratie weigering

[Home](#)
[Apotheker](#)
[Artsen](#)
[Beheer](#)
[Fysiotherapeut](#)
[Gezondheidszorgpsycholoog](#)
[Inschrijvingen](#)
[Klinisch\\_technoloog](#)

### Beheer

[Geslacht](#)
[Nationaliteit](#)
[Register](#)

#### Geslacht

Geslacht	Omschrijving
M	Man
V	Vrouw
X	Onbekend

Figure 27: Beheergeslacht

Beheer

Geslacht Nationaliteit Register

### Nationaliteit

Nationaliteit	Ingangsdatum	Einddatum	
Nederlandse	01-01-1975	dd-mm-yyyy	+
Belgische	01-01-1965	dd-mm-yyyy	
Spaanse	01-01-1985	dd-mm-yyyy	+

Figure 28: Beheernationaliteit

Beheer

Geslacht Nationaliteit Register

### Register

Register	Ingangsdatum	Einddatum	
arts	01-01-2000	dd-mm-yyyy	
orthopedagoog-generalist	01-01-2000	dd-mm-yyyy	
klinisch technoloog	01-01-2000	dd-mm-yyyy	
tandarts	01-01-2000	dd-mm-yyyy	
apotheker	01-01-2000	dd-mm-yyyy	
gezondheidszorgpsycholoog	01-01-2000	dd-mm-yyyy	
psychotherapeut	01-01-2000	dd-mm-yyyy	
fysiotherapeut	01-01-2000	dd-mm-yyyy	
verloskundige	01-01-2000	dd-mm-yyyy	
verpleegkundige	01-01-2000	dd-mm-yyyy	
physician assistant	01-01-2000	dd-mm-yyyy	+

Figure 29: Beheerregister



### Artsen

Inschrijvingen	Persoon	Naam	BIG-nummer	Registratie
I001	P001	Edelaar	B001	R001
I003	P003	Pietersen	B003	Registratie_b0f161ce-2d98-4944-8b33-52188c5335bd

Figure 30: Artsregister



## **I. Conceptual analysis**

# Conceptuele Analyse van 'BIG

Gerard Edelaar

5 januari 2022

## Table of Contents

1	Conceptuele Analyse.....	2
1.1	Persoon.....	2
1.1.1	Persoon.....	3
1.2	Inschrijving.....	3
1.2.1	Inschrijfd.....	4
1.3	Register.....	5
1.4	Arts.....	6
1.5	Tandarts.....	7
1.6	Apotheker.....	7
1.7	Gezondheidszorgpsycholoog.....	8
1.8	Psychotherapeut.....	9
1.9	Fysiotherapeut.....	10
1.10	Verloskundige.....	10
1.11	Verpleegkundige.....	11
1.12	Physician_assistant.....	12
1.13	Orthopedagoog_generalist.....	12
1.14	Klinisch_technoloog.....	13
1.15	Inschrijfduur.....	14
1.16	Registratie.....	15
1.17	Weigering.....	16
1.18	Aantekening.....	17
1.19	Geslacht.....	19
1.20	Nationaliteit.....	20
1.20.1	Nationaliteitid.....	20
1.21	Adres.....	20
1.22	Specialisme.....	21
1.23	Overig.....	22

# 1 Conceptuele Analyse

Dit hoofdstuk analyseert de "taal van de business", om functionele eisen ten behoeve van 'BIG' te kunnen bespreken. Deze analyse beoogt om een bouwbare, maar oplossingsonafhankelijke specificatie op te leveren. Het begrijpen van tekst vereist deskundigheid op het gebied van conceptueel modelleren.

## 1.1 Persoon

In artikel 3 lid 2 wordt aangegeven dat bij elke inschrijving in het register de naam, voornamen, geslacht, geboortedatum, nationaliteit en adres van de betrokkene en het nummer en het tijdstip van inschrijving wordt vermeld. Bij ministeriële regeling kunnen gegevens worden aangewezen die ten behoeve van het identificeren van beroepsbeoefenaren bij de inschrijving worden vermeld. Deze beroepsbeoefenaren zijn personen.

### **Definitie Persoon:**

Een Persoon representeert een persoons-id dat opgenomen is in het BIG-register.

In artikel 3 lid 2 is aangegeven dat de naam een onderdeel is van de identificatie van de zorgverlener.

### **Definitie Naam:**

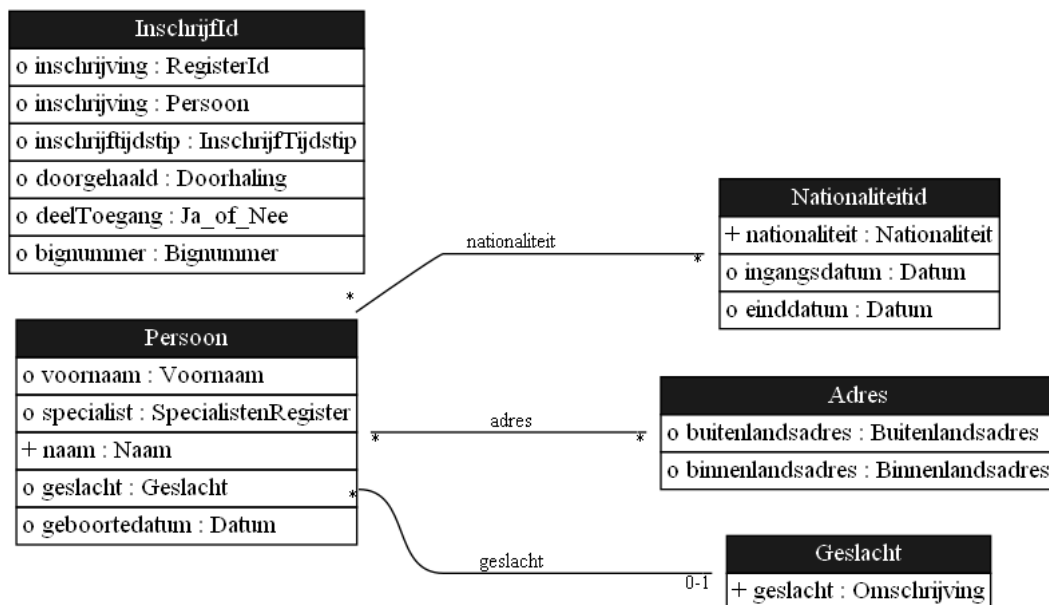
Aanduiding van de familienaam zoals vastgelegd in de BRP.

In artikel 3 lid 2 is aangegeven dat de voorna(a)m(en) een onderdeel is/zijn van de identificatie van de zorgverlener.

### **Definitie Voornaam:**

Alle voornamen van de Persoon zoals dit is vastgelegd binnen de BRP.

1.1 Conceptueel diagram van 'Persoon'.



Figuur 1.1: Conceptueel diagram van Persoon

### 1.1.1 Persoon

Attribuut	Betekenis
voornaam	Elke ingeschrevene moet een voornaam hebben.
naam	Elke ingeschrevene moet een naam hebben en een naam kan bij meerdere personen behoren.
geslacht	Elke ingeschrevene behoort tot een geslacht
geboortedatum	Elke ingeschrevene heeft een geboortedatum

Relatie	Betekenis
adres [Persoon*Adres]	Elke ingeschrevene heeft een adres
nationaliteit [Persoon*Nationaliteitid]	Elke ingeschrevene heeft een nationaliteit

## 1.2 Inschrijving

Inschrijving legt de relatie vast tussen Persoon en het Register.

In artikel 3 lid 2 wordt aangegeven dat bij elke inschrijving worden in het register vermeld de naam, voornamen, geslacht, geboortedatum, nationaliteit en adres van de betrokkene en het nummer en het tijdstip van inschrijving. Bij ministeriële regeling kunnen gegevens worden aangewezen die ten behoeve van het identificeren van beroepsbeoefenaren bij de inschrijving worden vermeld.

