Human-centred explanation of rule-based decision-making systems in the legal domain

Suzan Zuurmond (4834186)

Faculty of Science, Utrecht University

Master Artificial Intelligence

Supervisors

A. Borg (Project supervisor, First examiner)

M. van Kempen (Daily supervisor)

S. Renooij (Second examiner)

R. Wieten (Second daily supervisor)

Master thesis

Credits: 30 EC

June 5th, 2023

Abstract

This thesis develops a human-centred explanation method for rule-based automated decision-making systems in the legal domain. The research consists of theoretical exploration and practical implementation. Theoretical research establishes a framework for developing explanation methods, representing its key internal components (content, communication and adaptation) and external factors (decision-making system, human recipient and domain). Further investigation of human-centred research highlights the importance of considering both the recipient's knowledge and goals. Besides, we found that one way to accomplish this is by creating a question-driven explanation method and visualising the decision-making process to aid understanding. Accordingly, the proposed explanation method involves representing a decision model in a graph database to be able to both question and visualise it. This proposed explanation method is implemented for a real-world scenario, generating tailored explanations for different target groups. The evaluation highlights the method's ability to answer specific questions but identifies limitations in handling logical checks and hypothetical scenarios. Future research can focus on improving these aspects and exploring additional reasoning properties and customisable interfaces to adapt the method to recipients' evolving needs.

Contents

1 Introduction

In the mid-eighties, governments worldwide started using Automated Decision-making Systems (ADS) to improve the efficiency and effectiveness of public administration (Timmer & Rietveld, 2019). Often, these systems are so-called rule-based systems, with rules based on legislation. These systems have advantages over human workers since they are consistent, cost-efficient and often increase speed. However, they can also affect individuals' rights or legal quality (van Eck, 2018), especially when there is no careful and transparent design process (Lokin, 2018).

As the ongoing digitisation of governance raises concerns globally, there is a pressing need for innovation and digital technologies that improve transparency (Open Government Partnership, 2020). A transparent decision-making process ensures that individuals have access to relevant information about the system and its decision-making processes while also allowing them to understand that information and make sense of the system's decisions (The High-Level Expert Group on Artificial Intelligence, 2019). In the European Union, the rights of individuals regarding automated decision-making are addressed in the General Data Protection Regulation (GDPR). The GDPR provides individuals, to some extent, with a right to explanation by allowing them to obtain "meaningful explanations of the logic involved" in such processes. While the views on the actual scope of these clauses differ (Goodman & Flaxman, 2016), there is a general agreement on the need for implementing such a principle (Wachter et al., 2017).

One way to improve the transparency of automated decision-making systems is by using Model-Driven Engineering (MDE). MDE is an approach to software development that involves creating models of a system's behaviour and using those models to generate code automatically (Schmidt, 2006). Often, this approach uses Domain-Specific Languages (DSL) to create domain-specific models, allowing domain experts to build accurate and understandable models of the system's behaviour without relying on software developers to translate their requirements into code (van Deursen et al., 2000; Völter, 2013). An example of such an approach in the legal domain is the Agile Law Execution Factory (ALEF) (JetBrains, 2019). ALEF is a tool for developing service applications which can perform specific legal tasks like tax calculations. It uses a DSL called Regelspraak to specify rules, data descriptions and test cases, which are interpretable for legal experts, developers and the software system itself (Corsius et al., 2021). Using Regelspraak, ALEF combines legislative and policy analysis descriptions in interpretable knowledge representations. While engineering with these interpretable models could contribute to explaining the automated decisions (Lokin & van Kempen, 2019), they often lack adequate explanation mechanisms.

Therefore, this study examines the newest insights on the explanation of automated decisions and uses those insights to design an explanation method for a rule-based decision-making system in the legal domain. These newest insights are drawn from the field of eXplainable Artificial Intelligence (XAI). Explainable AI is a branch of AI research that focuses on making complex AI models explainable to humans by exposing the decision-making processes of AI systems in a systematic and interpretable way (Samek et al., 2019). By building more transparent, interpretable, or explainable systems, individuals can gain valuable insights into the decision-making processes of intelligent agents (Miller, 2019). The use of XAI in this study helps ensure that the explanation method designed for the decision-making system is effective and human-friendly.

1.1 Research question and objectives

The question that will be answered in this thesis is:

> *How can state-of-the-art insights of XAI be used for the human-centred explanation of rule-based decision-making systems in the legal domain?*

So, the main objective of this research is to develop a human-centred explanation method for rule-based decision-making systems in the legal domain. The study has been divided into theoretical and practical research to achieve this objective.

The theoretical research involves exploring the concept of explainability in AI and building a framework for developing explanation methods. Then, following this framework, the study delves deeper into exploring research on human-centred explanation, explanation techniques for rule-based systems, and explanation within the legal domain. Finally, to complete the theoretical research, we propose a human-centred explanation method of rule-based decision-making systems in the legal domain.

The practical research involves a case study where the proposed explanation method is implemented for a real-world scenario, namely the automated decision-making systems developed using the earlier mentioned tool named ALEF. Then, this implementation is used to generate explanations tailored to different target groups. Finally, the case study results will be used to conclude the effectiveness of the proposed explanation method.

Overall, this research explores insights from Explainable AI to design a human-centred explanation method for rule-based automated decision-making systems in the legal domain (theoretical research). It also implements this proposed explanation method in a real-world scenario to evaluate its effectiveness (practical research). By doing so, we hope to increase the explainability of these systems and provide different target groups with explanations tailored to their specific needs.

1.2 Overview of the thesis

This section offers a concise overview of the chapters within this thesis. The thesis is divided into two main sections, namely a theoretical and a practical part, and comprises nine chapters. Chapters 1 and 9 serve as bookends, with the former providing a forward-looking overview of the thesis and the latter a backwards-looking retrospective.

The theoretical part is structured into three chapters and presents research on explainability from a more conceptual level to research more specifically tailored to the problem addressed in this thesis. Chapter 2 explores the concept of explainability, describing its relevance in the legal domain and XAI. Chapter 3 describes research on the concept of explanation and introduces a conceptual framework to build an explanation method. Chapter 4 follows this conceptual framework and discusses research more tailored to the specific problem this research is trying to solve: human-centred explanation of rule-based decision-making systems in the legal domain.

Chapter 5 bridges the theoretical and practical parts of the thesis by proposing an explanation method building on the main findings of the theoretical research. The practical part of the thesis is also divided into three chapters. First, Chapter 6 discusses the technical details of implementing the proposed explanation method and its integration with the legal decision system used by the DTCA. Then Chapter 7 presents a case study to evaluate the proposed method and its results. Lastly, Chapter 8 evaluates these results found in Chapter 7.

Overall, the thesis contributes to Artificial Intelligence by proposing a human-centred explanation method for rule-based decision-making systems in the legal domain. The proposed method is grounded in a conceptual framework and evaluated through a practical case study. The results offer insight into the method's feasibility and potential to improve the explainability of rule-based decision-making systems in the legal domain.

2 Exploring Explainability

This chapter explores the concept of explainability in the field of Artificial Intelligence (AI). It provides an overview of the terminology associated with explainability, its purpose and the different approaches that can be taken to achieve it.

2.1 Terminology - transparency, interpretability and explainability

In the field of eXplainable AI (XAI), there is an ongoing debate around the terminology used to describe concepts related to explainability, which has led to some ambiguity in the field. Some researchers use the terms interpretability and explainability interchangeably (Vilone & Longo, 2020), while others argue for a clear distinction between the two concepts. For example, Gilpin et al. (2019) state that explainable models are inherently interpretable, but the reverse is not always true. In contrast, Rudin (2019) argues that inherently interpretable models provide their own explanations. Nevertheless, even clearly interpretable models such as rule lists may provide rules that seem illogical to humans trying to understand them (Angelino et al., 2017), and so are not able to explain something to a point "when you can no longer keep asking why" (Gilpin et al., 2019). Barredo et al. suggests a difference between interpretability, or transparency, as a passive aspect of a system's ability to explain itself and explainability as an active aspect of a system to clarify itself (Barredo Arrieta et al., 2020).

Rudin's (2019) argues that "interpretability is a domain-specific notion," implying that any general-purpose definition may be inherently flawed or insufficient. This perspective highlights the need for a more nuanced approach when formulating the terms used in this thesis. It recognises the importance of adapting the definitions to suit the specific context or domain in which a system operates. By doing so, the concepts related to explainability can be better tailored to address each domain's unique challenges and requirements.

The EU High-level Expert Group on Artificial Intelligence identifies three key components of transparency: 1) the ability to track the AI system's processes, known as traceability, 2) the capacity to clarify how the AI system operates, referred to as explainability, and 3) honest communication regarding the use of AI systems and its limitations (The High-Level Expert Group on Artificial Intelligence, 2019). According to the GDPR, transparency is about being open and clear with people about how and why organisations use personal data (GDPR Summary, 2018).

Despite the lack of consistent terms and definitions in XAI research, some common themes can be identified. For example, gaining an understanding (although in various degrees) seems to be a fundamental intention underlying these concepts. Besides, a differentiation is recognised between something that concerns only the system itself and something that goes beyond the system, referring to some form of openness or accessibility. In the light of building an explanation method as a means to share information with individuals, this openness or accessibility is essential. Hence, we embrace the definition of transparency from the GDPR, referring to being open and clear with people about how and why organisations use personal data. Here, interpretability refers to understanding *how* a decision is made based on the processed data. Explainability goes a step further than interpretability, requiring that the decision-making process be able to provide meaningful explanations to individuals about *why* specific decisions were made. Furthermore, transparency could also refer to merely making information accessible such as system code (European Parliament. Directorate General for Parliamentary Research Services., 2019) or system referring to a more general principle of *what* information is being processed.

2.2 Purpose - description and comprehension

Explanation aims to make something clear or understandable between two parties, namely the explainer and the explainee. Explaining can be initiated by either party involved in the communication process (Gregor & Benbasat, 1999). When the explainer initiates the explanation, it is often done to clarify a point, justify a decision, or convince the other person of a particular perspective. On the other hand, when the explainee initiates the explanation, it is usually to resolve a misunderstanding or disagreement that may have arisen during the conversation. In both cases, the purpose of the explanation is to ensure that both parties have a shared understanding of the topic at hand.

Adadi and Berrada (2018) suggested that four reasons support the necessity to explain the logic of an inferential system or a learning algorithm. The first reason is to explain to justify - the decisions made by utilising an underlying model should be explained to increase their justifiability. The second reason is to explain to control - explanations should enhance the transparency of a model and its functioning, allowing its debugging and the identification of potential flaws. The third reason is to explain to improve - explanations should help scholars improve the accuracy and efficiency of their models. Finally, the fourth reason is to explain to discover - explanations should support the extraction of novel knowledge and learning relationships and patterns.

Lacave and Diez (2004) discuss that explanations, i.e. the product that is created through the act of explaining, can serve one of two purposes: description or comprehension. Descriptive explanations aim to show or provide more details about the underlying knowledge or conclusions. In contrast, comprehensive explanations help the user understand the implications of the model or system's conclusions and their relationships. So, description involves displaying intermediate results or providing further details, whereas comprehension involves explaining how each finding affects the conclusion or in combination with other findings. In other words, explanations can be used to show you *how* something works or help you understand *why* something happened.

2.3 Approaches - explainability and explanation

The field of explainable AI has been exploring different approaches to create explainability in decision-making systems, and consequently, several taxonomies have been proposed to categorise these approaches. These taxonomies provide a standardised way to categorise and compare different methods, making identifying and selecting the best approach for our problem easier. Vilone and Longo (2020) discuss the following distinctions:

- Scope: considers the range of an explanation, which can be either global or local. Global explanations aim to make the entire inferential process of a model transparent and comprehensible as a whole, while local explanations explicitly explain only one inference of a model.

- Stage: considers the phase at which a method generates explanations. Ante-hoc methods aim to consider the explainability of a model from the beginning and during training to make it naturally explainable while still trying to achieve optimal accuracy or minimal error. In contrast, post-hoc methods aim to keep a trained model unchanged and mimic or explain its behaviour using an external explainer at testing time.

- Problem Type: considers the kind of problem being addressed, which can be either classification or regression. Methods for explainability can vary according to the underlying problem being addressed.

- Input Data: a model's intake data can play an important role in constructing a method for explainability. The mechanisms followed by a model to classify images can substantially differ from those used to classify textual documents. Thus, the input format of a model (numerical/categorical, pictorial, textual, or time series) must be considered while constructing an explanation method.

- Output Format: considers different formats of explanations, which can be useful for different circumstances and can be considered by a method for explainability. These formats include numerical, rules, textual, visual, or mixed formats.

Miller's (2019) distinction between explainability and explanation can be linked to the Stage category. He distinguishes XAI methods that involve designing understandable decision-making methods, referred to as explainability or interpretability. In contrast, the second approach involves generating explanations for specific decisions made by a system, which Miller calls explanation. Moreover, he notes that these two approaches are complementary, meaning that together, these two approaches can provide a more comprehensive understanding of the decision-making process.

3 A Conceptual Framework for Explanation Methods

This section introduces a framework representing the key concepts that must be considered when developing an explanation method. Section 3.1 discusses the components of an explanation method, while Section 3.2 considers the external factors that influence how each component should be addressed.

3.1 Explanation components - content, communication and adaptation

In this subsection, we analyse three articles from distinct branches of AI that explore processes of human-to-human explanation, the phases for developing explanation methods, and the properties of explanation. By identifying the shared concepts across these articles, we can better understand the components that form explanation methods.

Building on research from social sciences, Miller (2019) provides a comprehensive distinction of the components of human explanation. This distinction shows that explanation consists of a cognitive process involving mental activities like organising relevant information and generating inferences, as well as a product, which is the actual explanation. Furthermore, Miller highlights the importance of the social process of explanation, emphasising the communication and interaction between the explainer and the explainee. When visualised, Miller's ideas distinguish between a one-way cognitive process, a static or tangible explanation product and a two-way interactive process.

Drawing from cognitive engineering, Neerincx et al. (2018) propose a three-phase framework for developing explanations. This framework guides the process of developing explanation methods between agents and humans. The first phase involves generating a coherent justification for a particular action, while the second phase focuses on effectively communicating the explanation to the user. Lastly, the third phase evaluates how well the user comprehends the explanation. Neerincx's framework provides a systematic approach to explanation development, ensuring that relevant steps are considered and optimised.

In their work on expert systems, Lacave and Diez (2002) propose a framework that categorises explanation properties into content, communication, and adaptation. While the specific properties will be discussed later in this thesis, the categorisation enables an understanding of the different aspects of explanation methods. For example, the content category addresses what should be included in the explanation, the communication category focuses on how the explanation should be effectively conveyed, and the adaptation category refers to whom the explanation should be tailored.

Analysing the overlapping ideas across these articles reveals a common thread. While each article may use distinct labels, they all refer to three key aspects that share some conceptual similarities (visualised in Figure 1). The first aspect corresponds to a one-way process involving identifying relevant information and constructing a justification. The second aspect encompasses a tangible component of explanation, such as the actual explanation itself or its delivery to the user. Lastly, the third aspect pertains to a two-way process involving interaction and adapting the explanation to the user's needs. Note that the Conceptual Framework uses the terms content, communication and adaptation as they closely depict the fundamental components of an explanation method.

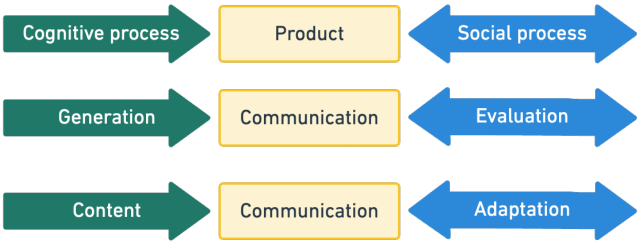


Figure 1

Visualisation of the different explanation components. From the top down, this figure shows the processes (Miller, 2019), phases (Neerincx et al., 2018) and property categories (Lacave & Díez, 2002) of explanation.