### Definitie InschrijfId:

De aanmelding van persoon in een register

In artikel 3 lid 2 is aangegeven dat de datum en het tijdstip van inschrijving een onderdeel is van de identificatie van de zorgverlener.

**Definitie InschrijfTijdstip:**

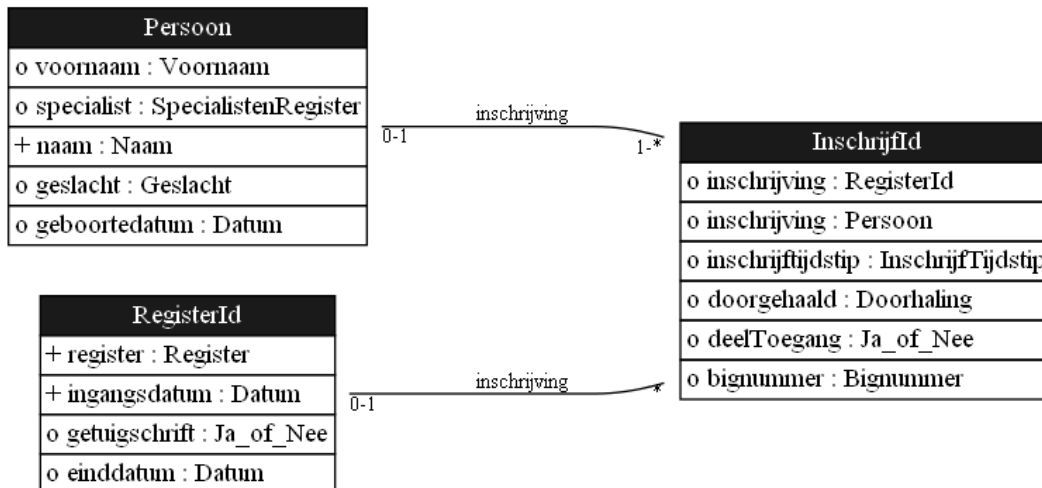
Het inschrijftijdstip bevat de datum en tijdstip van inschrijving in Y-m-d h:i:s-formaat.

In artikel 3 lid 2 wordt aangegeven dat bij elke inschrijving in het register de naam, voornamen, geslacht, geboortedatum, nationaliteit en adres van de betrokkene en het nummer en het tijdstip van inschrijving wordt vermeld. Bij ministeriële regeling kunnen gegevens worden aangewezen die ten behoeve van het identificeren van beroepsbeoefenaren bij de inschrijving worden vermeld. Het BIG-nummer identificeert de BIG-ingeschrevene.

**Definitie Bignummer:**

Het identificatienummer van de BIG-ingeschrevene.

1.2 Conceptueel diagram van ‘Inschrijving’.



Figuur 1.2: Conceptueel diagram van Inschrijving

1.2.1 InschrijfId

Attribuut	Betekenis
inschrijving	Het vastleggen van de koppeling tussen het register en de inschrijving.
inschrijving	Elk persoon die BIG geregistreerd wil zijn, moet zijn ingeschreven. Een persoon kan meerdere inschrijvingen hebben.
inschrijftijdstip	Elke inschrijving vindt plaats op een tijdstip.

Relatie	Betekenis
bignummer [InschrijfId*Bignummer]	De koppeling tussen een Persoon en een Bignummer. Dit is een één op één koppeling die automatisch wordt aangebracht.

Relatie	Betekenis
deelToegang [InschrijfId*Ja_of_Nee]	In Artikel 3, lid 4 van de Wet Big wordt aangegeven dat indien de inschrijving van een beroepsbeoefenaar plaatsvindt op basis van een gedeeltelijke toegang als bedoeld in artikel 12, eerste lid, van de Algemene wet erkenning EU-beroepskwalificaties, wordt dit bij de inschrijving aangetekend, waarbij wordt vermeld voor welke beroepswerkzaamheden de gedeeltelijke toegang geldt en onder welke beroepstitel die beroepsbeoefenaar zijn beroepswerkzaamheden op grond van artikel 12, derde lid, uitvoert.

### 1.3 Register

Het BIG-register (Beroepen in de Individuele Gezondheidszorg) is een wettelijk, online en openbaar register. Alleen wie in het BIG-register staat, mag een beschermde beroepstitel voeren en mag de bij het beroep horende voorbehouden handelingen zelfstandig uitvoeren. Iedereen kan het register raadplegen. Het BIG-register verzorgt ook de erkenning van buitenlandse diploma's.

#### **Definitie RegisterId:**

Technisch element voor het register.

In artikel 1 lid 5 wordt aangegeven dat elk Register wordt ingesteld en beheerd door Onze Minister. In artikel 3 lid 6 wordt gesteld dat de registers worden ingesteld ten einde te kunnen voldoen aan een verzoek om informatie als bedoeld in artikel 12 en ten behoeve van het toezicht op de uitvoering van de artikelen 4 en 17.

#### **Definitie Register:**

Een register is een officiële lijst van personen die aan de, door het register gestelde voorwaarden voldoen.

1.3 Conceptueel diagram van 'Register'.



*Figuur 1.3: Conceptueel diagram van Register*

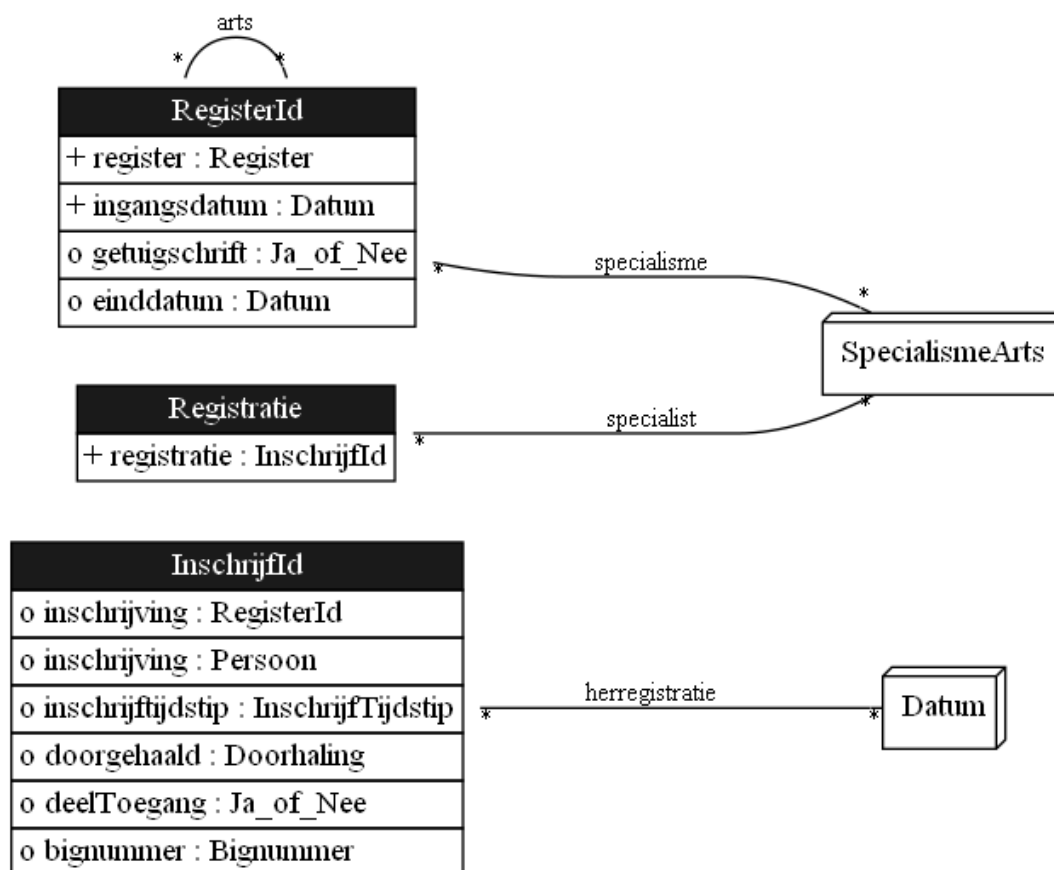


Relatie	Betekenis
einddatum [RegisterId*Datum]	
getuigschrift [RegisterId*Ja_of_Nee]	
ingangsdatum [RegisterId*Datum]	
register [RegisterId*Register]	

## 1.4 Arts

Het register voor arts bevat alle attributen van het register arts.