3.2 Contextual dependencies - human recipient, decision-making system and domain

In addition to understanding the different components of an explanation method, it is crucial to consider the contextual dependencies that influence how each component should be addressed. While the previous subsection notes some dependencies, this subsection substantiates the specific dependencies included in the Conceptual Framework.

Sormo (2005) proposes that when selecting an appropriate explanation method for an AI system, it is crucial to consider three distinct dependencies: user, system, and domain. User dependency refers to the expectations and needs of the individuals interacting with the system, which can vary based on their expertise and familiarity with the technology. The system dependency relates to the properties of the AI system itself, including its complexity, type of input or output, and level of autonomy. Finally, domain dependency refers to the specific context in which the AI system operates, such as the industry, field, or application area. By taking into account these three dependencies, developers can make informed decisions about the most suitable explanation method that aligns with the users' needs, the properties of the AI system, and the contextual factors of the domain.

While "human-centred" and "user-centred" are often used interchangeably, there is a subtle difference between them. "User-centred" refers to designing products or systems that meet the end-user's or customer's needs and preferences. On the other hand, "human-centred" takes a broader approach that encompasses a diverse range of individuals and groups that interact with or are affected by the system. Therefore, to highlight the importance of considering all the individuals that could benefit from an explanation, the term "human-centred" will be used instead of "user-centred" (unless describing other user-centred studies).

3.3 Conceptual framework for explanation methods

This section combines the findings of sections 3.1 and 3.2 to introduce a conceptual framework of explanation methods. This framework is visualised in Figure 2 and shows the key concepts that must be considered when developing an explanation method, including their interdependencies.

Section 3.1 discussed three essential components for developing an explanation method: content, communication, and adaptation. Recall that the content component focuses on determining *what* should be included in the explanation. The communication component addresses *how* the explanation should be conveyed, while the adaptation component deals with how to tailor the explanation to *whom* it addresses (Lacave & Díez, 2002).

Section 3.2 established that determining an appropriate explanation method necessitates considering three distinct dependencies: the user (or human), the system, and the domain (Sørmo et al., 2005). The role of these dependencies is added to the Conceptual Framework to enhance clarity and completeness. So, the term "human recipient" refers to the person who receives the explanation. Additionally, "decision-making system" refers to the system's part responsible for generating decisions rather than the part responsible for generating explanations, as both can be considered part of the system.

Note that the contextual dependencies are represented in the main objective of this study, which is to develop a **human**-centred explanation method for rule-based **decision-making systems** in the legal **domain**.

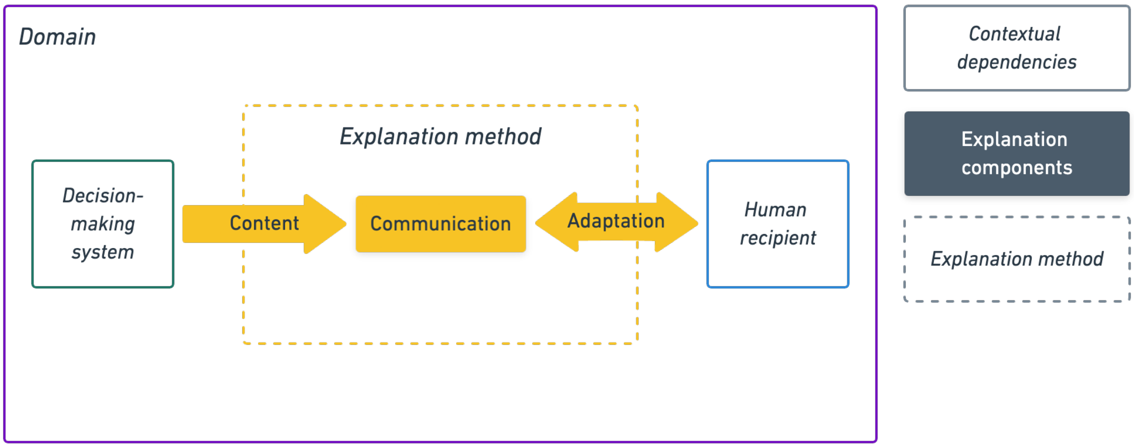


Figure 2

The conceptual framework for explanation methods. This figure shows the key components of an explanation method and contextual dependencies.

3.4 Considerations specific for the context of this study

Before investigating how an explanation method could conform to the contextual dependencies (in Section 4), we briefly examine each factor to describe some of their properties that are specific to this study. These considerations will clarify the focus of this further theoretical analysis and determine the scope of this thesis.

These specifications are based on the context of the decision-making systems considered in the practical research, which are developed using the ALEF tool of the Dutch tax authority. While a more in-depth analysis of these systems will be discussed in the practical part of this thesis (from Section 6 forward), some more generic characteristics from this context are highlighted here. These generic characteristics ensure that the explanation method can be applied to other rule-based decision-making systems beyond the ALEF tool, expanding its applicability and relevance.

The decision-making system operates on a *rule-based* reasoning process employing rules that are typically structured as propositional statements with condition and action parts, commonly expressed as if-then statements. These pre-defined rules are explicitly encoded into the system and serve as the basis for decision-making. Secondly, the decisions made by these systems are *deterministic*, meaning that the system's behaviour is predictable and consistent without any randomness or ambiguity. Probability does not play a role in the decision-making process. Finally, the system's reasoning relies on knowledge from *(domain) experts*, who provide the necessary expertise and information to define and encode the rules into the system.

Developing an explanation method that is catered to the human recipient poses a great challenge, namely regarding the subjectivity of the explanation experience. Each individual may perceive and interpret an explanation differently based on their unique background. Considering the wide range of individuals that could benefit from an explanation, conducting a user study with a representable group is deemed unfeasible within this study. Instead, as suggested by Sörmo (2005), the goals and capabilities of prototypical recipients will be assumed. The following individuals that could benefit from an explanation are identified:

- **Legal experts:** These individuals possess legal knowledge and expertise in a specific area of law. They deeply understand the legal principles and regulations relevant to the decision-making process.

- **Model experts:** Model experts utilise the legal expert's knowledge to create a formal model that captures the system's legal behaviour, structure, and other relevant aspects. They are crucial for translating legal concepts into a formal representation that the system can understand and utilise.

- **Software developers:** Software developers are responsible for developing the automated decision system itself. They encapsulate the legal knowledge within the system models and translate it into a usable application for generating legal decisions.

- **Legal professionals:** Legal professionals utilise the automated decision system to perform calculations, interpret the results, and make informed legal decisions. They rely on the system's outputs to support their decision-making process.

- **Legal support professionals:** Legal support professionals work within customer service or other relevant departments to provide legal support and explanation to data subjects regarding the decisions made by the system. They serve as a bridge between the system and the individuals affected by its decisions.

- **Data subjects:** Data subjects are the individuals whose data is processed by the decision system and whose legal rights and interests are directly affected by the decisions made by the system. They have a stake in understanding the reasoning behind the decisions that impact their lives.

In the context of the legal domain, two important factors should be considered when designing an explanation method: regulations on the explanation of automated decisions and the structure behind legal decisions. Firstly, the legislative aspect is important as legal decisions often have significant consequences for the data subjects involved. Therefore, it is imperative for the explanation method to be compliant with the relevant legislation and align with the legal requirements that govern the domain. Secondly, the legal domain is characterised by a complex network of concepts, rules, and relationships. Understanding these elements is essential to comprehend the underlying reasoning behind the decision-making systems.

At last, it should be noted that the explanation method must also be practical and workable in real-world settings. A requirement that will be considered within this study is that the provision of explanations should be automatable, i.e., the explanation method should generate explanations with minimal human intervention after the initial setup. However, additional factors should be considered to ensure the method is efficient, scalable, and workable in various scenarios. However, these factors are considered out of the scope of this thesis.

4 Theoretical Investigation

4.1 Human-centred explanation

This section focuses on the human-centred aspects of explaining decision-making systems by discussing XAI research that offer insights from philosophy and social sciences. These insights can help design explanation methods that align with human cognitive processes and communication dynamics. Section 4.1.1 discusses Miller's findings on what constitutes a human-friendly explanation and Section 4.1.2 explores the concept of explanation goodness and provides a set of objective criteria for evaluating and designing explanations.

4.1.1 Human-friendly explanations

Research from social sciences can provide valuable insights into how humans explain things to each other and what constitutes a human-friendly explanation. Miller conducted an extensive survey on this topic and presented insights for explainable AI (Miller, 2019). Miller notes that these findings are often overlooked in artificial intelligence but are critical for designing intelligent systems that offer effective and useful explanations.

The first major finding is that explanations are contrastive (Lipton, 1990). People do not simply want to know why an event happened but why it happened instead of some other event. In other words, people want to know the reason for the difference between what happened and what could have happened. This has computational implications for explainable AI, requiring models that can handle contrastive explanations.

The second finding is that explanations are selected in a biased manner. People rarely expect an explanation that consists of an actual and complete cause of an event. Instead, they select one or two causes from a sometimes infinite number of causes as the explanation. However, this selection is influenced by certain cognitive biases. Therefore, it is crucial to understand these biases and their impact on explanation selection. The contrasting idea between selectiveness and truthfulness can be challenging when designing explainable AI systems: it requires a balance between providing comprehensive and accurate explanations while being understandable and useful to the user.

The third finding highlights that explanations relying solely on probabilities are often less effective than explanations incorporating causal factors. While truth and likelihood play a role in explanations, it is essential to provide underlying causal explanations for statistical generalisations to be satisfying. For example, people tend to favour abnormal causes when explaining events (Tversky & Kahneman, 1981). These abnormal causes represent factors that, if removed, have the potential to alter the outcome, making them crucial in comprehending the event. Therefore, for explanations to be effective, the emphasis should be placed on clarifying causal relationships rather than solely relying on probabilities.

The fourth and final finding is that explanations are social as they involve sharing knowledge within conversation or interaction. The explainer may adjust the explanation to align with the explainee's beliefs to ensure that the explainee accepts and integrates the explanation into their existing knowledge as people tend to ignore information inconsistent with their beliefs, also known as confirmation bias (Nickerson, 1998). Because of the social nature of explanations, models of how people interact regarding explanations are essential for building truly explainable AI systems.

Miller notes that these four significant findings converge on a crucial point - explanations are not just about presenting associations and causes but are also highly contextual.

4.1.2 Explanation goodness

Explanation goodness is a term used to describe how well an explanation is constructed, based on axiomatic assumptions about what makes an explanation 'good' (Mueller et al., 2019). In the following section, we will delve deeper into the concept of explanation goodness and explore its underlying principles, building on a series of papers on Explaining explanation for "Explainable AI" (Hoffman et al., 2018; Hoffman et al., 2019; Mueller et al., 2019).

Mueller et al. (2019) refer to explanation goodness as "the set of principles by which guide the development of explanations, and through which they can be reasonably evaluated without relying on human participants" (p. 105). Accordingly, their research presents a list of requirements for a good explanation, called the goodness criteria. These criteria indicate that a good explanation should be understandable, satisfying, sufficiently detailed, complete, usable, useful, accurate and trustworthy (Hoffman et al., 2018; Hoffman et al., 2019).

While a set of objective and validated criteria are useful for designing and evaluating explanation methods, the researchers do not provide further specifications on these criteria (e.g. by defining each term). Hence, we explored complementary research to extract the meaning, approach and relevance of each criterion, summarised in Table 1. Understanding the meaning of each criterion is necessary to determine whether it is met, while the approach is crucial for developing explanations that meet the desired standards. Additionally, considering the relevance helps us understand its importance for recipients.

In conclusion, this section has shed light on the human-centred perspective of explaining decision-making systems. By exploring the concept of human-friendly explanations, we have learned that effective explanations are contrastive, selective, and incorporate causal factors. Moreover, understanding the biases and social dynamics involved in explanation is crucial for designing explainable AI systems that are useful and understandable to users. Additionally, the exploration of explanation goodness has provided a set of objective criteria for evaluating and designing explanations. Incorporating these criteria ensures that explanations are understandable, satisfying, detailed, complete, usable, useful, accurate, and trustworthy.

4.2 Explanation of rule-based decision-making systems

4.2.1 Question-driven explanation of decision-making systems

At the core of question-driven explanations is the understanding that an explanation is "an answer to a (why-)question" (Miller, 2019). Furthermore, these questions are closely related to users' goals because they are often the underlying motivation behind user questions (Hoffman et al., 2019). In other words, users ask questions because they have a particular goal, such as understanding why a system behaves in a certain way or how to accomplish a specific task. Understanding the relationship between users' questions and their

Criteria	Meaning	Approach	Relevance
Understandable	The clarity and comprehensibility of the explanation.	Use simple language, avoid jargon or technical terms, provide context and examples, and use visual aids if necessary.	To ensure that recipients can comprehend the reasons behind automated decisions.
Satisfying	Whether the expectations of a recipient are met.	Address the recipient's concerns and answer their questions effectively, providing relevant and accurate information.	To give recipients confidence in the system and ensure they accept and trust the automated decisions.
Detailed	Depth of information to be useful for the recipient.	Provide transparent and complete information about the data and algorithms used and the specific features that led to the decision.	To ensure recipients understand the system's decision and detect potential biases or errors.
Complete	Coverage of all relevant aspects of the decision-making process.	Ensure that all relevant information about the system, including the factors that influenced the decision, is provided.	To ensure that recipients comprehensively understand the decision-making process.
Usable	A system's ease of use, considering user-friendliness, intuitiveness, and clarity of instructions and interfaces.	Use a user-friendly format, such as interactive visualisations or summaries, that the recipient can easily access.	To ensure that recipients can easily access and understand the explanation.
Useful	How well the system meets recipients' needs and helps them achieve their goals.	Provide actionable information that helps recipients to understand the decision and how to act upon it.	To ensure that recipients can make informed decisions based on the explanation.
Accurate	The correctness of the system's information or decisions, including information on past performance and consistency.	Provide information on how well the decision system has performed in the past or how it is expected to perform in the future and provide information on the consistency or stability of the decision system or algorithm	To ensure that recipients can trust the decision-making process and its decisions.
Trustworthy	The recipient's ability to rely on and trust the system, including transparency about limitations or uncertainties.	Provide complete and accurate information about the data and algorithms used and the validation process of the system. Use credible sources, be transparent about limitations or uncertainties, and provide clear and accurate explanations of technical terms or concepts.	To ensure that recipients have confidence in the system and trust the automated decisions it provides.

Table 1

Goodness criteria for evaluating explanations of automated decision system (Hoffman et al., 2018; Hoffman et al., 2019; Mueller et al., 2019).

underlying goals is thus crucial for designing an effective explanation method.

One way to do this is by creating question banks (Liao et al., 2020), which are collections of questions designed to help users understand the system's behaviour. These questions can generate explanations tailored to the user's needs and provide a more personalised and effective explanation. Furthermore, the question banks can be updated and expanded over time as the system evolves and the user's needs change.

Lim and Dey (2009) developed a taxonomy of user needs for context-aware systems. They crowdsourced user questions in various scenarios and identified ten types of information needs: input, output, conceptual model (including why, how, why not, what else, and what if), and non-functional types (such as certainty and control). A conceptual model represents a system's decision and action processes to transform inputs into outputs. This taxonomy, summarised in table 2, enabled them to develop a toolkit that supports the generation of explanations for context-aware applications (Lim & Dey, 2009).