1.4 Conceptueel diagram van 'Arts'.



Figuur 1.4: Conceptueel diagram van Arts

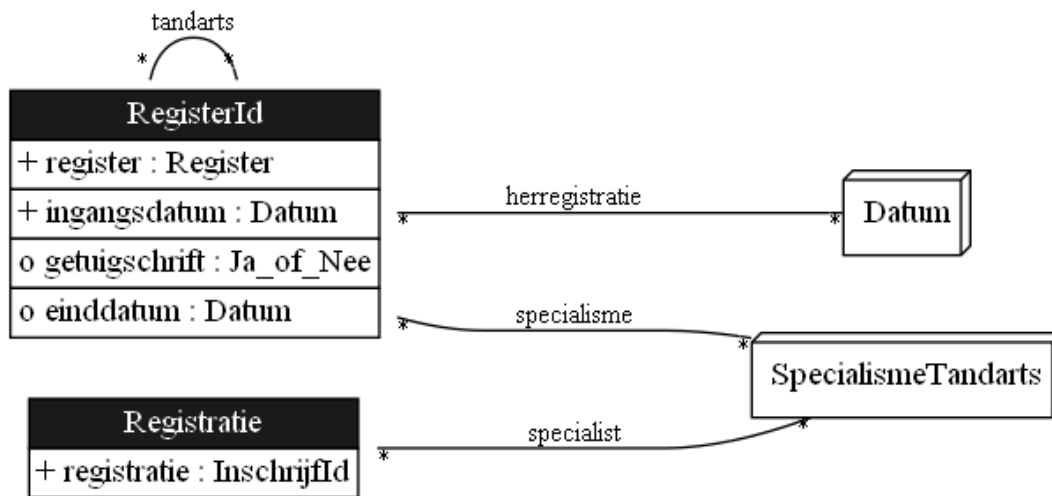
Relatie	Betekenis
arts [RegisterId*RegisterId]	Het identificerende nummer van de het artsenregister.
herregistratie [InschrijfId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
specialisme	Alle specialismes voor arts.

Relatie	Betekenis
[RegisterId*SpecialismeArts]	
specialist	De specialist is een uitbreiding van arts met een specialisme.
[Registratie*SpecialismeArts]	

## 1.5 Tandarts

Het register voor tandarts bevat alle attributen van het register tandarts.

1.5 Conceptueel diagram van 'Tandarts'.



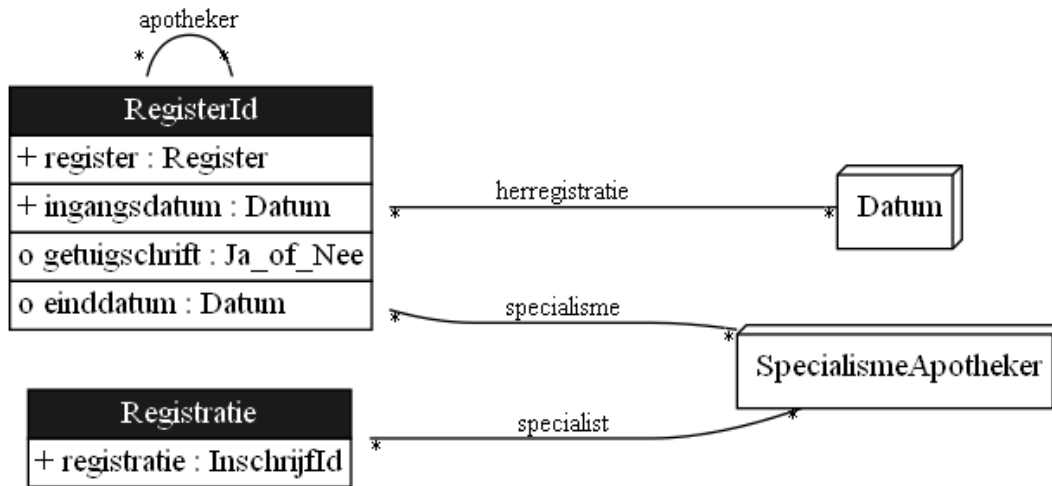
Figuur 1.5: Conceptueel diagram van Tandarts

Relatie	Betekenis
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
specialisme [RegisterId*SpecialismeTandarts]	Alle specialismes voor tandarts.
specialist [Registratie*SpecialismeTandarts]	De specialist is een uitbreiding van tandarts met een specialisme.
tandarts [RegisterId*RegisterId]	

## 1.6 Apotheeker

Het register voor apotheker bevat alle attributen van het register apotheker.

1.6 Conceptueel diagram van 'Apotheeker'.



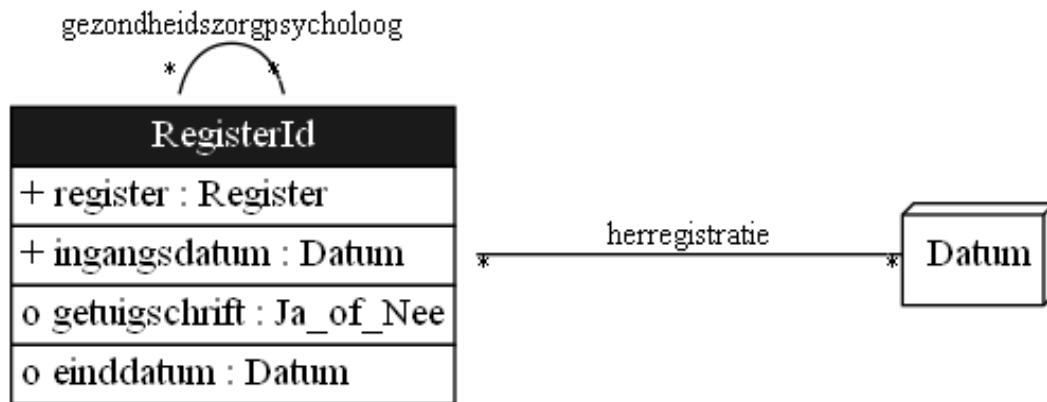
Figuur 1.6: Conceptueel diagram van Apotheker

Relatie	Betekenis
apotheker [RegisterId*RegisterId]	
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
specialisme [RegisterId*SpecialismeApotheker]	Alle specialismes voor Apotheker.
specialist [Registratie*SpecialismeApotheker]	De specialist is een uitbreiding van tandarts met een specialisme.

## 1.7 Gezondheidszorgpsycholoog

Het register voor gezondheidszorgpsycholoog bevat alle attributen van het register gezondheidszorgpsycholoog.

1.7 Conceptueel diagram van 'Gezondheidszorgpsycholoog'.



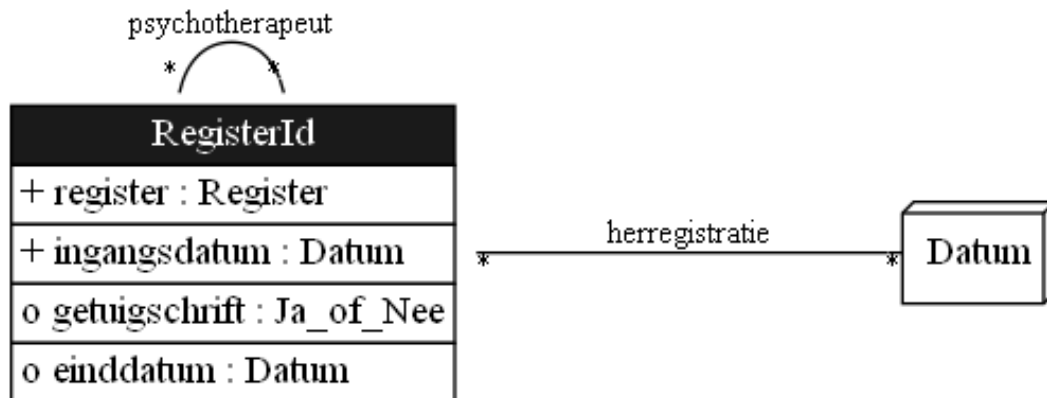
Figuur 1.7: Conceptueel diagram van Gezondheidszorgpsycholoog

Relatie	Betekenis
gezondheidszorgpsycholoog [RegisterId*RegisterId]	
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.

## 1.8 Psychotherapeut

Het register voor psychotherapeut bevat alle attributen van het register psychotherapeut.

1.8 Conceptueel diagram van 'Psychotherapeut'.



Figuur 1.8: Conceptueel diagram van Psychotherapeut

Relatie	Betekenis
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
psychotherapeut	

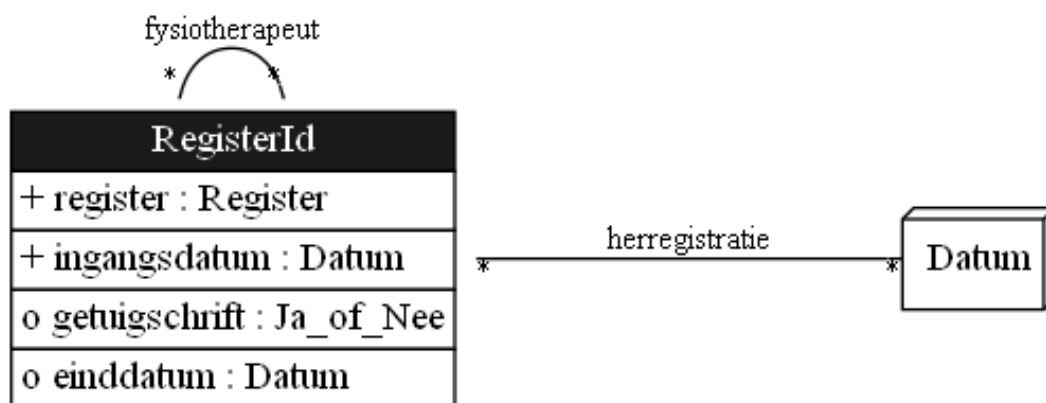
Relatie	Betekenis
---------	-----------

[RegisterId*RegisterId]	
-------------------------	--

## 1.9 Fysiotherapeut

Het register voor fysiotherapeut bevat alle attributen van het register fysiotherapeut.

1.9 Conceptueel diagram van 'Fysiotherapeut'.



*Figuur 1.9: Conceptueel diagram van Fysiotherapeut*

Relatie	Betekenis
---------	-----------

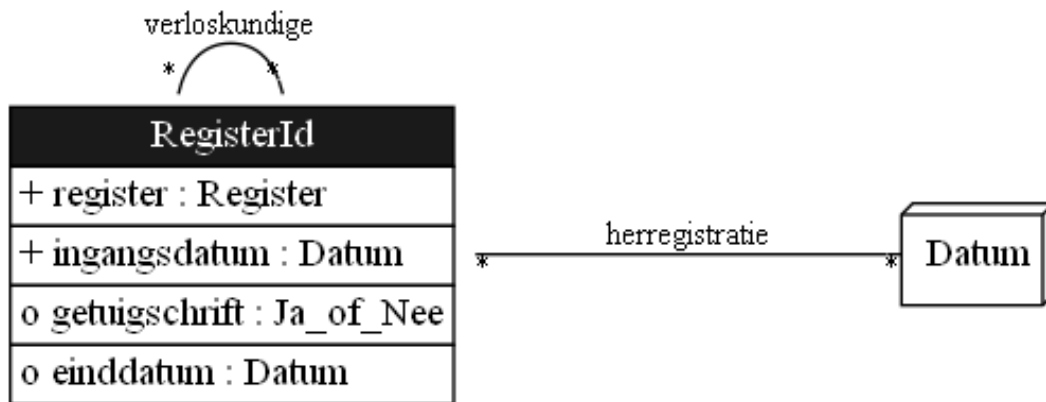
fysiotherapeut [RegisterId*RegisterId]	
---	--

herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
--------------------------------------	--

## 1.10 Verloskundige

Het register voor verloskundige bevat alle attributen van het register verloskundige.

1.10 Conceptueel diagram van 'Verloskundige'.



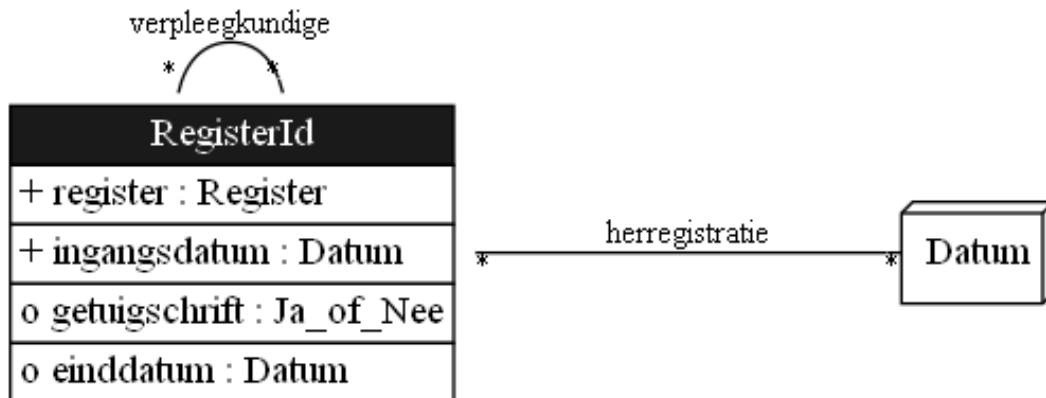
*Figuur 1.10: Conceptueel diagram van Verloeskundige*

Relatie	Betekenis
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
verloeskundige [RegisterId*RegisterId]	

### 1.11 Verpleegkundige

Het register voor verpleegkundige bevat alle attributen van het register verpleegkundige.

1.11 Conceptueel diagram van 'Verpleegkundige'.



*Figuur 1.11: Conceptueel diagram van Verpleegkundige*

Relatie	Betekenis
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
verpleegkundige	

Relatie

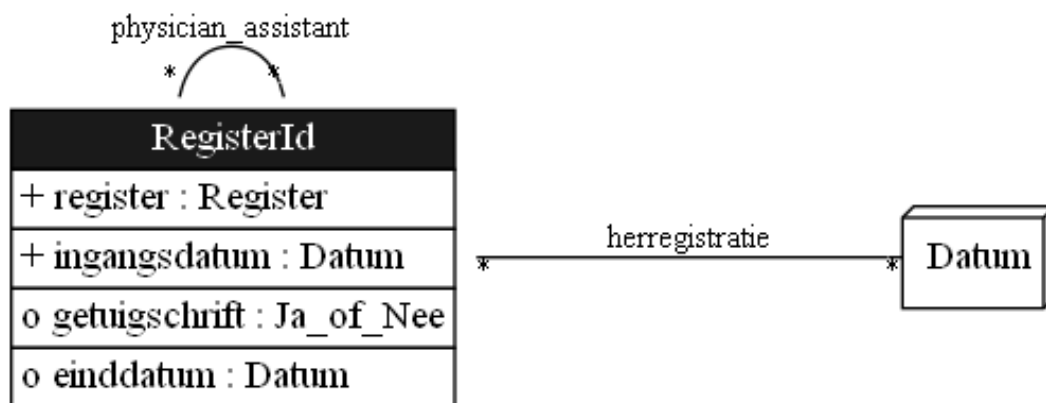
Betekenis

[RegisterId\*RegisterId]

### 1.12 Physician\_assistant

Het register voor physician\_assistant bevat alle attributen van het register physician\_assistant.