Type	**Definition**	**Purpose**
Inputs	Describes the input options and information sources used by the system	Helps users understand the types of inputs the system uses and the sources it relies on to operate.
Outputs	Informs users of the output options that the system can produce	Helps users understand the various outcomes or actions that the system can produce or perform based on the inputs it receives.
What	Reveals the current system state, such as what output value it produced based on the input it received.	Helps users understand the system's current behaviour and what to expect.
What If	Allows users to speculate what the system might do given a specific set of user-set input values	Helps users anticipate the system's behaviour.
Why	Discloses the reasoning behind the system's output value given the input values	Helps users understand the decision-making process.
Why Not	Reveals why an alternative output value was not generated given the input values	Helps users achieve an alternative output value.
How (To)	Explains how the system can generally produce alternative output values	Helps users understand how to achieve a desired output.
Certainty	Informs users about the confidence level of the output value produced by the system	Helps users determine how much trust to place in the output value.
Control	Describes how to change settings or thresholds in the system	Helps users understand how to control the system.
Situation	Provides information about the current situation or context in which the system operates	Helps users understand the context of the system's behaviour.

Table 2

Summary of the Explanation Types in Context-Aware Systems (Lim et al., 2009).

Context-aware systems adapt their behaviour based on the user's situation (or context), such as activity, location, and environmental conditions (Lim et al., 2009). Often, these systems use complex rules or machine learning models to make decisions and rely on implicit input that may be collected without explicit user involvement, so-called calm computing (Weiser & Brown, 1995). This can create challenges for users of context-aware applications in understanding the system's behaviour (Bellotti et al., 2002; Bellotti & Edwards, 2004). Moreover, a lack of system intelligibility can lead the user to mistrust, misuse, or even discontinue using the system, especially when there is a mismatch between user expectations and the system's behaviour (Muir, 1994).

While rule-based legal decision systems are not context-aware systems, some similarities exist. For example, both systems may rely on data sources that are not directly known or visible to the user (or data subject), which can create challenges in understanding how decisions are made and why. So, we expect that the explanation types discussed above are also applicable to explain legal decision-making systems.

Additionally, the researchers insist that designers should use visualisations as a means of augmenting explanations as it enhances the understandability of the users (Lim & Dey, 2009). For example, visualising a system's overall conceptual model can help someone understand how different components and processes are interconnected and how they contribute to overall system behaviour. Specifically, when it comes to decision trees, visualising the tree and tracing the "Why" trace can offer a quickly interpretable overview of the questions that someone may have about a decision model (Lim et al., 2009). While some technical knowledge may be required to utilise the visualisation properly, it is proven effective in providing a complete explanation to people with varying technical expertise.

In conclusion, creating human-centred explanations in explainable AI is a challenging task that requires a highly adaptable method that can cater to the unique needs of each person. Key findings suggest that a question-driven framework, aided by visualisations, can help achieve this goal and provide an adaptable and effective explanation method. By focusing on answering individuals' specific questions and visualising the decision-making process, designers can create explanations that are easy to understand and will help someone achieve their specific goals.

4.2.2 Explanation of rule-based decision-making systems

This thesis considers rule-based systems as systems that use so-called propositional rules, which are widely used in various domains and comprise condition and action parts (Huysmans et al., 2011). The condition part includes a combination of conditions on input variables, while the action part specifies the appropriate action to take when these conditions are satisfied. Figure 3 shows three common ways of representing propositional rules using textual descriptions, decision tables, and decision trees. They all represent whether someone is eligible for a benefit. To be eligible for the benefit, a person must meet the income and assets criteria defined by the first rule. If either criterion is not met, the person is not eligible.

Textual descriptions often use an if-then format (see 3a). They provide a natural and straightforward way to represent propositional rules. Decision tables offer a structured and organised way to represent propositional rules (Vanthienen & Wets, 1994). Each row in a decision table represents a combination of conditions on input variables, while each column represents a different condition or action. Decision trees represent propositional rules graphically, with the input variables used to create a tree structure. Each internal node in the decision tree represents a condition on an input variable, while each leaf node represents an action.

Note that these examples are very simple, and representations could become unclear when the complexity of the rules and the number of input variables and conditions increase.

IF income $\leq$ threshold AND assets $\leq$ threshold THEN eligibility = Yes

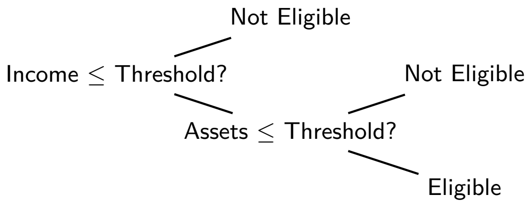
IF income $>$ threshold THEN eligibility = No

IF assets $>$ threshold THEN eligibility = No

(a) *Textual description*

Income	Assets	Eligible
$\leq$ Threshold	$\leq$ Threshold	Yes
$>$ Threshold	-	No
-	$>$ Threshold	No

(b) *Decision table*



(c) *Decision tree*

Figure 3

Example of three representations of propositional rules.

4.3 Explanation in the legal service domain

This section describes the legal context of the explanation of automated decisions. This section examines the need for an explanation regarding individuals' rights and the legislation that applies to the decision systems and its explanations. Then, it discusses the network of concepts, rules, and relationships, which must be considered to understand the reasoning behind these systems.

4.3.1 The need for explainability

Automated decision-making systems are widely used in the legal domain, allowing for efficient and mass decision-making based on legal rules (Timmer & Rietveld, 2019). For example, in the Netherlands, various executing agencies, such as DUO, UWV, and SVB, use these systems to automate complex legal rules, make decisions, disburse and collect money, and communicate messages to populations (Lokin & van Kempen, 2019). To keep up with societal demands, these systems need to be agile, i.e. able to adapt to external changes such as modifications in legislation or policies.

While these systems can improve the efficiency of legal processes and services, they can also affect individuals' rights or legal quality, especially when there is no careful and transparent design process (Timmer & Rietveld, 2019). Hence, regulations are formulated for automated-decision systems to protect human rights. Still, the negative impact remains as, in practicality, the used systems do not always comply with these regulations (van Eck, 2018). Van Eck studied automated decisions in Dutch legal practice and identified conflicts between automation and some legal principles. For instance, when rule-based systems are not tailored to people with exceptional circumstances, the principle of equality and non-discrimination may be infringed. Besides, the right of defence may be infringed when the underlying rules of a decision can not be assessed. This lack of insight is also noticed by Lokin, who concluded that the process from legislation to rule-based systems is often not transparent (Lokin, 2018).

The importance of transparent law enforcement systems is evident. While technological innovations can boost efficiency and possibilities for governmental organisations, the rapid digitalisation of legal processes also negatively impacts society if used incorrectly. These issues raise questions about the explainability of automated decision systems.

4.3.2 Regulations on explanation

The General Data Protection Regulation (GDPR) is a regulation in EU law on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA)[1]. As personal data is required to make automated legal decisions, the GDPR applies to the research subject of this thesis. However, not all GDPR is relevant for designing an explanation method.

The "right to explanation" within the GDPR has received significant attention, as it can be disruptive and technically challenging for AI systems to explain their decisions (Wachter et al., 2017). While the "right to explanation" only appears in a nonbinding Recital, the "right of access" guarantees individuals "meaningful information" about the "logic involved" in automated decision-making. This regulation notes some requirements on *how* information should be provided. For example, organisations must provide subjects with clear and easily accessible information. This means must be in plain language, free of charge, and include all relevant details. It can be provided in written form or through other means, such as an online portal. Although this "right of access" does not specify *which* information must be provided in practice (Wachter et al., 2017), a general and easily understood overview of system functionality is likely sufficient (Goodman & Flaxman, 2016).

To provide insights into *what* information must be provided, guidelines are created to clarify the regulations and their implications concerning specific subjects, such as the one provided by the Information Commissioners Office (ICO), which distinguishes six explanation types (Information Commissioners Office & Alan Turing Institute, n.d.).

- Rationale: explains the reasoning behind a decision-making system, including the logic or methodology used to arrive at a decision. This is particularly relevant to understanding the decision-making process.
- Responsibility: explains who is responsible for the decision-making system. It clarifies who has ownership or control over the system, who is accountable for its use, and who is responsible for ensuring that it is used in a way that is consistent with legal and ethical standards.
- Data: explains the data used in a decision-making system, including how it is collected and processed. It ensures that individuals know what information is used to make decisions about them.

[1] Within the Netherlands, the GDPR is implemented in the Algemene Verordening Gegevensbescherming (AVG)

- Fairness: explains whether the decision-making system is fair and unbiased. It ensures that decisions are not based on discriminatory or arbitrary factors, in line with human rights and the rule of law.
- Safety and performance: explains whether the decision-making system is accurate, reliable, secure, and robust. It ensures that the system performs as intended so that it can be relied upon and trusted.
- Impact: explains whether the impacts of the decision-making system on individuals are monitored. It ensures that the system is designed to assess and monitor the impacts on individuals over time to ensure that it is responsive to their needs and concerns and can be adjusted if necessary.

Together, these six types of explanations help to ensure that decision-making systems are transparent and accountable. By understanding these different types of explanations, businesses and organisations should be able to ensure that their decision-making processes align with legal and ethical standards and ultimately benefit individuals (Information Commissioners Office & Alan Turing Institute, n.d.).

4.3.3 Structure of legal decisions

Translating legislation into practical application requires a structured approach where all the steps for interpreting, specifying, and elaborating on the legislation are explicitly documented. Such an approach ensures that the choices made when translating legislation into practice are clear, and decisions based on those choices can be explained and justified. Besides, such an approach makes it more straightforward to determine what adjustments are needed in ICT systems when legislation changes, making implementation organisations more adaptable to changes in legislation. An example of such an approach, called Law analysis ('Wetsanalyse' in Dutch), is described in the book by Ausems, Bulles and Lokin (2021).

A crucial element of this approach is the legal analysis schema, which provides a visual overview of the different elements within legislation and their relationships. The schema is based on the work of American jurist Wesley Newcomb Hohfeld (1913, 1917), who identified various types of rights and obligations based on American jurisprudence in the early 20th century.

As shown in Figure 4, the legal analysis schema identifies various elements within the legislation, including the legal subject, legal object, legal relationship, legal fact, condition, derivation rule, variable, variable value, parameter, parameter value, operator, time indicator, place indicator, delegation power, delegation elaboration, and source definition. Each of these elements plays a crucial role in understanding and applying the legislation, and the legal analysis schema helps illustrate how these elements interact.

For example, the legal subject is the person or entity holding rights and responsibilities, while the legal object is the object of a legal relationship or legal fact. The legal relationship describes the connection between two legal subjects, while the legal fact refers to an act, event, or time-lapse causing a change in the legal situation. The schema also includes other important elements such as conditions, which describe the requirements that must be fulfilled, and derivation rules, which create new values or facts.

Using the legal analysis schema, legal professionals can better understand the various elements within legislation and their interrelationships. This can help them to interpret and apply legislation more accurately and effectively, ensuring that the choices made during the translation process are clear and justifiable. Ultimately, this can lead to a more transparent and adaptable implementation of legislation.

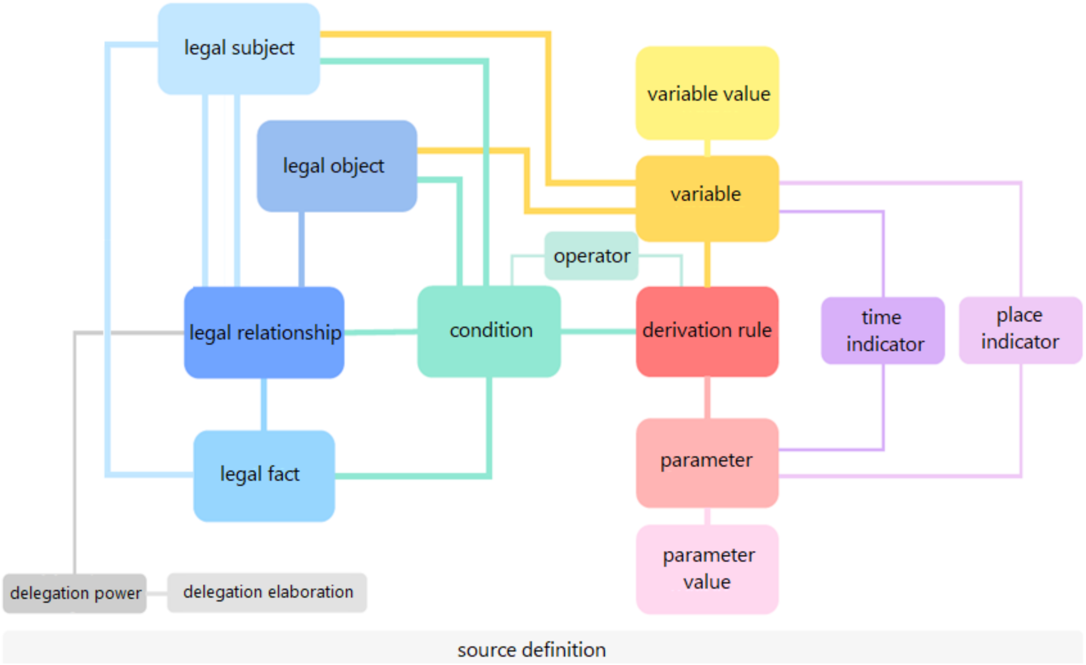


Figure 4

Diagram representing the legal analysis framework. This diagram is a translation of the diagram presented by Lokin et al. (Lokin et al., 2021; Open Rules, n.d.).

Legal subject	Person or entity holding rights and responsibilities.
Legal object	Object of a legal relationship or legal fact.
Legal relationship	Connection between two legal subjects.
Legal fact	Act, event, or time-lapse causing a change in legal situation.
Condition	Requirements that must be fulfilled.
Derivation rule	Rule that creates new values or facts.
Variable	Value description that could vary per legal situation.
Variable-value	Specific value of a variable in a given legal situation.
Parameter	Value that remains constant during a specific period.
Parameter-value	Specific value of a parameter during a specific period.
Operator	Formula representing an arithmetic operation, an equation, or a compound condition.
Time indicator	Point or period of time.
Place indicator	Specific location or area.
Delegation power	Authority to establish additional rules in a lower regulation based on an act or decree.
Delegation elaboration	Delegated regulation that establishes further rules.
Source definition	Definition that is set in legislation.

5 Design of the explanation method

This chapter links the theoretical research discussed in previous chapters and the practical application of the proposed explanation method. It contains two subsections: "Requirements of the Explanation Method" and "Proposing the Explanation Method." The objective of this chapter is to derive a set of requirements from theoretical research and introduce an explanation method that seeks to meet these requirements.

5.1 Requirements of the explanation method

For this thesis, a 'good' explanation method is considered as one that can provide 'good' explanations. Building on the properties and requirements listed in Table 3 can be used to ensure that the explanation method effectively produces these 'good' explanations. Building on theoretical research, the properties of explanation methods (Lacave & Díez, 2002) are connected to the requirements for a 'good' explanation (Hoffman et al., 2018). Together, they provide a complete overview of what to consider when developing and evaluating an explanation method and its generated explanations.

	Property	Description of Property	Requirement	Description of Requirement
Content	Focus	Breadth of the explanation.	Complete	Cover all relevant aspects.
	Level	Depth of the explanation.	Detailed	Provide enough detail on each aspect.
	Causality	Decision-cause relationships.		
	Purpose	Intent of the explanation.	Useful	Align purpose with recipient's goals.
Communication	Interaction	Interface used to request explanations.	Usable	Accessible, or easy to use.
	Display	Format used to present explanations.	Understandable	Clear, or easy to understand.
Adaptation	Domain knowledge	Adaptability to recipient's domain expertise.	Adjustable	Tailor to the recipient's understanding of the modelled domain.
	System knowledge	Adaptability to recipient's system expertise.	Adjustable	Tailor to the recipient's understanding of the system's inner workings.
	Goals	Adaptability to recipient's goals.	Adjustable	Tailor to the recipient's goals for using this system.

Table 3

Required properties of an explanation method with the corresponding requirements for its generated explanations, building on the properties of explanation methods according to Lacave and Diez (2002) and the goodness criteria presented by Hoffman et al. (2018).

So, a 'good' explanation method is considered as one that can provide 'good' explanations, whereas a 'good' explanation is considered to be both understandable and useful to the recipient. When we say an explanation is understandable, we mean that it should consider the individual's knowledge and be presented in a clear and accessible manner. On the other hand, when we say an explanation is useful, we mean that it should consider the individual's goals and provide information that helps them achieve those goals. So, to generate 'good' explanations, we need to clarify the relevant characteristics of each group that explanations can target. Table 4 shows the characteristics of each target group, including their goals, legal domain knowledge, and understanding of the decision system's reasoning. These groups can be divided into those involved in creating the decision system (creators) and those involved in using the system (users).

	Recipient	**Goals**	**Domain Knowledge**	**Reasoning Knowledge**
Creators	Domain experts	Knowledge acquisition and validation	High	Low
	Model experts	Model creation and verification	Medium	Medium
	Software Developers	System development and software testing	Low	High
Users	System end-users	Decision making	Medium	Low
	Support staff	Explanation	Medium	Low
	Data subjects	Comprehension and protection of rights	Low to medium	Low

Table 4

Characteristics of the individuals involved with automated legal decision-making systems.

When considering an explanation as an answer to a question (Miller, 2019), we want the explanation method to create understandable explanations that can answer all of the target group's questions. These questions can be categorised into specific categories, such as the one provided by Lim et al. (Lim & Dey, 2009, 2010) (see Table 2).

However, it is important to note that not all of these explanation types may be relevant in the context of automated legal decisions. For example, the explanation type of control may not be applicable for since the rules used in the decision model are fixed and cannot be modified by the user. Besides, the explanation type of certainty is irrelevant since the outputs of rule-based decision systems are deterministic and thus considered certain.

Looking at the questions in Table 2, they specifically consider the users of a system and fit the goals of the user target groups in Table 4. Consequently, they do not seem to fit the goals of the individuals involved in creating the decision system. To also satisfy these goals, three other question types are assumed to be necessary. The first one regards displaying the elements of a model, which is useful for model creation and system development. The second one considers the ability to retract the legal sources used for building the model, which is useful for validation. Finally, whether questions help with answering questions regarding verifying and testing a decision model and system.

Accordingly, Table 5 presents the relevant explanation types for automated legal decisions, divided into three categories: decision model, decision system and decision. The decision model category helps with answering questions for the creation groups (as discussed before). The decision system's category includes explanations on how to use the system, while the decisions category includes local explanations regarding a specific decision-making instance. Besides, we also differentiated the purpose of explanation types based on whether they describe something or clarify something, like the distinction in purpose between description and comprehension (Lacave & Díez, 2002).

Note that there are other differences with the explanation types as presented in Table 2. For example, the What Else explanation provides information closely related to the situation explanation and more about what the system has done but did not reveal (Lim & Dey, 2009). We also distinguished between How and How To explanations, where the first clarifies how the system generally makes certain (intermediate) decisions, while the latter clarifies how to achieve a desired decision, given some (but not all necessary) input values.

	Type	**Purpose**
Decision	What	Describes what decision(s) it made and what intake it used.
	What else	Describes any intermediate decision(s) or used data not provided by the user.
	What If	Describes the system's decision when an input value is changed.
	Why	Clarifies the reasoning behind the system's output value given the input values.
	Why Not	Clarifies why an alternative output value was not generated given the input values.
	How To	Clarifies how to achieve a desired decision, given some (but not all necessary) input values.
System	Inputs	Describes the intake the system could use.
	Outputs	Describes which decisions the system could make.
	How	Clarifies how the system generally makes a certain decision.
Model	Model Elements	Describes the elements (such as rules) and relations within the decision model.
	Legal Sources	Describes the legal sources used to create the model.
	Whether	Clarifies whether the system meets a certain requirement.

Table 5

Different explanation type questions for automated legal decisions, building on the explanation types by Lim et al. (Lim & Dey, 2009).

5.2 The proposed explanation method

This section presents the proposed human-centred explanation method for rule-based decision-making systems in the legal service domain. It describes the key characteristics of the technology behind the explanation method and the possible explanations it could generate. Moreover, it discusses how these address the requirements of the explanation of automated decision-making systems, as discussed before.

5.2.1 Explanation through graph database management systems

The proposed explanation method leverages Graph Database Management System (GDBMS) to generate explanations. In short, a GDBMS makes it possible to store data in graph structures (graph databases) and allows for intuitive questioning and visualising of these data structures Robinson et al., 2015. While Chapter 6 will go into more technical detail, this section substantiates why GDBMS provide a solution for explaining rule-based decision-making systems.

The use of GDBMS offers several benefits for generating explanations that can be related to the requirements of an explanation method as given in Table 3. The following properties of GDBMS follow a reverse order of the adaptation, communication and content categories:

- Perspectives: GDBMS enable the creation of customised views that can be *adapted* to different target groups.
- Querying and visualisation: GDBMS enable someone to *interact* or question the stored data through intuitive expressions (queries). Besides, they not only enable someone to view stored data in a (textual) table but could also allow to *display* the underlying graphical structures through visualisations.
- Subsetting and revealing relationships: GDBMS enable adjusting the presented information through the use of filters (f.e. by only representing certain data types) and queries (f.e. by only presenting the answer to a specific question). This changes which subset of the database is shown, adjusting both the *level* and *focus* of an explanation. Besides, GDBMS can help uncover the connections between variables (representing storage locations of a decision system) and rules (representing conditional actions that are dependent on and initialise variables), revealing *causal* relationships between them. When displaying these points and their relations in a hierarchical structure, the underlying steps of the decision-making process become clear.

All in all, using GDBMS for the explanation of rule-based systems provides flexibility in adapting both the content and communication of an explanation to different target groups. Individuals can explore the decision-making process through visual graphs, answers to questions, and customised filtering options, allowing them to gain insights into the reasoning behind automated legal decisions tailored to their specific needs.

5.2.2 Explanations of the proposed method

As is now known, the proposed explanation method utilises a graph database management system to create visual representations of the decision-making process and offers textual answers to key questions related to the system and its decisions. This subsection presents the possible explanation components that could result from using such a database system, which can be used for construing explanations.

Figure 5 provides a simplified illustration of the possible explanation components. The first graph can be used for global explanation, representing the general structure of the decision-making process. Secondly, the illustration features a similar graph but populated with specific values, providing more detailed information about a specific decision (local explanation). The overall explanation also includes textual answers to predetermined questions about data and reasoning. By addressing these questions, the explanation method facilitates understanding the decision-making process for f.e. support staff, who can then convey the necessary information to data subjects in compliance with legal requirements.

The nodes in the graph represent the key components of a decision-making system, namely variables or values (blue) and rules (purple, red or green). Variables are named storage locations in programming that hold values that can change. Rules, in the context of rule-based systems, are conditional statements that define conditions and actions. Variables are used within the conditions and actions of rules to represent and manipulate data, enabling the system to make decisions based on the values stored in the variables.

Note that the visualisation of the graphs may vary depending on the decision system it represents. When representing the same system, both the global and local graphs should be of the same form, where the global graph describes 'empty' variables that are instantiated with specific values in the local graph. The decision system that is represented in Figure 5 has a few constraints: all variables are initialised at the start (possibly with null), rules only change one value and values only change once. As a result, one row in the graphs represents what could happen with one particular variable. So, a row in the local graph consists of an input value (light blue), one rule at most that could be fired (green) or not (red) and the output value (dark blue).

The explanation method's value lies in its ability to provide insights into the decision-making process for different stakeholders. To show the method's ability to generate the different explanation components as illustrated in Figure 5, we will look at the model expert and support staff groups. The global graph allows modellers to evaluate the decision model's structure and correctness, enabling them to refine and optimise it. Simultaneously, support staff could receive more detailed explanations of specific decisions to help fulfil their legal obligations to inform data subjects.

In conclusion, the proposed explanation method offers a human-centred approach to explaining rule-based decision-making systems in the legal service domain. By focusing on content, communication, and adaptation, the method ensures that individuals can obtain the information they need to understand the decision-making process and trust its decisions. The use of GDBMS further enhances the explanatory value of the method, facilitating more effective communication and adaptation to user needs. While some aspects of the explanation method could not be fully implemented within the scope of this research, the proposed approach serves as a foundation for future work in explainable artificial intelligence in the legal domain.

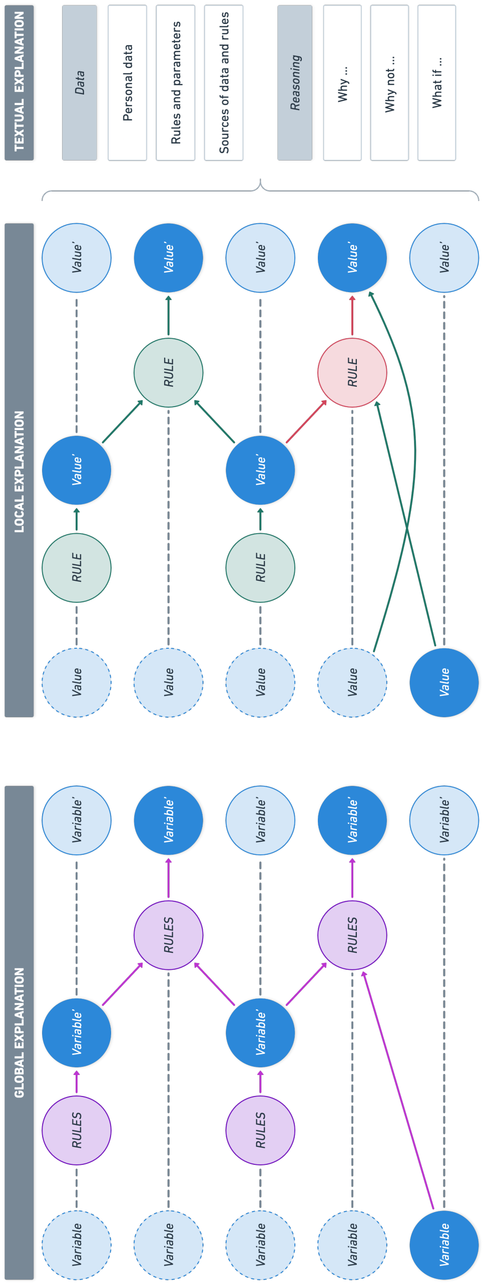


Figure 5

Simplified illustration of explanation using graph database management systems. This explanation involves visualising a graph that represents the decision model (global explanation) and a single decision (local explanation), together with (textual) answers to pre-specified questions.

6 Technical implementation of the explanation method

This chapter provides a detailed description of the technical implementation of the proposed explanation method. It discusses the design of the explanation tool, the technology used to implement it, and how the tool is integrated with the legal decision system. The chapter also describes the challenges encountered during the implementation process and how they were addressed.

6.1 Implementing the explanation tool

For this thesis, Neo4j is used. Neo4j is a graph database management system (GDBMS) designed to store, manage, and query large and complex graphs. It is based on the concept of a graph, where data is represented as nodes and relationships between those nodes. Neo4j is a popular[2] choice for applications that require querying large datasets with complex relationships, such as social networks, recommendation engines, and fraud detection systems (Robinson et al., 2015). The following section aims to create an understanding of database management systems and explains why Neo4j is used for this thesis.

6.1.1 Graph database management systems

This section aims to create an understanding of graph data structures and their applications in database management, building on (Robinson et al., 2015). First, the basic structure of a graph, including nodes and edges, will be introduced. Then, the labelled property graph model will be discussed, which adds additional information about nodes and edges. Finally, graph databases will be explored, including their key characteristics, advantages, and how they differ from traditional relational databases.

A graph is a data structure used in computer science and mathematics to represent relationships between objects. A graph comprises vertices or nodes, which represent the objects, and edges, which represent the relationships between them. For example, in a social network, each person would be represented by a node, and the edges would represent friendships or other connections. In a transportation system, each station or stop would be a node, and the edges would represent the connections between them, such as train or bus routes. Graphs can represent a wide range of objects and relationships and have many applications in fields such as computer science, mathematics, and social science.

The labelled property graph model is a popular way of representing graphs that includes additional information about its nodes and edges (Robinson et al., 2015). In this model, each node can have one or more labels that describe its type or category, such as "person" or "station." Additionally, nodes can have properties that are given as key-value pairs, describing attributes of the object the node represents. For example, a person node might have properties such as name, age, or location. Edges in a labelled property graph model are named and directed, meaning they have a specific start and end node. In addition, edges can also have properties describing attributes of the node relationship. This rich structure makes the labelled property graph model intuitive and easy to work with.

[2] According to DB-Engines, a website that ranks database management systems, Neo4j is currently the most popular graph database (DB-Engines, 2023)

A graph database is a database management system (DBMS) designed to store and manage graph data. Graph databases have two key characteristics: the underlying storage and the processing engine. The first considers how the data is stored and uses the graph structures as explained above, while the second characteristic considers how queries are processed. Graph databases offer Create, Read, Update, and Delete (CRUD) methods that allow users to interact with the graph data using graph queries and traversal algorithms. One of the key advantages of graph DBMS is its ability to handle complex queries efficiently. In traditional relational databases, joins between tables can quickly become a performance bottleneck as the number of tables and data volumes grow. In contrast, a graph structure enables queries to traverse relationships between nodes directly, resulting in faster query times and more efficient use of system resources.

To sum up, graphs are a fundamental data structure representing relationships between objects. A popular example is the labelled property graph model, which provides a way to add additional information about nodes and edges. Graph DBMSs use these underlying structures for managing and analysing complex, interconnected data.

6.1.2 Graph database management system Neo4j

This section aims to illustrate why the Neo4j DBMS is used by showing the explanatory value of its two key characteristics: the underlying storage and the processing engine. Besides, the section notes why Neo4j is suitable for conducting a science project like this thesis.

Neo4j stores data as nodes and relationships, and its database engine allows users to traverse these connections quickly. Another advantage of Neo4j is its uncovering hidden connections that explain complex relationships between data points. With the help of its graph-based model, Neo4j can visualise data and their relationships, enabling users to identify patterns and gain insights that may not be readily apparent using traditional relational databases.

Neo4j is an open-source and ACID-compliant database (Neo4j, n.d.). "Open-source" means that Neo4j's source code is freely available for anyone to view, modify, and redistribute. This allows developers and organisations to use and customise Neo4j without licensing costs or restrictions. This also means that the community of developers provided numerous plugins and extensions, expanding Neo4j's functionality and making it a highly versatile GDBMS. ACID is an acronym that stands for Atomicity, Consistency, Isolation, and Durability. ACID compliance means the database system guarantees that transactions are processed reliably and consistently, even during failures or system errors. In practice, this means that transactions are treated as an indivisible work unit, either completed in full or not at all. So, for example, if a transaction requires multiple steps to be completed successfully, and any of those steps fails, the entire transaction is rolled back, ensuring that the database remains consistent.

In conclusion, the Neo4j DBMS is a powerful tool for visualising data and relationships, enabling users to identify patterns and gain insights that may not be readily apparent with traditional relational databases. Based on a labelled property graph model, its underlying storage allows for displaying hierarchical relationships, which is particularly useful in explaining interrelated steps of an automated decision. Additionally, its processing engine, which uses the Cypher query language, provides an intuitive and versatile mechanism for generating question-driven explanations. Finally, as an open-source and ACID-compliant database, Neo4j is highly versatile, and its functionality can be expanded with numerous plugins and extensions.

6.1.3 Query language Cypher

Cypher is a query language designed explicitly for querying graph databases like Neo4j. It allows users to express complex queries simply and intuitively, making working with complex graph data easier than traditional relational databases (Robinson et al., 2015).