1.12 Conceptueel diagram van 'Physician\_assistant'.



Figuur 1.12: Conceptueel diagram van Physician\_assistant

Relatie

Betekenis

herregistratie  
[RegisterId\*Datum]

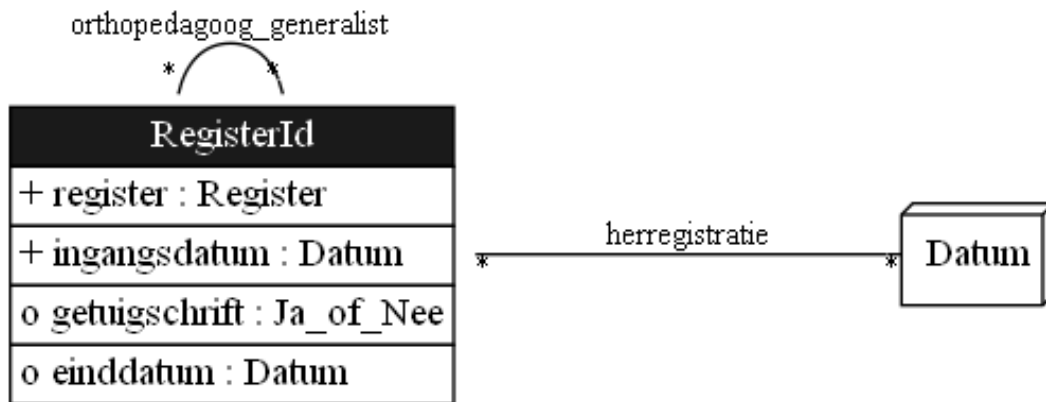
Artikel 2, tweede lid van het Besluit periodieke registratie  
Wet Big stelt dat de datum van herregistratie op vijf jaar na  
datum van registratie.

physician\_assistant  
[RegisterId\*RegisterId]

### 1.13 Orthopedagoog\_generalist

Het register voor orthopedagoog\_generalist bevat alle attributen van het register orthopedagoog\_generalist.

1.13 Conceptueel diagram van 'Orthopedagoog\_generalist'.



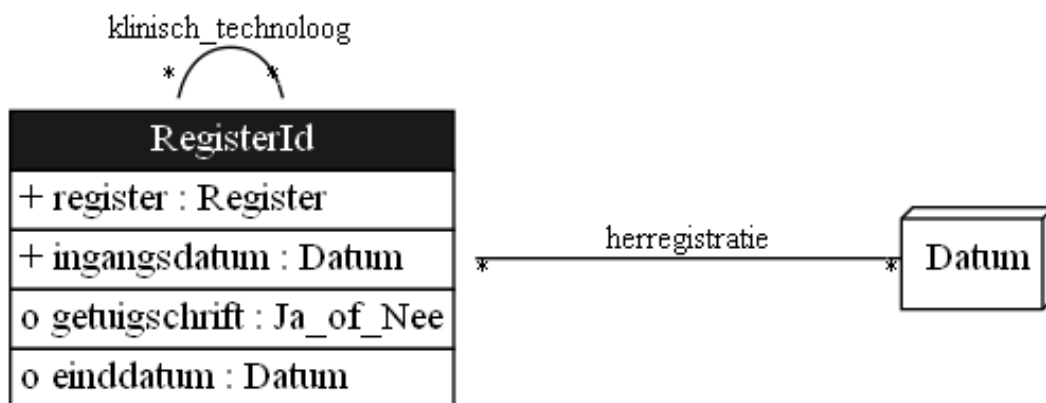
*Figuur 1.13: Conceptueel diagram van Orthopedagoog\_generalist*

Relatie	Betekenis
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.
orthopedagoog_generalist [RegisterId*RegisterId]	

### 1.14 Klinisch\_technoloog

Het register voor klinisch\_technoloog bevat alle attributen van het register klinisch\_technoloog.

1.14 Conceptueel diagram van 'Klinisch\_technoloog'.



*Figuur 1.14: Conceptueel diagram van Klinisch\_technoloog*

Relatie	Betekenis
herregistratie [RegisterId*Datum]	Artikel 2, tweede lid van het Besluit periodieke registratie Wet Big stelt dat de datum van herregistratie op vijf jaar na datum van registratie.



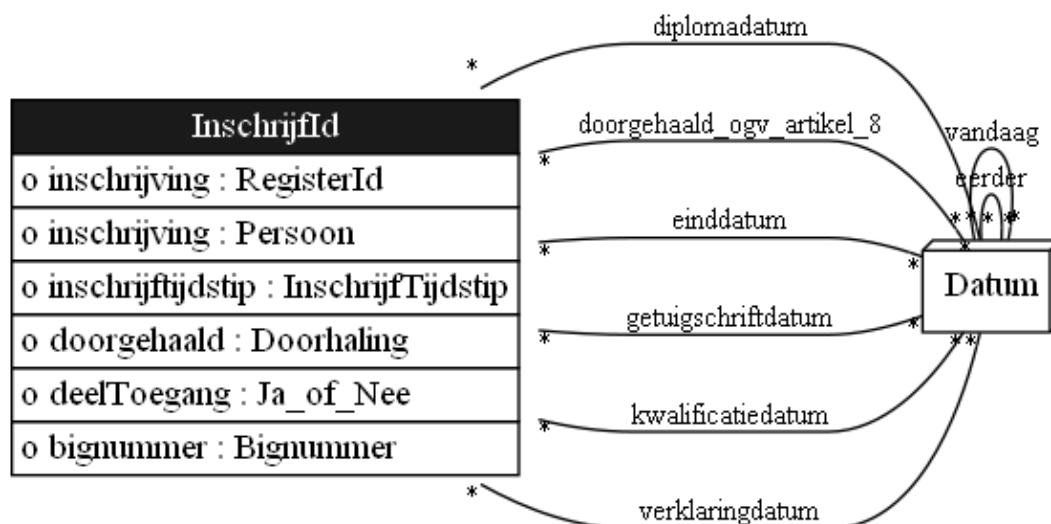
Relatie Betekenis

klinisch\_technoloog  
[RegisterId\*RegisterId]

### 1.15 Inschrijfduur

Inschrijvingen in een register zijn beperkt geldig. Artikel 8, eerste lid van de Wet BIG stelt: "Bij algemene maatregel van bestuur wordt bepaald dat de inschrijving in een bij de maatregel aangewezen register wordt doorgehaald indien na de in het tweede lid bedoelde datum een bij de maatregel aangegeven periode is verstreken." Artikel 2, tweede lid van het Besluit periodieke registratie Wet BIG stelt deze periode op vijf jaren.