In a traditional relational database, the database schema defines the structure of the data and how it is organised into tables, columns, and relationships. Therefore, to query the data, the user needs to know the schema and the query language (such as SQL) used to retrieve the data. This requires knowledge of the underlying data model and the syntax of the query language. On the other hand, Cypher is designed to abstract away many of the details of the underlying graph data model and query language, which means that users can write queries without needing to have a deep understanding of the graph data model or the syntax of the query language. For example, in a Cypher query, you can specify a pattern of nodes and relationships you want to match rather than specifying the exact table and column names in a relational database.

Cypher is based on patterns: sets of nodes and relationships that match specific criteria (Robinson et al., 2015). Cypher queries start by specifying a matching pattern and then use various commands to extract, filter, and aggregate data from the matched nodes and relationships. Again, this makes Cypher easy to work with complex graph data, as the user can focus on the relationships between the data rather than the structure of the data itself.

Beneath is an example of a simple Cypher query. This query matches all pairs of nodes connected by a "FRIEND" relationship and labelled "Person". It then filters the results only to include pairs where the first node is named "Alice". Finally, it returns the names of the second nodes in each pair.

```
MATCH (n:Person)-[:FRIEND]->(m:Person)
WHERE n.name = ’Alice’
RETURN m.name
```

Overall, Cypher allows users to work with graph data more naturally and intuitively without requiring a deep understanding of the underlying graph data model or query language. This means that the user can focus on the relationships between the data rather than the structure of the data itself.

6.1.4 Visualisation tool Neo4j Bloom

Bloom is a visualisation tool developed by Neo4j to help users explore and gain insights into their graph data through interactive visual representations. It allows users to create customised graph visualisations and perform various actions, such as filtering, sorting, and grouping, on those visualisations. Users can also edit nodes and relationships directly within the Bloom interface to further interact with their data.

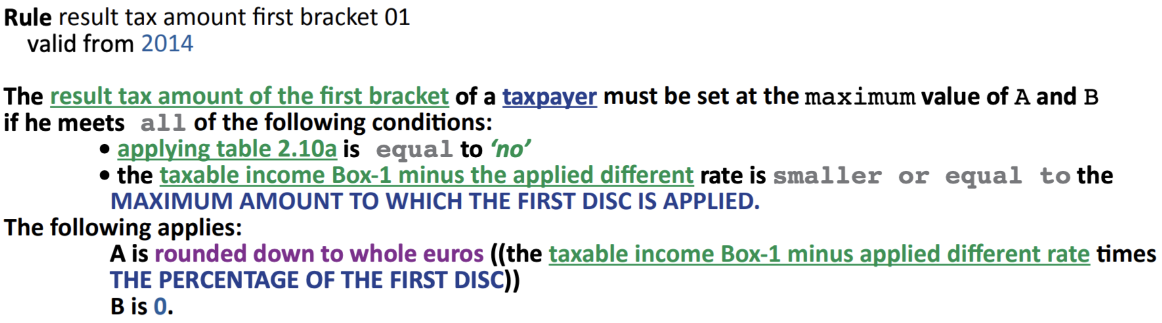
The primary advantage of using Bloom is its ability to reveal complex relationships within data. Patterns and connections that may not be immediately apparent in tabular or text-based formats can be easily identified by providing a visual representation of data. This feature is convenient for obtaining valuable insights that traditional data analysis approaches may struggle to uncover.

Bloom is designed to be user-friendly and accessible to non-technical users, making it available to many users. It offers a drag-and-drop interface, various layout options, and the ability to apply different styles and themes for customising visualisations. The range of customisation options allows users to create visualisations tailored to their specific requirements.

6.2 Integrating with the legal decision system

To illustrate the practical application of the explanation method, we will examine a case study of the Dutch Tax and Customs Administration (DTCA). Due to the high stakes in tax collection, the DTCA is extremely careful when changing the technology responsible for tax calculations. To facilitate adaptation to frequent changes in laws, regulations, and policies while meeting organisational objectives, the DTCA has adopted the Agile Law Execution Factory (ALEF) (JetBrains, 2019). ALEF is a tool developed with JetBrains MPS that uses natural language to specify services, fact patterns, tax rules, and test cases. By analysing legislation and internal policies, ALEF produces structured descriptions of rights, duties, legal concepts, fact patterns, concept descriptions, legal actions, legal actors, legal documents, and legal rules, which are then used to generate service applications.

An example of a tax rule generated by ALEF is the tax calculation for a taxpayer born after 1946 (see Figure 6). The rule is valid from 2014 and calculates the taxes a taxpayer owes based on their income and other factors. The ALEF-generated tax rules are used by a rule engine, a highly scalable solution for mass calculations used by the DTCA back office systems and legal experts. Deploying the Blaze rule code as a decision service on the mainframe ensures efficient tax calculations and adherence to legal requirements.



Rule result tax amount first bracket 01
valid from 2014

The result tax amount of the first bracket of a taxpayer must be set at the maximum value of A and B
if he meets all of the following conditions:
- **applying table 2.10a is equal to *'no'***
- **the taxable income Box-1 minus the applied different rate is smaller or equal to the MAXIMUM AMOUNT TO WHICH THE FIRST DISC IS APPLIED.**

The following applies:
A is rounded down to whole euros ((the taxable income Box-1 minus applied different rate times THE PERCENTAGE OF THE FIRST DISC))
B is 0.

Figure 6

Example of a tax rule in the Agile Law Execution Factory (from JetBrains, 2019)

This section examines the steps involved in retrieving the expert knowledge into graphs. First, the technology used to capture the expert knowledge in models is discussed. Then, the three models specifically used by the decision-making system are discussed and related to the legal concepts from the Legal Analysis Diagram (see Figure 4). The elements that should be represented in the summarising graphs for each model could be determined based on the legal concepts associated with each model. These graph representations will offer a comprehensive view of the decision-making process, aiding decision-makers in understanding the dependencies and relationships between entities, rules, and services in the legal domain.

In short, the implementation process is as follows: we start by importing all the model elements in a structured form, creating an Abstract Syntax Graph. This graph represents the knowledge from all the decision models in a structured form, so we name this the Knowledge Graph. From this Knowledge Graph, we create a simplified graph representing the decision models' key structural and semantic information (the Global Graph). To represent a specific decision, we can "fill in" the Global Graph with the values of a model instance (Local Graph).

6.2.1 Technology behind the decision-making system

As mentioned, ALEF is created with JetBrains Meta Programming System (MPS). MPS is an open-source tool for creating computer languages specialised for particular fields or industries (Bucchiarone et al., 2021). These languages are called Domain-Specific Languages (DSLs), designed to solve specific problems more effectively than mainstream general-purpose languages (van Deursen et al., 2000).

MPS is based on Model-Driven Engineering (MDE), an approach to dealing with complex problems using simplified models of real-world systems (Schmidt, 2006). These models are abstractions of real problems, defined using a set of concepts and relationships that are formalised in modelling language definitions. In the case of MPS, these models are used to generate computer languages that are specific to a particular domain.

The MPS editor is a language workbench tool that helps programmers design, implement, test, and use DSLs created with MPS (Völter, 2013). It provides powerful editing and automation features, allowing programmers to specify the concepts and relationships used in the new language with corresponding semantics and notations. The editor can also generate necessary programming environments, such as editors, auto-completion and validation features, and versioning tools. These features are necessary for creating and using domain-specific languages.

The MPS generator is another integral part of the MPS system. It bridges the gap between the business and implementation domains, transforming the original domain-specific model into one represented in a low-level general-purpose language, such as Java, C, JavaScript, or XML (Bucchiarone et al., 2021). This transformation happens in stages, resulting in a model that can be turned into textual source files. These source files can then be used with traditional compilers, allowing developers to take the business knowledge contained in the domain-specific language and turn it into a working program. Essentially, the DSL formalises the business knowledge of the domain experts, while the generator encapsulates the implementation of that knowledge in a given technology.

6.2.2 Models involved in the decision-making system

In MPS, each model exists of a set of nodes structured in an Abstract Syntax Tree (AST), where each node has a parent node (excluding root nodes), child nodes, properties, and references to other nodes. In addition, each node stores a reference to its declaration, called its concept, which defines the class of a node and its structure. In turn, these concepts are part of a language.

To obtain the AST of the MPS model, MPS provides an export function that can generate an XML file representing the AST. XML (Extensible Markup Language) is a flexible and widely-used markup language used to represent structured data, including the AST of an MPS model. Each node in the AST can be represented as an XML element, with its attributes and children representing the properties and sub-nodes of the corresponding AST node. To represent the AST in the Neo4j graph database, each node in the AST is represented as a node in the graph database, with its properties and relationships representing the attributes and child nodes.

In practice, a decision system uses multiple models representing its reasoning. Hence, the graph database will be more like a graph than a tree, where different elements of models cross-reference to each other. This complex knowledge graph can extract more insightful graphs like those discussed in Chapter 5. ALEF uses three models: the objectmodel, the rulemodel, and the servicemodel. Each model represents different components of the decision-making process:

- The objectmodel represents entities and their attributes involved in decision-making.
- The rulemodel represents rules and their dependencies for making decisions.
- The servicemodel represents services used by legal professionals for decision-making by providing input and output variables.

Variables play a central role in the decision-making process and are the different models' overlapping factors. Hence, a clear representation of variables across all models could aid in understanding the decision-making process. For example, variables represent the attributes of objects in the objectmodel, the input and output fields of services in the servicemodel are mapped onto variables, and the rules in the rulemodel use variables in their conditions or calculations or to derive new variable values.

6.2.3 Relation between legal concepts and models

Earlier, the Legal Analysis Diagram (see Figure 4) was presented, which provides a framework for understanding the components of legal decisions. We can relate these legal concepts in the diagram to the elements in the models used by the decision-making system.

For example, the objectmodel represents legal subjects, objects, and their relationships. Legal subjects and objects correspond to the object nodes in the objectmodel, while the connections between these nodes can represent legal relationships. The rulemodel can represent conditions, derivation rules, and requirements that must be fulfilled. In addition, operators can also be represented as rules in the rulemodel. Rule nodes correspond to specific rules or conditions, while variable-type nodes in the rulemodel represent the variables used by these rules. At last, the servicemodel can represent services and their input and output variables. Service nodes correspond to specific services used during the decision-making process, while variable type nodes in the servicemodel represent input and output variables for these services.

Note that not all the diagram elements are represented and captured in the models. For example, delegation power and delegation elaboration are not included. However, the modeller can add other relevant information. For example, they can set the sources of rules. This added information is extracted from the model elements and represented as properties in the summarising graph.

6.2.4 Relation between legal concepts and models

The summarising graphs for each model will consist of nodes and relationships representing the legal concepts associated with each model. Figure 7 shows how these nodes and relationships are represented in the Neo4j graphs. In these graphs, the colours of the nodes and relationships have been matched to the colours used in the Legal Analysis Diagram, ensuring a clear and consistent representation of the components of the decision-making process. Furthermore, as these nodes are part of the Abstract Syntax Tree underlying the models, the names of the nodes match the concepts of the used Domain-Specific Language. However, the relationships between these nodes are created as part of the summarising process and use names closer to the Legal Analysis Diagram.

The objectmodel will consist of variables and object types with "relates_to" relationships between object types. In the decision-making system, a distinction is made between boolean features (or "Kenmerken") and attributes (or "Attributen") of objects. However, in line with the diagram, we will refer to them as variables (or "Variabelen"). The servicemodel will consist of input messages (or "InvoerBerichtType") and output messages (or "UitvoerBerichtType") with input and output relationships to the connected variables. Finally, the rulemodel will consist of rules with calculation, derivation, and condition relationships from and towards the corresponding variables.

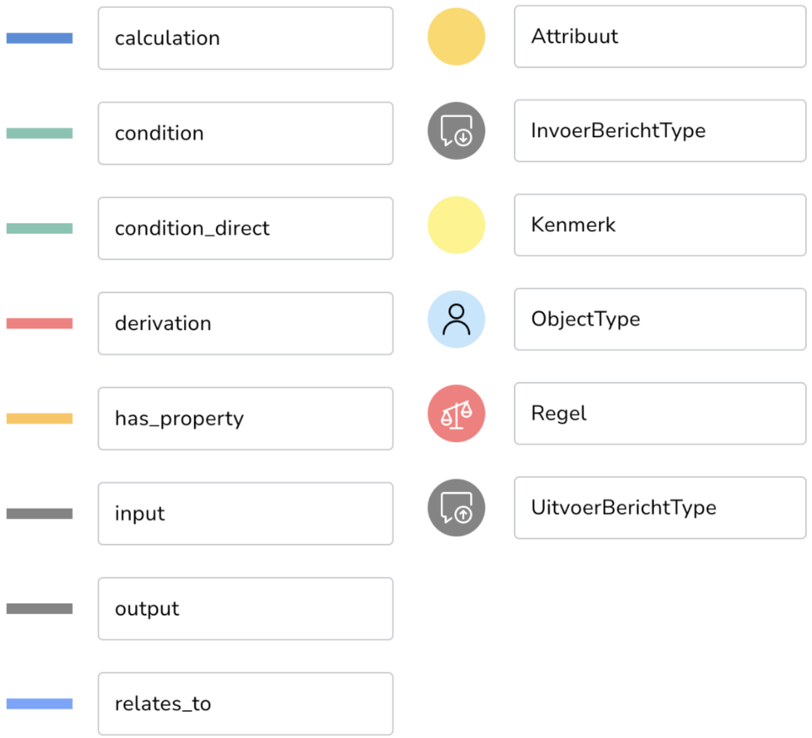


Figure 7

Legenda of how the nodes and relationships are represented in the Neo4j graphs.

7 Application of the explanation method to a specific case

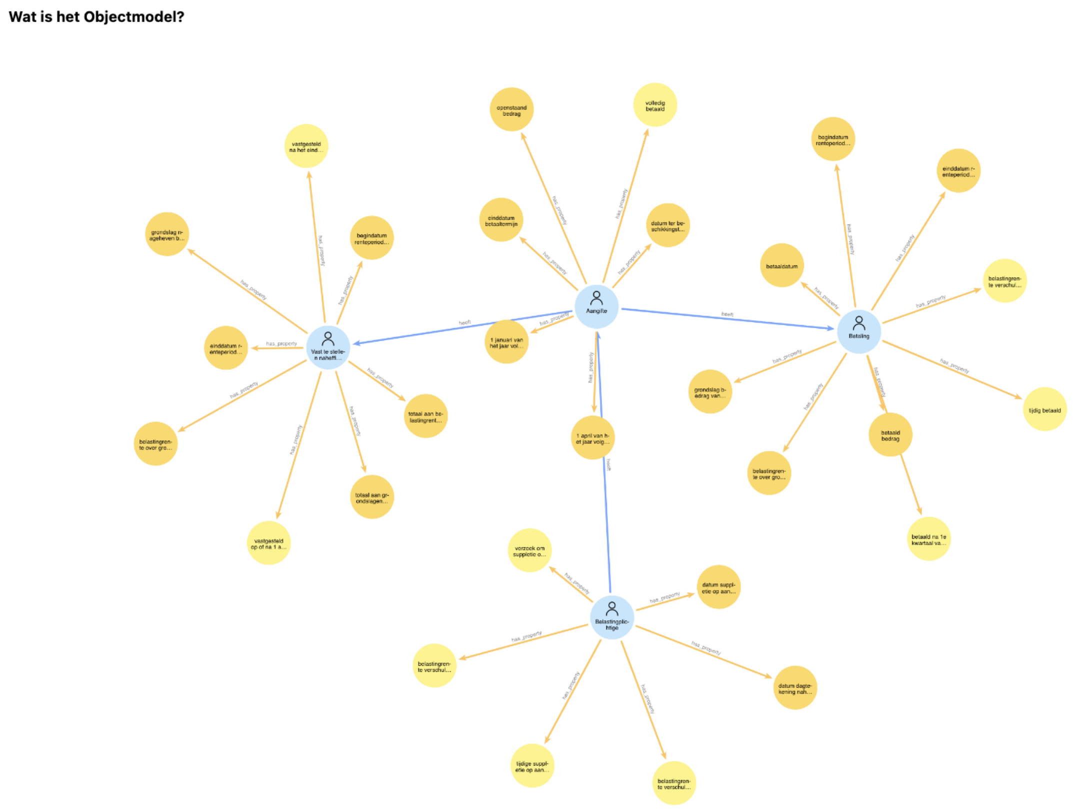
This section outlines the questions the explanation method considered and references figures that illustrate the answers to these questions. The description of each explanation is provided in the caption to prevent switching back and forth between the text and figures.

7.1 Model experts - creation and verification

7.1.1 Explanation for creation

The process of model creation is fundamental for constructing a rule-based decision-making system. The modeller explanation offers guidance in this modelling process by providing a visual representation of each model to aid the textual editor. Hence, the explanation method answers the following questions:

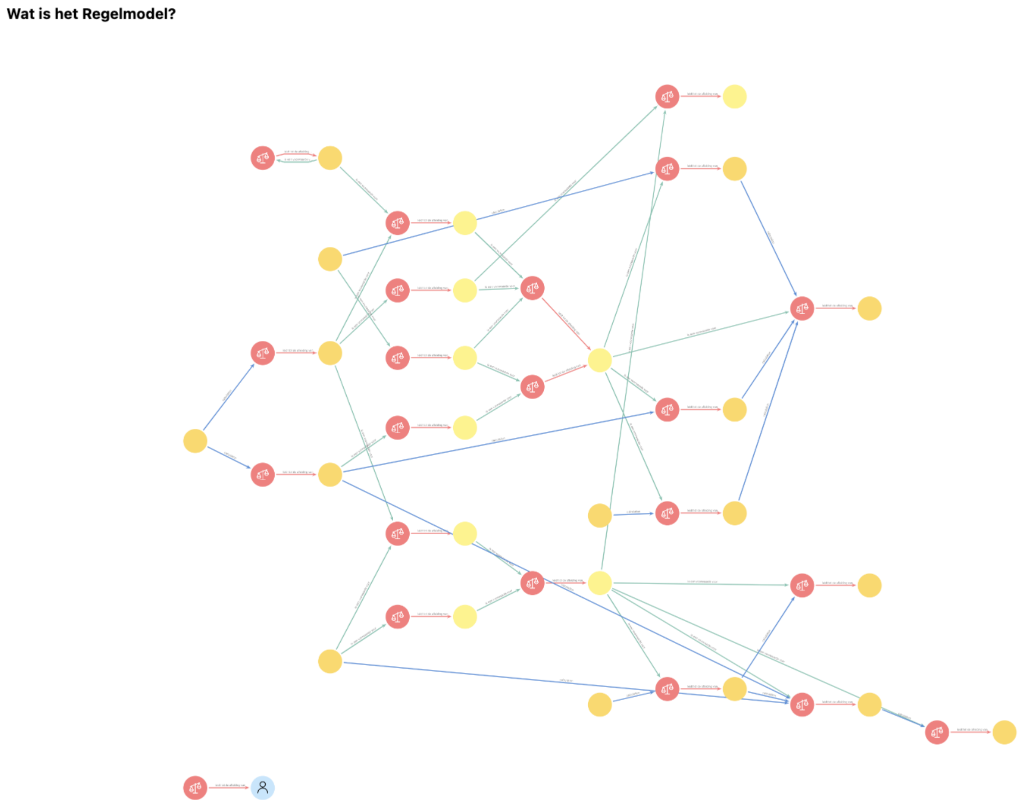
- Figure 8: Answer to *"What elements has the Objectmodel?"*
- Figure 9: Answer to *"What elements has the Rulemodel?"*
- Figure 10: Answer to *"What elements has the Servicemodel?"*



Wat is het Objectmodel?

Figure 8

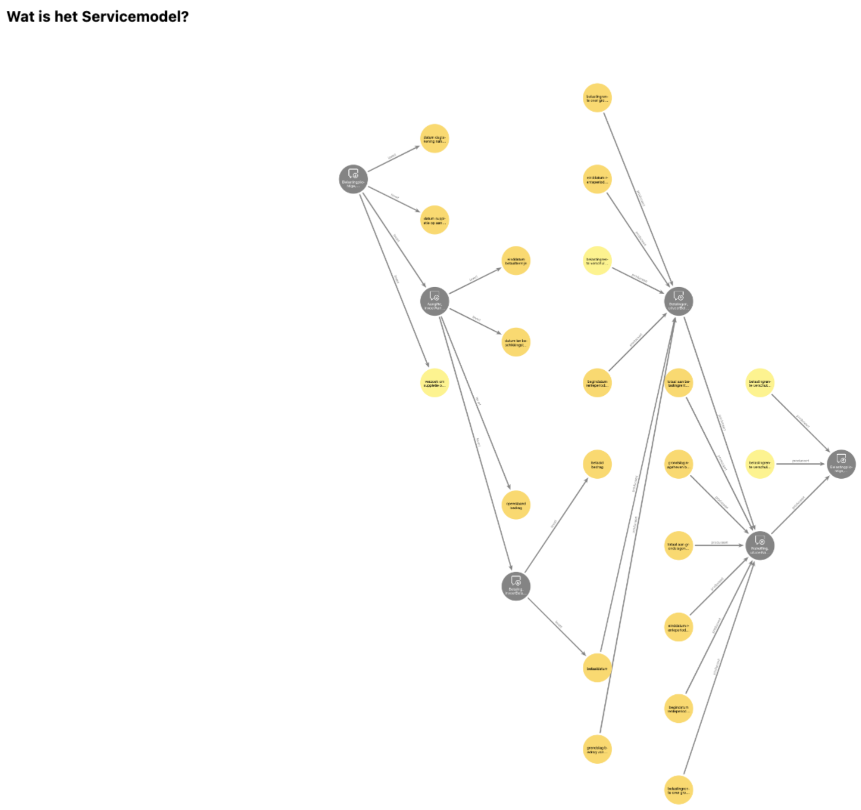
Part of the model expert explanation and answer to "What elements has the Objectmodel?" (translation of Dutch question given on top left corner). This answer shows the variable (yellow) and object (light blue) nodes, as well as the property relationships from an object to a variable (yellow) and the relationships between objects (blue). This explanation is a visual aid for the model expert who designs this model in a textual editor.



Wat is het Regelmodel?

Figure 9

Part of the model expert explanation and answer to "What elements has the Rulemodel?" (translation of Dutch question given on top left corner). This answer shows the variable (yellow), rule (red) and object (light blue) nodes, as well as the condition relationships from a variable to a rule (green), the calculation relationships from a variable to a rule (blue) and the derivation relationships from a rule to a variable or object (red). This explanation is a visual aid for the model expert who designs this model in a textual editor.



Wat is het Servicemodel?

Figure 10

Part of the model expert explanation and answer to "What elements has the Servicemodel?" (translation of Dutch question given on top left corner). This answer shows the variable (yellow), the input message (dark grey) and the output message (dark grey) nodes, as well as the input relationships from an input message to a variable (dark grey) and the output relationships from a variable to an output message (dark grey). This explanation is a visual aid for the model expert who designs this model in a textual editor.

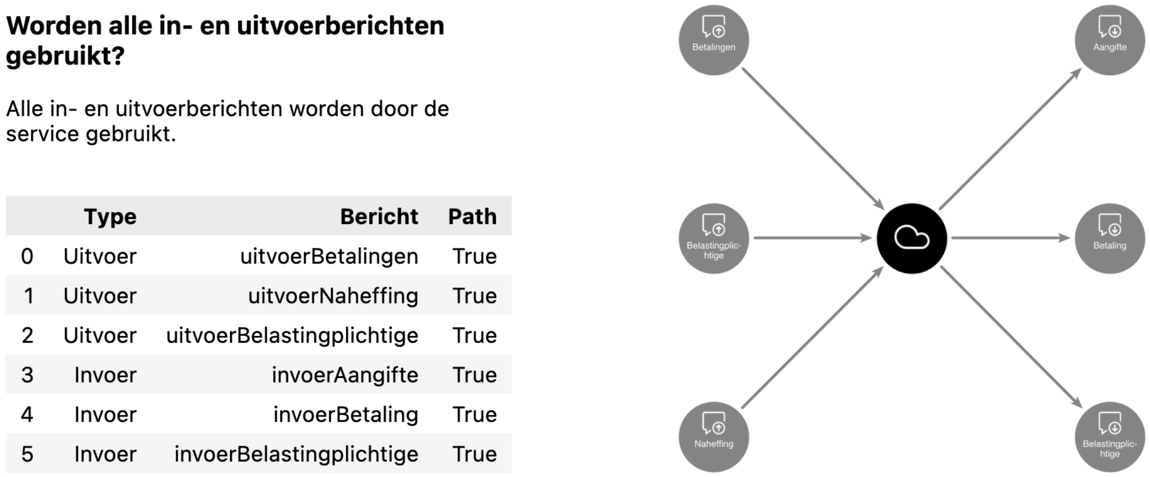
7.1.2 Explanation for verification

The verification process for a rule-based decision-making system is essential to ensure its proper functioning. Hence, the explanation for the model expert should help guide the verification process. For this process, a distinction is made between three types of checks:

- Path Checks: These checks verify if the system utilises an element and help identify redundant elements.
- Assignment Checks: These checks ensure an element is assigned a value, confirming that these elements can provide results.
- Logical checks: These checks confirm a rule's absence of logical contradictions in its conditions and help identify any inconsistencies.

This distinction generalises the verification documentation for the ALEF system used in the case study. However, these different checks are likely applicable to other decision-making systems. The same documentation states several questions a modeller should answer themselves to verify the models they made. Hence, the explanation for the model expert answers these questions:

- Figure 11: Answer to *"Is each in- and output message used by the Service?"* (Path check)
- Figure 12: Answer to *"Is all input used to create the output?"* and *"Can all output be created, given the input?"*(Path checks)
- Figure 13: Answer to *"Is each variable used?"* (Path check)
- Figure 14: Answer to *"Are all variables assigned?"* (Assignment check)



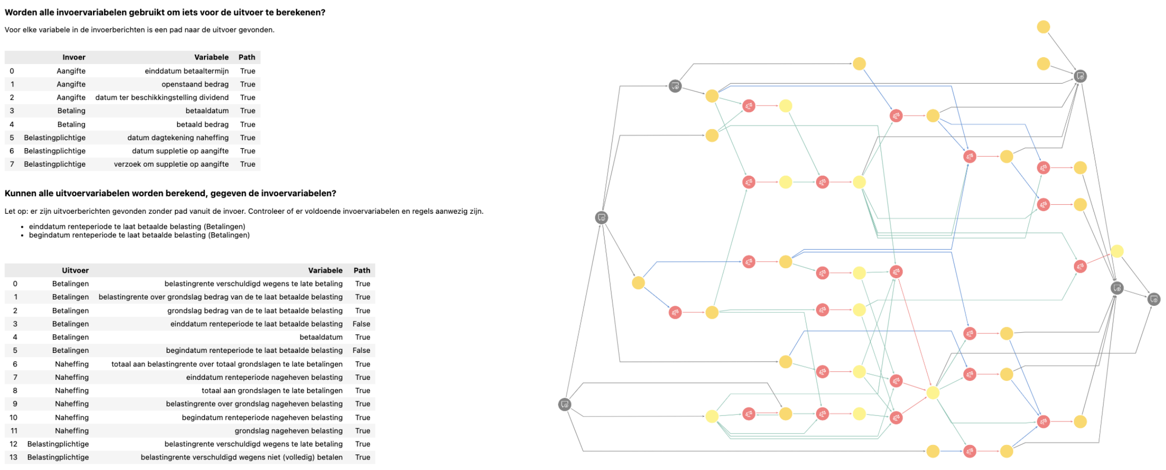
Worden alle in- en uitvoerberichten gebruikt?

Alle in- en uitvoerberichten worden door de service gebruikt.

	Type	Bericht	Path
0	Uitvoer	uitvoerBetalingen	True
1	Uitvoer	uitvoerNaheffing	True
2	Uitvoer	uitvoerBelastingplichtige	True
3	Invoer	invoerAangifte	True
4	Invoer	invoerBetaling	True
5	Invoer	invoerBelastingplichtige	True

Figure 11

Part of the model expert explanation and answer to "Is each in- and output message used by the Service?" (translation of Dutch question given on top left corner). This answer contains an error message, noting that "The service uses all in- and output messages.", and a table showing, for each message, whether it is used. This textual answer is aided by a visualisation which shows the service (black), the input message (dark grey) and the output message (dark grey) nodes, as well as the output relationships from an output message to the service (dark grey) and the input relationships from a service to an input message (dark grey).



Worden alle invoervariabelen gebruikt om iets voor de uitvoer te berekenen?

Voor elke variabele in de invoerberichten is een pad naar de uitvoer gevonden.

	Invoer	Variabele	Path
0	Aangifte	einddatum betaaltermijn	True
1	Aangifte	openstaand bedrag	True
2	Aangifte	datum ter beschikkingstelling dividend	True
3	Betaling	betaaldatum	True
4	Betaling	betaald bedrag	True
5	Belastingplichtige	datum dagtekening naheffing	True
6	Belastingplichtige	datum suppletie op aangifte	True
7	Belastingplichtige	verzoek om suppletie op aangifte	True

Kunnen alle uitvoervariabelen worden berekend, gegeven de invoervariabelen?

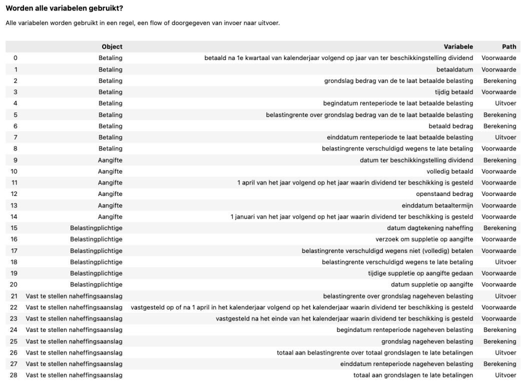
Let op: er zijn uitvoerberichten gevonden zonder pad vanuit de invoer. Controleer of er voldoende invoervariabelen en regels aanwezig zijn.

- einddatum renteperiode te laat betaalde belasting (Betalingen)
- begindatum renteperiode te laat betaalde belasting (Betalingen)

	Uitvoer	Variabele	Path
0	Betalingen	belastingrente verschuldigd wegens te late betaling	True
1	Betalingen	belastingrente over grondslag bedrag van de te laat betaalde belasting	True
2	Betalingen	grondslag bedrag van de te laat betaalde belasting	True
3	Betalingen	einddatum renteperiode te laat betaalde belasting	False
4	Betalingen	betaaldatum	True
5	Betalingen	begindatum renteperiode te laat betaalde belasting	False
6	Naheffing	totaal aan belastingrente over totaal grondslagen te late betalingen	True
7	Naheffing	einddatum renteperiode nageheven belasting	True
8	Naheffing	totaal aan grondslagen te late betalingen	True
9	Naheffing	belastingrente over grondslag nageheven belasting	True
10	Naheffing	begindatum renteperiode nageheven belasting	True
11	Naheffing	grondslag nageheven belasting	True
12	Belastingplichtige	belastingrente verschuldigd wegens te late betaling	True
13	Belastingplichtige	belastingrente verschuldigd wegens niet (volledig) betalen	True

Figure 12

Part of the model expert explanation and answer to two questions, namely "Is all input used to create the output?" and "Can all output be created, given the input?" (translation of Dutch questions given in bold font). The first answer contains an error message, noting that "For each variable in the input messages, a path to the output has been found." and a table showing, for each variable, whether a path is found from this variable to an output message. The second answer contains an error message, noting that "Note: output messages have been found without a path from the input. Check if there are enough input variables and rules present." with a list of the respective variables and a table showing, for each variable, whether a path is found from an input message to this variable. These textual answers are aided by a visualisation which shows the input and output message (dark grey), variable (yellow) and rule (red) nodes, as well as the input relationships from an input message to a variable (dark grey), the condition relationships from a variable to a rule (green), the calculation relationships from a variable to a rule (blue), the derivation relationships from a rule to a variable or object (red) and the output relationships from a variable to an output message (dark grey). The visualisation shows the two output variables without a path in the top right corner. This explanation is a textual and visual aid for the model expert who verifies the models within a service. This explanation is a textual and visual aid for the model expert who verifies the models within a service. See Figure 17 for an expanded version of this explanation.