1.15 Conceptueel diagram van 'Inschrijfduur'.



Figuur 1.15: Conceptueel diagram van Inschrijfduur

Relatie Betekenis

diplomadatum  
[Inschrijfd\*Datum]

Er is een erkend diploma bij een inschrijving geregistreerd dat relevant is voor het bepalen van de geldigheid van die inschrijving.  
Een register registreert de datum waarop de ingeschrevene een diploma heeft behaald op grond waarvan de ingeschrevene een erkenning van beroepskwalificaties als bedoeld in de Algemene wet erkenning EU-beroepskwalificaties heeft verkregen, zoals bedoeld in Art. 8 lid 2 sub a van de Wet BIG. Als er meerdere diploma's zijn, kunnen er dus ook meerdere diplomadata zijn voor dezelfde inschrijving.

doorgehaald\_ovg\_artikel\_8  
[Inschrijfd\*Datum]

Om het einde van een inschrijving te bepalen tellen we vijf jaar (zie Artikel 2.2 van het besluit) op bij de meest recente kwalificatiedatum. Als gevolg daarvan verandert de

Relatie	Betekenis
eerder [Datum*Datum]	einddatum als de ingeschrevene tijdig verlenging krijgt. Voor elke denkbare datum d1 en d2 geldt d1 eerder d2 dan en slechts dan als $d1 < d2$ .
einddatum [Inschrijfd*Datum]	Om het einde van een inschrijving te bepalen tellen we vijf jaar (zie Artikel 2.2 van het besluit) op bij de meest recente kwalificatiedatum. Als gevolg daarvan verandert de einddatum als de ingeschrevene tijdig verlenging krijgt.
getuigschriftdatum [Inschrijfd*Datum]	Er is een getuigschrift bij een inschrijving geregistreerd dat relevant is voor het bepalen van de geldigheid van die inschrijving. Een register registreert datum waarop de ingeschrevene een bij of krachtens hoofdstuk III of VI aangewezen getuigschrift heeft verkregen, zoals bedoeld in Art. 8 lid 2 sub a van de Wet BIG. Als er meerdere getuigschriften zijn, kunnen er dus ook meerdere data zijn bij dezelfde inschrijving.
kwalificatiedatum [Inschrijfd*Datum]	Om de inschrijvingsduur te bepalen rekenen we met de meest recente van de geregistreerde diplomadata, verklaringdata en getuigschriftdata. Uiteindelijk is er dus precies één datum die gebruikt wordt om de inschrijfduur te bepalen.
vandaag [Datum*Datum]	
verklaringdatum [Inschrijfd*Datum]	Er is een verklaring bij een inschrijving geregistreerd die relevant is voor het bepalen van de geldigheid van die inschrijving. Een register registreert de datum waarop de ingeschrevene een in artikel 41, eerste lid, onder b, bedoelde verklaring heeft verkregen, zoals bedoeld in Art. 8 lid 2 sub a van de Wet BIG. Als er meerdere van dit soort verklaringen zijn, kunnen er dus ook meerdere data zijn bij dezelfde inschrijving.

## 1.16 Registratie

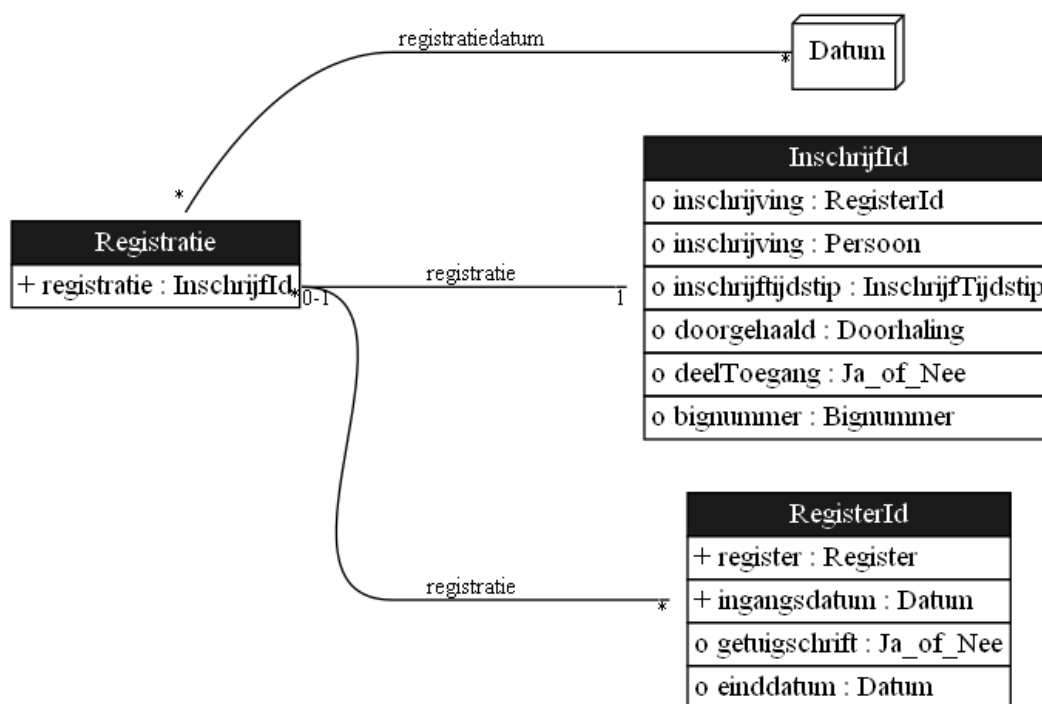
*Een registratie is de inschrijving in een, door de Minister vastgesteld, zorgregister van een persoon.*

Er is sprake van registratie van een ingeschrevene wanneer het inschrijvingsproces geheel afgerond is en aan alle voorwaarden is voldaan.

### **Definitie Registratie:**

Betreft een complete afgeronde inschrijving

1.16 Conceptueel diagram van 'Registratie'.



Figuur 1.16: Conceptueel diagram van Registratie

Relatie	Betekenis
registratie [Registratie*InschrijfId]	
registratie [Registratie*RegisterId]	
registratiedatum [Registratie*Datum]	

### 1.17 Weigering

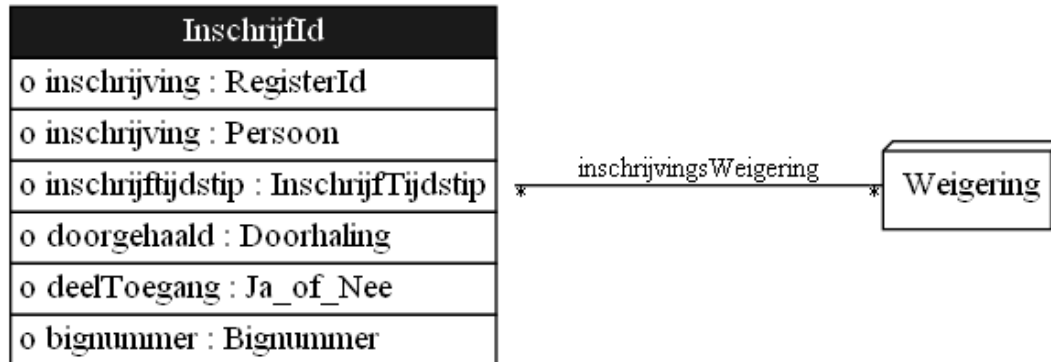
De inschrijving wordt geweigerd:

1. indien de aanvrager niet voldoet aan de in hoofdstuk III bedoelde opleidingseisen;
2. indien de aanvrager ingevolge in kracht van gewijsde gegane rechterlijke uitspraak onder curatele is gesteld wegens lichamelijke of geestelijke toestand;
3. indien de aanvrager ingevolge rechterlijke uitspraak ontzet is van het recht het betrokken beroep uit te oefenen;
4. indien zulks voortvloeit uit een op grond van deze wet jegens de aanvrager genomen maatregel;
5. indien ten aanzien van de aanvrager een maatregel, berustende op een in het buitenland gegeven rechterlijke, tuchtrechtelijke of bestuursrechtelijke beslissing, van kracht is op grond waarvan de aanvrager zijn rechten ter zake van de uitoefening van het betrokken beroep in het land waar de beslissing is gegeven tijdelijk of blijvend geheel heeft verloren,
6. indien de aanvrager de Nederlandse taal niet voldoende beheerst om zijn beroep in Nederland uit te kunnen oefenen.