Worden alle variabelen gebruikt?

Alle variabelen worden gebruikt in een regel, een flow of doorgegeven van invoer naar uitvoer.

	Object	Variabele	Path
0	Betaling	betaald na 1e kwartaal van kalenderjaar volgend op jaar van ter beschikkingstelling dividend	Voorwaarde
1	Betaling	betaaldatum	Voorwaarde
2	Betaling	grondslag bedrag van de te laat betaalde belasting	Berekening
3	Betaling	tijdig betaald	Voorwaarde
4	Betaling	begindatum renteperiode te laat betaalde belasting	Uitvoer
5	Betaling	belastingrente over grondslag bedrag van de te laat betaalde belasting	Berekening
6	Betaling	betaald bedrag	Berekening
7	Betaling	einddatum renteperiode te laat betaalde belasting	Uitvoer
8	Betaling	belastingrente verschuldigd wegens te late betaling	Voorwaarde
9	Aangifte	datum ter beschikkingstelling dividend	Berekening
10	Aangifte	volledig betaald	Voorwaarde
11	Aangifte	1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Voorwaarde
12	Aangifte	openstaand bedrag	Voorwaarde
13	Aangifte	einddatum betaaltermijn	Voorwaarde
14	Aangifte	1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Voorwaarde
15	Belastingplichtige	datum dagtekening naheffing	Berekening
16	Belastingplichtige	verzoek om suppletie op aangifte	Voorwaarde
17	Belastingplichtige	belastingrente verschuldigd wegens niet (volledig) betalen	Voorwaarde
18	Belastingplichtige	belastingrente verschuldigd wegens te late betaling	Uitvoer
19	Belastingplichtige	tijdige suppletie op aangifte gedaan	Voorwaarde
20	Belastingplichtige	datum suppletie op aangifte	Voorwaarde
21	Vast te stellen naheffingsaanslag	belastingrente over grondslag nageheven belasting	Uitvoer
22	Vast te stellen naheffingsaanslag	vastgesteld op of na 1 april in het kalenderjaar volgend op het kalenderjaar waarin dividend ter beschikking is gesteld	Voorwaarde
23	Vast te stellen naheffingsaanslag	vastgesteld na het einde van het kalenderjaar waarin dividend ter beschikking is gesteld	Voorwaarde
24	Vast te stellen naheffingsaanslag	begindatum renteperiode nageheven belasting	Berekening
25	Vast te stellen naheffingsaanslag	grondslag nageheven belasting	Berekening
26	Vast te stellen naheffingsaanslag	totaal aan belastingrente over totaal grondslagen te late betalingen	Uitvoer
27	Vast te stellen naheffingsaanslag	einddatum renteperiode nageheven belasting	Berekening
28	Vast te stellen naheffingsaanslag	totaal aan grondslagen te late betalingen	Uitvoer

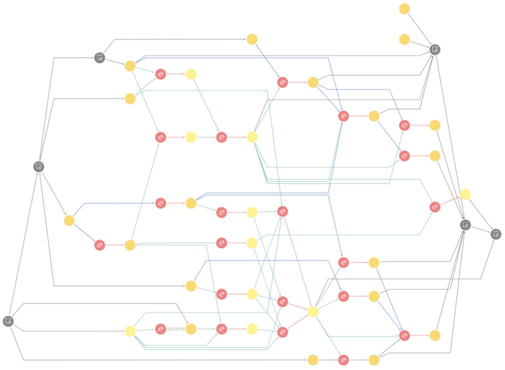
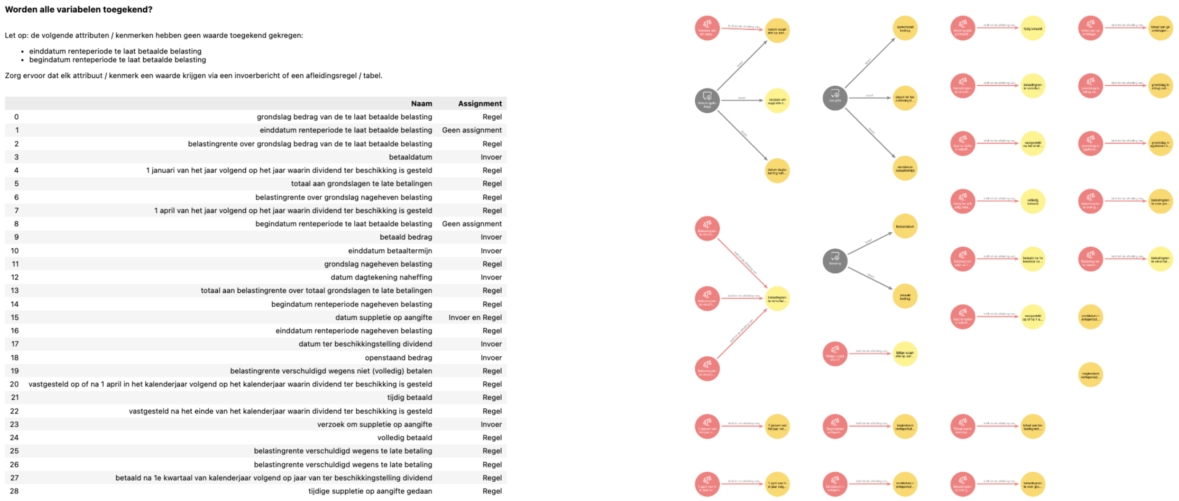


Figure 13

Part of the model expert explanation and answer to "Is each variable used?" (translation of Dutch question given on top left corner). This answer contains an error message, noting that "All variables are used in a rule, a flow, or passed from input to output.", and a table showing whether each variable is used. This textual answer is aided by a visualisation which shows the input and output message (dark grey), variable (yellow) and rule (red) nodes, as well as the input relationships from an input message to a variable (dark grey), the condition relationships from a variable to a rule (green), the calculation relationships from a variable to a rule (blue), the derivation relationships from a rule to a variable or object (red) and the output relationships from a variable to an output message (dark grey). This explanation is a textual and visual aid for the model expert who verifies the models within a service. See Figure 18 for an expanded version of this explanation.



Worden alle variabelen toegekend?

Let op: de volgende attributen / kenmerken hebben geen waarde toegekend gekregen:

- einddatum renteperiode te laat betaalde belasting
- begindatum renteperiode te laat betaalde belasting

Zorg ervoor dat elk attribuut / kenmerk een waarde krijgen via een invoerbericht of een afleidingsregel / tabel.

	Naam	Assignment
0	grondslag bedrag van de te laat betaalde belasting	Regel
1	einddatum renteperiode te laat betaalde belasting	Geen assignment
2	belastingrente over grondslag bedrag van de te laat betaalde belasting	Regel
3	betaaldatum	Invoer
4	1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Regel
5	totaal aan grondslagen te late betalingen	Regel
6	belastingrente over grondslag nageheven belasting	Regel
7	1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Regel
8	begindatum renteperiode te laat betaalde belasting	Geen assignment
9	betaald bedrag	Invoer
10	einddatum betaaltermijn	Invoer
11	grondslag nageheven belasting	Regel
12	datum dagtekening naheffing	Invoer
13	totaal aan belastingrente over totaal grondslagen te late betalingen	Regel
14	begindatum renteperiode nageheven belasting	Regel
15	datum suppletie op aangifte	Invoer en Regel
16	einddatum renteperiode nageheven belasting	Regel
17	datum ter beschikkingstelling dividend	Invoer
18	openstaand bedrag	Invoer
19	belastingrente verschuldigd wegens niet (volledig) betalen	Regel
20	vastgesteld op of na 1 april in het kalenderjaar volgend op het kalenderjaar waarin dividend ter beschikking is gesteld	Regel
21	tijdig betaald	Regel
22	vastgesteld na het einde van het kalenderjaar waarin dividend ter beschikking is gesteld	Regel
23	verzoek om suppletie op aangifte	Invoer
24	volledig betaald	Regel
25	belastingrente verschuldigd wegens te late betaling	Regel
26	belastingrente verschuldigd wegens te late betaling	Regel
27	betaald na 1e kwartaal van kalenderjaar volgend op jaar van ter beschikkingstelling dividend	Regel
28	tijdige suppletie op aangifte gedaan	Regel

Figure 14

Part of the model expert explanation and answer to "Are all variables assigned?" (translation of Dutch question given on top left corner). This answer contains an error message, noting that "Note: the following attributes/characteristics have not been assigned a value: list of values , Ensure that each attribute/characteristic receives a value through an input message or a deduction rule/table.", and table showing whether and how each variable is assigned. This textual answer is aided by a visualisation which shows the input message (dark grey), variable (yellow) and rule (red) nodes, as well as the input relationships from an input message to a variable (dark grey) and the derivation relationships from a rule to a variable (red). The visualisation shows the two variables without assignment on the bottom right. This explanation is a textual and visual aid for the model expert who verifies the models within a service. See Figure 19 for an expanded version of this explanation.

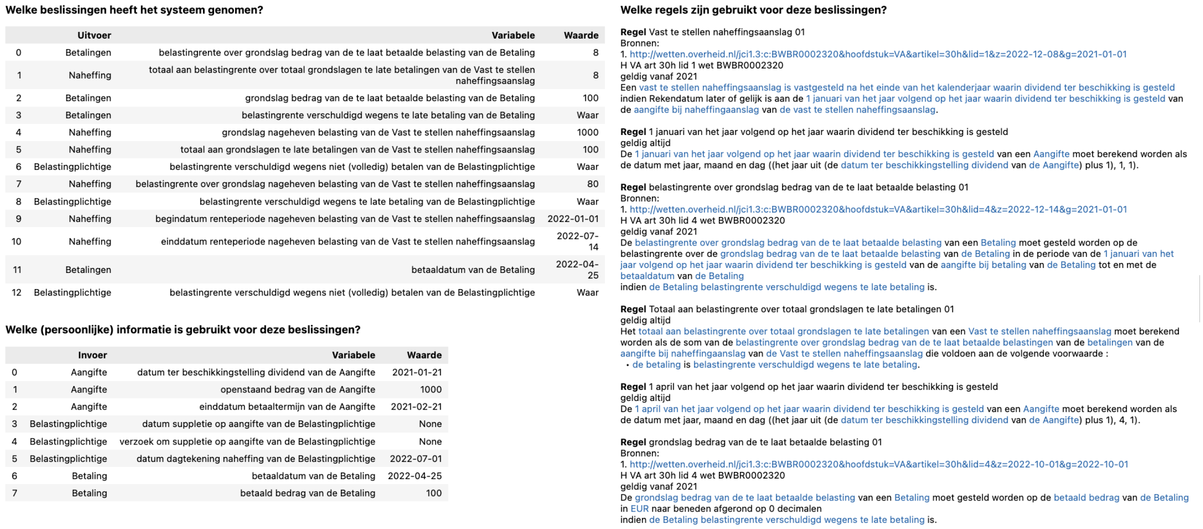
7.2 Legal support staff - explanation

Legal support professionals work within customer service or other relevant departments to provide legal support and explanation to data subjects regarding the decisions made. Hence, they need to be able to answer relevant questions for a data subject. For these questions, a distinction is made between two types (matching two of the explanation types made by ICO (Information Commissioners Office & Alan Turing Institute, n.d.)):

- Data explanation: explains the data used in a decision-making system.
- Rationale explanation: explains the reasoning behind a decision-making system.

Accordingly, an explanation addressed to such a professional should consider the following questions, given a certain decision:

- Figure 15: Answer to *"What decisions did the system make?"*, *"What (personal) information was used for these decisions?"* and *"Which rules were used for these decisions?"* (Data)
- Figure 16: Answer to *"Why is this decision made?"* and *"Why is this decision made? (trace)"* (Rationale)



Welke beslissingen heeft het systeem genomen?

	Uitvoer	Variabele	Waarde
0	Betalingen	belastingrente over grondslag bedrag van de te laat betaalde belasting van de Betaling	8
1	Naheffing	totaal aan belastingrente over totaal grondslagen te late betalingen van de Vast te stellen naheffingsaanslag	8
2	Betalingen	grondslag bedrag van de te laat betaalde belasting van de Betaling	100
3	Betalingen	belastingrente verschuldigd wegens te late betaling van de Betaling	Waar
4	Naheffing	grondslag nageheven belasting van de Vast te stellen naheffingsaanslag	1000
5	Naheffing	totaal aan grondslagen te late betalingen van de Vast te stellen naheffingsaanslag	100
6	Belastingplichtige	belastingrente verschuldigd wegens niet (volledig) betalen van de Belastingplichtige	Waar
7	Naheffing	belastingrente over grondslag nageheven belasting van de Vast te stellen naheffingsaanslag	80
8	Belastingplichtige	belastingrente verschuldigd wegens te late betaling van de Belastingplichtige	Waar
9	Naheffing	begindatum renteperiode nageheven belasting van de Vast te stellen naheffingsaanslag	2022-01-01
10	Naheffing	einddatum renteperiode nageheven belasting van de Vast te stellen naheffingsaanslag	2022-07-14
11	Betalingen	betaaldatum van de Betaling	2022-04-25
12	Belastingplichtige	belastingrente verschuldigd wegens niet (volledig) betalen van de Belastingplichtige	Waar

Welke (persoonlijke) informatie is gebruikt voor deze beslissingen?

	Invoer	Variabele	Waarde
0	Aangifte	datum ter beschikkingstelling dividend van de Aangifte	2021-01-21
1	Aangifte	openstaand bedrag van de Aangifte	1000
2	Aangifte	einddatum betaaltermijn van de Aangifte	2021-02-21
3	Belastingplichtige	datum suppletie op aangifte van de Belastingplichtige	None
4	Belastingplichtige	verzoek om suppletie op aangifte van de Belastingplichtige	None
5	Belastingplichtige	datum dagtekening naheffing van de Belastingplichtige	2022-07-01
6	Betaling	betaaldatum van de Betaling	2022-04-25
7	Betaling	betaald bedrag van de Betaling	100

Welke regels zijn gebruikt voor deze beslissingen?

Regel Vast te stellen naheffingsaanslag 01
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=1&z=2022-12-08&g=2021-01-01
H VA art 30h lid 1 wet BWBR0002320
geldig vanaf 2021
Een vast te stellen naheffingsaanslag is vastgesteld na het einde van het kalenderjaar waarin dividend ter beschikking is gesteld indien Rekendatum later of gelijk is aan de 1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld van de aangifte bij naheffingaanslag van de vast te stellen naheffingsaanslag.

Regel 1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld
geldig altijd
De 1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld van een Aangifte moet berekend worden als de datum met jaar, maand en dag ((het jaar uit (de datum ter beschikkingstelling dividend van de Aangifte) plus 1), 1, 1).

Regel belastingrente over grondslag bedrag van de te laat betaalde belasting 01
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=4&z=2022-12-14&g=2021-01-01
H VA art 30h lid 4 wet BWBR0002320
geldig vanaf 2021
De belastingrente over grondslag bedrag van de te laat betaalde belasting van een Betaling moet gesteld worden op de belastingrente over de grondslag bedrag van de te laat betaalde belasting van de Betaling in de periode van de 1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld van de aangifte bij betaling van de Betaling tot en met de betaaldatum van de Betaling
indien de Betaling belastingrente verschuldigd wegens te late betaling is.