### Definitie Weigering:

art.6-redenen voor afwijzing inschrijving.

#### 1.17 Conceptueel diagram van 'Weigering'.



Figuur 1.17: Conceptueel diagram van Weigering

Relatie	Betekenis
inschrijvingsWeigering [InschrijfId*Weigering]	In artikel 6 staan de redenen voor het weigeren van de inschrijving bepaald.

#### 1.18 Aantekening

Artikel 9 1. In het register wordt, indien zulks voortvloeit uit een op grond van deze wet genomen maatregel of besluit, een aantekening geplaatst van: a. een opgelegde berisping indien dit op grond van artikel 48, elfde lid, door het regionale tuchtcollege of het centraal tuchtcollege is beslist; b. een opgelegde geldboete indien dit op grond van artikel 48, elfde lid, door het regionale tuchtcollege of het centraal tuchtcollege is beslist; c. de schorsing van de bevoegdheid, bedoeld in artikel 48, eerste lid, onder d; d. de voorwaarden die zijn opgelegd; e. de gedeeltelijke ontzegging van de bevoegdheid het betrokken beroep uit te oefenen; f. de doorhaling van de inschrijving in het register op grond van artikel 7, onder c, d of e; g. de ontzegging van het recht wederom in het register te worden ingeschreven; h. het eindigen van een schorsing, anders dan ten gevolge van het verstrijken van de in een maatregel vastgestelde tijdsduur; i. het niet langer gelden van de onder e bedoelde voorwaarden, anders dan ten gevolge van het verstrijken van de proeftijd, en van de onder f bedoelde ontzegging; j. de bevoegdheid van een krachtens artikel 5 aangewezen beroepsbeoefenaar om de krachtens artikel 36, veertiende lid, aangewezen UR-geneesmiddelen voor te schrijven, onder vermelding van de categorie van beroepsbeoefenaren waartoe de betrokken beroepsbeoefenaar behoort; k. de op grond van artikel 48, tweede lid, opgelegde beperkingen met betrekking tot het beroepsmatig handelen op het gebied van de individuele gezondheidszorg; l. de beslissing als bedoeld in artikel 48a, tweede lid, tot de tenuitvoerlegging van een voorwaardelijke maatregel; m. de last tot onmiddellijke onthouding van de beroepsactiviteiten, bedoeld in artikel 85a. 2. In het register wordt ten aanzien van een geregistreerd of voormalig geregistreerd beroepsbeoefenaar een aantekening geplaatst van: a. een in het buitenland gegeven

rechterlijke, tuchtrechtelijke of bestuursrechtelijke beslissing op grond waarvan de beroepsbeoefenaar zijn rechten ter zake van de uitoefening van het recht het betrokken beroep uit te oefenen in het land waar de beslissing is gegeven tijdelijk of blijvend geheel of gedeeltelijk heeft verloren. Indien die rechterlijke uitspraak tevens inhoudt een beperking in het recht om andere beroepen in de individuele gezondheidszorg uit te oefenen, wordt die beperking eveneens aangetekend. b. een op grond van de Wet medisch tuchtrecht BES gegeven tuchtrechtelijke beslissing op grond waarvan de beroepsbeoefenaar zijn rechten ter zake van de uitoefening van het betrokken beroep op Bonaire, St. Eustatius en Saba tijdelijk of blijvend geheel of gedeeltelijk dan wel voorwaardelijk heeft verloren. Indien die tuchtrechtelijk beslissing tevens inhoudt een beperking in het recht om andere beroepen in de individuele gezondheidszorg uit te oefenen, wordt die beperking eveneens aangetekend.

3. In het register wordt een aantekening geplaatst van een aan de beroepsbeoefenaar op grond van de Wet kwaliteit, klachten en geschillen zorg gegeven bevel of aanwijzing, indien dat bevel of die aanwijzing inhoudt dat aan de betrokkene een beperking is opgelegd in de uitoefening van het betrokken beroep.

4. In het register wordt ten aanzien van een geregistreerd of voormalig geregistreerd beroepsbeoefenaar een aantekening geplaatst van: a. rechterlijke uitspraken inhoudende de ontzetting van of beperking op het recht het betrokken beroep uit te oefenen. Indien die rechterlijke uitspraak tevens inhoudt een ontzetting van of beperking in het recht om ook andere beroepen in de individuele gezondheidszorg uit te oefenen, wordt die ontzetting of beperking eveneens aangetekend. b. een op grond van artikel 14c, tweede lid, van het Wetboek van Strafrecht gestelde bijzondere voorwaarde waaruit een inperking voortvloeit van de bevoegdheid het betrokken beroep uit te oefenen. Indien die bijzondere voorwaarde tevens inhoudt een beperking van de bevoegdheid om andere beroepen in de individuele gezondheidszorg uit te oefenen, wordt die inperking eveneens aangetekend.

5. Bij een aantekening als bedoeld in het eerste tot en met vierde lid wordt vermeld: a. de datum waarop van de schorsing een aantekening wordt geplaatst alsmede de duur van de schorsing, indien die reeds bekend is; b. de datum waarop de berisping, de geldboete, de in het eerste lid bedoelde voorwaarden, de ontzegging, de doorhaling, de ontzegging van het recht op wederinschrijving, de last tot onmiddellijke onthouding van de beroepsactiviteiten of het bevel of de aanwijzing, bedoeld in het derde lid, zijn gaan gelden alsmede, ingeval de voorwaarden of de in het tweede lid bedoelde maatregel tot een proeftijd zijn beperkt, de duur daarvan dan wel c. de datum waarop de schorsing of de last tot onmiddellijke onthouding van de beroepsactiviteiten is geëindigd of vanaf welke de in eerste lid bedoelde voorwaarden of de in het tweede en derde lid bedoelde maatregelen niet langer gelden.

6. Indien de in het tweede lid bedoelde aantekening in het register is geplaatst, geldt de in het buitenland dan wel de op grond van de Wet medisch tuchtrecht BES opgelegde bevoegdheidsbeperking ook voor de beroepsuitoefening in Nederland.

7. De in het eerste, tweede, derde, vierde en achtste lid bedoelde aantekening wordt gedurende een bij algemene maatregel van bestuur bepaalde termijn in het register vermeld en daarbij wordt indien bekend de aard van het vergrijp vermeld dat tot de aantekening heeft geleid, alsmede een met redenen omklede toelichting op een genomen maatregel als bedoeld in artikel 48, eerste lid, onder b en c.

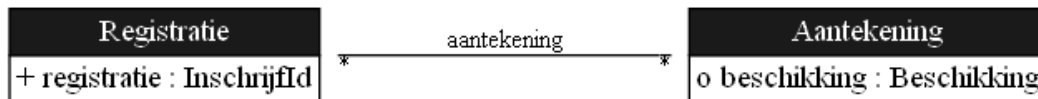
8. In het register wordt voorts een aantekening geplaatst van een maatregel als bedoeld in artikel 7, eerste lid, onderdelen b en c, van de Wet medisch tuchtrecht BES, indien dit op grond van artikel 7, vijfde lid, van de Wet medisch tuchtrecht BES, door het College is beslist.

De aantekening wordt op het register geplaatst bij een beroepsbeoefenaar. De aantekening heeft conform Artikel 9 betrekking op het mogen uitvoeren van de zorgtaak.

**Definitie Aantekening:**

art.7a.2-zie artikel 9

1.18 Conceptueel diagram van 'Aantekening'.



*Figuur 1.18: Conceptueel diagram van Aantekening*

Relatie	Betekenis
aantekening [Registratie*Aantekening]	Een aantekening als bedoeld in artikel 9, tweede lid, onderdeel a van de Wet BIG is bedoeld om een maatregel ten aanzien van een ingeschrevene te registreren.
beschikking [Aantekening*Beschikking]	Een register registreert een beschikking als bedoeld in artikel 10 van de Wet BIG om een aantekening in datzelfde register te onderbouwen.

**1.19 Geslacht**

In Artikel 3 lid 2 is bepaald dat het geslacht van de inschrijver een onderdeel is van de identificatie van de zorgverlener.

**Definitie Geslacht:**

De sekse van een individu.

Nadere duiding van de afkorting die gebruik wordt voor geslacht.

**Definitie Omschrijving:**

Omschrijving van het geslacht van een individu.

1.19 Conceptueel diagram van 'Geslacht'.



*Figuur 1.19: Conceptueel diagram van Geslacht*

Relatie	Betekenis
geslacht [Geslacht*Omschrijving]	

## 1.20 Nationaliteit

Nationaliteit duidt de relatie aan tussen een individu en een staat.

In artikel 3 lid 2 is aangegeven dat de Nationaliteit van de betrokkene bij Inschrijving moet worden vermeld, als onderdeel van de identificatie van de zorgverlener.

### **Definitie Nationaliteitid:**

De Nationaliteit wordt aangeduid middels een 4-cijferige code.

Door een omschrijving toe te voegen wordt de nationaliteitscodering leesbaar.

### **Definitie Nationaliteit:**

De omschrijving van een nationaliteit bevat de tekstuele uitvoering van de nationaliteitscodering.

1.20 Conceptueel diagram van 'Nationaliteit'.



*Figuur 1.20: Conceptueel diagram van Nationaliteit*

### 1.20.1 Nationaliteitid

Attribuut	Betekenis
nationaliteit	Het aanbrengen van de koppeling tussen de nationaliteitcode en de bijbehorende omschrijving. Bij elke code hoort maar één omschrijving en de omschrijving behoort maar tot één code.
ingangsdatum	Ingangsdatum van de Nationaliteit.
einddatum	Einddatum van gebruik van de Nationaliteit.

Relatie    Betekenis

---

## 1.21 Adres

In artikel 3 lid 2 is aangegeven dat het adres een onderdeel is van de identificatie van de zorgverlener.

### **Definitie Adres:**

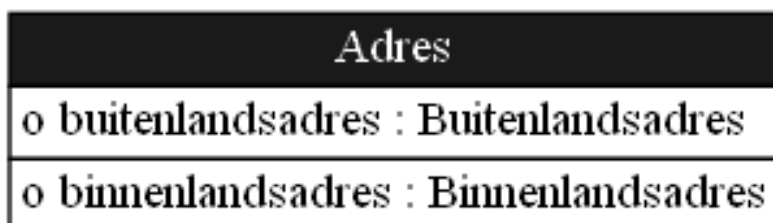
Bevat het adres van de Persoon zoals vastgelegd binnen de BRP.

Adres van de persoon conform BRP.

**Definitie Binnenlandsadres:**

**Definitie Buitenlandsadres:**

1.21 Conceptueel diagram van 'Adres'.



*Figuur 1.21: Conceptueel diagram van Adres*

Relatie	Betekenis
binnenlandsadres [Adres*Binnenlandsadres]	
buitenlandsadres [Adres*Buitenlandsadres]	

## 1.22 Specialisme

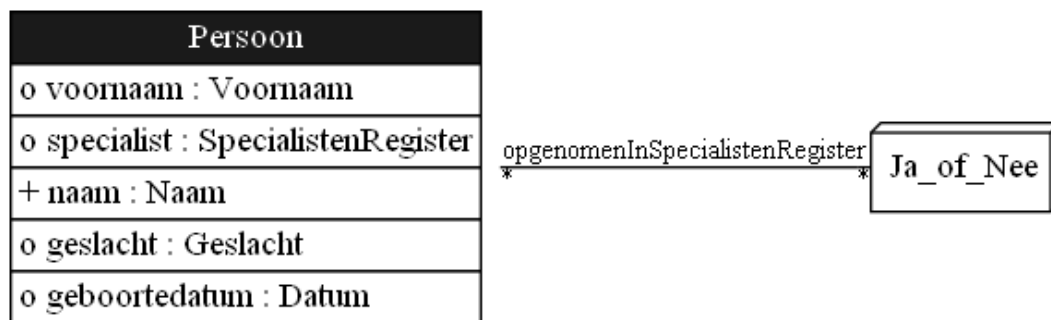
Een regeling als bedoeld in artikel 14, tweede lid, onder d, kan mede inhouden dat degene die de opleiding tot specialist heeft voltooid wordt ingeschreven als specialist voor een bij de regeling bepaalde periode en dat een aansluitende hernieuwde inschrijving slechts plaatsvindt indien de specialist gedurende een bij die regeling bepaald tijdvak, voorafgaand aan de indiening van de aanvraag tot hernieuwde inschrijving, regelmatig op het desbetreffende deelgebied van de beroepsuitoefening werkzaam is geweest dan wel het beroep zal uitoefenen onder de bij de hernieuwde inschrijving aan te geven scholingsvoorwaarden.

**Definitie SpecialistenRegister:**

In artikel 8 lid 3 wordt aangegeven dat een ingeschrevene opgenomen in kan zijn in een specialistenregister.

1.22 Conceptueel diagram van 'Specialisme'.





Figuur 1.22: Conceptueel diagram van Specialisme

Relatie	Betekenis
opgenomenInSpecialistenRegister [Persoon*Ja_of_Nee]	Zoals artikel 14 aangeeft kan een persoon opgenomen zijn in een specialistenregister.
specialist (Attribuut van Persoon)	Een Persoon is specialist wanneer deze in het specialistenRegister is opgenomen.

### 1.23 Overig

#### Definitie Datum:

#### Definitie Ja\_of\_Nee:

description

In Artikel 7 wordt beschreven in welke situatie de inschrijving wordt doorgehaald:

1. in geval van overlijden van de ingeschrevene;
2. op verzoek van de ingeschrevene;
3. indien de ingeschrevene in een der in artikel 6, onder b of c, genoemde omstandigheden is komen te verkeren;
4. indien zulks voortvloeit uit een op grond van deze wet jegens de ingeschrevene genomen maatregel;
5. indien ten aanzien van de ingeschrevene een maatregel, berustende op een in het buitenland gegeven rechterlijke, tuchtrechtelijke of bestuursrechtelijke beslissing van kracht is, op grond waarvan de ingeschrevene zijn rechten ter zake van de uitoefening van het betrokken beroep in het land waar de beslissing is gegeven tijdelijk of blijvend geheel heeft verloren;
6. indien zulks voortvloeit uit een maatregel, berustend op een op grond van de Wet medisch tuchtrecht BES opgelegde maatregel, op grond waarvan de ingeschrevene zijn rechten ter zake van de uitoefening van het betrokken beroep tijdelijk of blijvend geheel heeft verloren.

#### Definitie Doorhaling:

Relatie	Betekenis
doorgehaald [InschrijfId*Doorhaling]	In artikel 7 gestelde situaties waarin de inschrijving wordt doorgehaald. Deze heeft ook consequenties op de bijbehorende registraties.

Relatie

Betekenis

---

ja [Ja\_of\_Nee\*Ja\_of\_Nee]

nee [Ja\_of\_Nee\*Ja\_of\_Nee]

nietVerplicht

[Ja\_of\_Nee\*Ja\_of\_Nee]

sessionToday

[SESSION\*Datum]

verplicht

[Ja\_of\_Nee\*Ja\_of\_Nee]