Regel Totaal aan belastingrente over totaal grondslagen te late betalingen 01
geldig altijd
Het totaal aan belastingrente over totaal grondslagen te late betalingen van een Vast te stellen naheffingsaanslag moet berekend worden als de som van de belastingrente over grondslag bedrag van de te laat betaalde belastingen van de betalingen van de aangifte bij naheffingaanslag van de Vast te stellen naheffingsaanslag die voldoen aan de volgende voorwaarde :

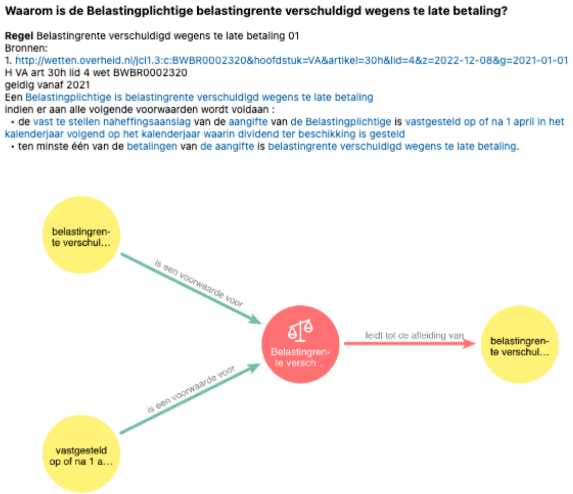
- de betaling is belastingrente verschuldigd wegens te late betaling.

Regel 1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld
geldig altijd
De 1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld van een Aangifte moet berekend worden als de datum met jaar, maand en dag ((het jaar uit (de datum ter beschikkingstelling dividend van de Aangifte) plus 1), 4, 1).

Regel grondslag bedrag van de te laat betaalde belasting 01
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=4&z=2022-10-01&g=2022-10-01
H VA art 30h lid 4 wet BWBR0002320
geldig vanaf 2021
De grondslag bedrag van de te laat betaalde belasting van een Betaling moet gesteld worden op de betaald bedrag van de Betaling in EUR naar beneden afgerond op 0 decimalen
indien de Betaling belastingrente verschuldigd wegens te late betaling is.

Figure 15

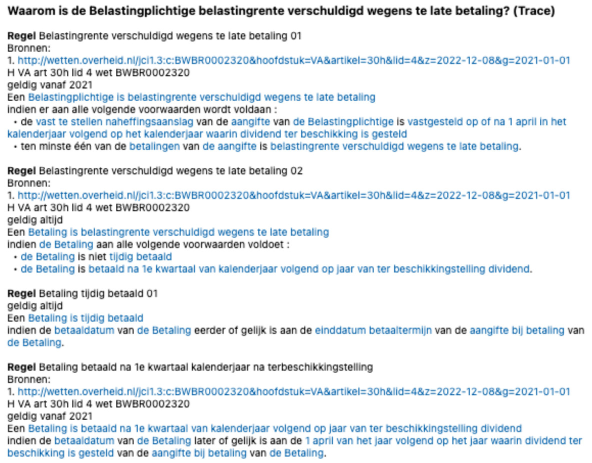
Part of the model expert explanation and answer to ""What decisions did the system make?", ""What (personal) information was used for these decisions?" and ""Which rules were used for these decisions?" (translation of Dutch question given bold text). The first answer displays a table containing the output message, variable and assigned values. The second answer displays a table with the input message, variable and given values. The third answer displays all rules used for these decisions (only six are included in this figure to prevent a very large figure, but this answer could return all as this number is manually adjustable).



Waarom is de Belastingplichtige belastingrente verschuldigd wegens te late betaling?

Regel Belastingrente verschuldigd wegens te late betaling 01
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=4&z=2022-12-08&g=2021-01-01
H VA art 30h lid 4 wet BWBR0002320
geldig vanaf 2021
Een Belastingplichtige is belastingrente verschuldigd wegens te late betaling
indien er aan alle volgende voorwaarden wordt voldaan :

- de vast te stellen naheffingsaanslag van de aangifte van de Belastingplichtige is vastgesteld op of na 1 april in het kalenderjaar volgend op het kalenderjaar waarin dividend ter beschikking is gesteld
- ten minste één van de betalingen van de aangifte is belastingrente verschuldigd wegens te late betaling.



Waarom is de Belastingplichtige belastingrente verschuldigd wegens te late betaling? (Trace)

Regel Belastingrente verschuldigd wegens te late betaling 01
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=4&z=2022-12-08&g=2021-01-01
H VA art 30h lid 4 wet BWBR0002320
geldig vanaf 2021
Een Belastingplichtige is belastingrente verschuldigd wegens te late betaling
indien er aan alle volgende voorwaarden wordt voldaan :

- de vast te stellen naheffingsaanslag van de aangifte van de Belastingplichtige is vastgesteld op of na 1 april in het kalenderjaar volgend op het kalenderjaar waarin dividend ter beschikking is gesteld
- ten minste één van de betalingen van de aangifte is belastingrente verschuldigd wegens te late betaling.

Regel Belastingrente verschuldigd wegens te late betaling 02
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=4&z=2022-12-08&g=2021-01-01
H VA art 30h lid 4 wet BWBR0002320
geldig altijd
Een Betaling is belastingrente verschuldigd wegens te late betaling
indien de Betaling aan alle volgende voorwaarden voldoet :

- de Betaling is niet tijdig betaald
- de Betaling is betaald na 1e kwartaal van kalenderjaar volgend op jaar van ter beschikkingstelling dividend.

Regel Betaling tijdig betaald 01
geldig altijd
Een Betaling is tijdig betaald
indien de betaaldatum van de Betaling eerder of gelijk is aan de einddatum betaaltermijn van de aangifte bij betaling van de Betaling.

Regel Betaling betaald na 1e kwartaal kalenderjaar na terbeschikkingstelling
Bronnen:
1. http://wetten.overheid.nl/jci1.3:c:BWBR0002320&hoofdstuk=VA&artikel=30h&lid=4&z=2022-12-08&g=2021-01-01
H VA art 30h lid 4 wet BWBR0002320
geldig vanaf 2021
Een Betaling is betaald na 1e kwartaal van kalenderjaar volgend op jaar van ter beschikkingstelling dividend
indien de betaaldatum van de Betaling later of gelijk is aan de 1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld van de aangifte bij betaling van de Betaling.

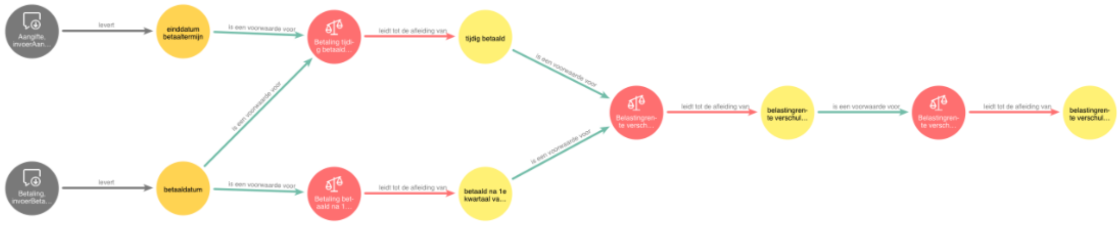


Figure 16

Part of the support staff explanation and answer to "Why is this decision made?" and "Why is this decision made? (trace)" (translation of Dutch question given bold text). Note: both questions consider the same decision (a variable), but this variable could be manually adjusted. The first answer shows a textual description of the rule and conditions that are met, aided by a visualisation which shows the variable representing the decision (right yellow node), the rule that determined this variable (red) and the derivation relationship from this rule towards the variable. Besides, it displays the conditional variables (left yellow nodes) and the conditional relationships from these variables towards the rule (green). The second answer shows a textual description of the rules and conditions that are met like a trace (the first rule is the one that is closest to the decision variable. Similar to the first question, this answer is aided with a visualisation that display the variables, rules and their relationships, traced back towards the input messages (dark grey).

8 Evaluation of the explanation method

8.1 Evaluation of generated explanations

The proposed explanation method's effectiveness can be evaluated based on its ability to answer a set of questions for both target groups.

Regarding the modeller, it was found that the explanation method can answer questions related to path and assignment checks. However, it cannot perform logical checks to ensure that there are no logical contradictions or omissions in a rule. Although the explanation method can alert the modeller to potential contradictions, it cannot compare two conditions, as it lacks the ability to reason independently.

Similarly, for the support staff, the explanation method can answer 'what' and 'why' questions. However, it cannot provide answers to 'what if' and 'why not' questions, as this would also require independent reasoning ability. While the current implementation cannot provide all the required answers, it is possible, with future research, to expand the system's capabilities. This would require the explanation method to possess reasoning capacity or a live integration with the decision system.

8.2 Evaluation of the explanation method

The proposed explanation method involves representing a decision model in a graph database to be able to both question (query) and visualise it.

This method gives flexibility in the content category: It makes it possible to question both the decision model and a specific decision (focus), to filter information based on the needs and preferences of the recipients (level of detail) and extract causal relations between conditions, rules and derivations of these rules (causality).

This also gives flexibility in the communication category: for now, we provided user-system interaction by giving answers to predefined questions. However, we argue that (by adding another explanatory layer) the explanation could also be presented as a menu-driven explanation or even natural language dialogue (by building a question base). Besides, the explanation is now multi-media. However, we argue that this can be changed to only text if deemed more fitting for a specific recipient.

This also gives flexibility in the adaptation category: We used a static model to adapt to a specific recipient by considering different target groups. For each target group, we assumed a certain level of knowledge (both for the domain and reasoning knowledge) and specified a goal. Then, we linked the specified goal to a specific set of questions that should be answered by the explanation method.

9 Conclusion and Discussion

9.1 Key findings

The main objective of this research was to develop a human-centred explanation method for rule-based automated decision-making systems in the legal domain. To achieve this objective, the study was divided into theoretical and practical research.

In the theoretical research, we explored the concept of explainability in AI and proposed a framework for developing explanation methods that considered the content, communication, and adaptation categories. We also delved deeper into research on human-centred explanations, explanation techniques for rule-based systems, and explanations within the legal domain. This exploration revealed that creating human-centred explanations in explainable AI is a complex task that requires a highly flexible method that can cater to the unique needs of each target group, considering both their knowledge and their goals. Based on this exploration, we proposed a human-centred explanation method of rule-based decision-making systems in the legal domain which involves method involves representing the elements of a decision model in a graph database, which allows for flexibility in the content, communication, and adaptation categories.

In the practical research, we implemented the proposed explanation method for a real-world scenario: an automated decision-making system used by the Dutch Tax and Customs Administration. Our results showed that the explanation method could answer questions regarding the service checks, path, and assignment checks but could not provide logical checks that ensure no logical contradictions or omissions in a rule. Additionally, we found that the explanation method could answer the 'what' and 'why' questions for the support staff but could not answer the 'what if' and 'why not' questions.

9.2 Main contributions

The main contribution of this research is the proposed human-centred explanation method for rule-based automated decision-making systems in the legal domain. This method gives flexibility in the content, communication, and adaptation categories, making it possible to adjust the level of detail and causal relationships between the different elements of a decision model, provide multi-media explanations, and tailor the explanations to the recipient's knowledge and goals.

Additionally, this research contributes to the wider discussion on explainability in AI. By identifying the components of an explanation method and external factors that should be considered when developing it in a specific context, we provide a framework for other researchers to use when developing their own explanation methods.

9.3 Future research

From evaluating the decision method, we revealed that future research could explore ways to enable the explanation method to reason independently or create a live integration with the decision system. In addition to evaluating the decision method, we could also lay the explanation properties alongside the explanation method requirements in Table 3 to reveal possible areas of future research. The explanation properties can be viewed as a checklist of properties that the explanation method should possess, categorised into Content, Communication, and Adaptation.

For Content, we have found that the explanation method can explain both the global and local models of a decision, and we can adjust the level of detail using filtering techniques. We can also use path-finding to highlight causal relationships between decision elements and reveal decision steps chronologically.

For Communication, we currently provide user-system interaction by answering predefined questions. However, we suggest that adding another explanatory layer could enable menu-driven or natural language dialogue-based explanations. The current explanation is multi-media, but we can provide text-only explanations if deemed more fitting for a specific individual.

For Adaptation, we determined the knowledge and goals of recipients based on predefined target groups using a static model. However, we argue that it is possible to create more dynamic models that can adapt to someone's knowledge of the domain, reasoning method, and goals. Using a graph database visualisation tool like Bloom, we can create different perspectives for each target group, pre-setting visibility, relationships and their properties, styling, and custom search phrases.

While not implemented, we could integrate Bloom's visualisation with the textual HTML snippets to create a custom website or HTML-based tool per target group, assuming a recipient's goals are based on which target group they belong to. We can link the specified goal to a set of questions that should be answered by the explanation method.

Implementing the explanation method for the Dutch Tax and Customs Administration provides a good starting point for applying the method to other domains. By applying the method to other domains, we can further explore the flexibility of the method and its ability to cater to the unique needs of different target groups. Overall, this research contributes to the ongoing conversation around explainability in AI and the need for human-centred approaches.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, *6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., & Rudin, C. (2017). Learning certifiably optimal rule lists. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 35–44. https://doi.org/10.1145/3097983.3098047

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., & Lopes, C. (2002). Making sense of sensing systems: Five questions for designers and researchers. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 415–422. https://doi.org/10.1145/503376.503450

Bellotti, V., & Edwards, K. (2004). Intelligibility and Accountability: Human Considerations in Context Aware Systems. *Human-Computer Interaction*, *16*. https://doi.org/10.1207/S15327051HCI16234_05

Bucchiarone, A., Cicchetti, A., Ciccozzi, F., & Pierantonio, A. (Eds.). (2021). *Domain-specific languages in practice: With JetBrains MPS*. Springer International Publishing. https://doi.org/10.1007/978-3-030-73758-0

Corsius, M., Hoppenbrouwers, S., Lokin, M., Baars, E., Sangers-Van Cappellen, G., & Wilmont, I. (2021). RegelSpraak: A CNL for executable tax rules specification. *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21)*. Retrieved October 22, 2022, from https://aclanthology.org/2021.cnl-1.6

European Parliament. Directorate General for Parliamentary Research Services. (2019). *Understanding algorithmic decision-making: Opportunities and challenges.* Publications Office. Retrieved September 16, 2022, from https://data.europa.eu/doi/10.2861/536131

GDPR Summary. (2018). Transparency. Retrieved April 20, 2023, from https://www.gdprsummary.com/gdpr-definitions/transparency/

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). Explaining explanations: An overview of interpretability of machine learning [arXiv:1806.00069 [cs, stat]]. Retrieved September 8, 2022, from http://arxiv.org/abs/1806.00069

Goodman, B., & Flaxman, S. (2016). European union regulations on algorithmic decision-making and a "Right to Explanation". https://doi.org/10.1609/aimag.v38i3.2741

Gregor, S., & Benbasat, I. (1999). Explanations from intelligent systems: Theoretical foundations and implications for practice [Publisher: Management Information Systems Research Center, University of Minnesota]. *MIS Quarterly*, *23*(4), 497–530. https://doi.org/10.2307/249487

Hoffman, R. R., Klein, G., & Mueller, S. T. (2018). Explaining explanation for "explainable AI". *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *62*(1), 197–201. https://doi.org/10.1177/1541931218621047

Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2019). Metrics for explainable AI: Challenges and prospects [arXiv:1812.04608 [cs]].

Hohfeld, W. N. (1913). Some fundamental legal conceptions as applied in judicial reasoning [Publisher: The Yale Law Journal Company, Inc.]. *The Yale Law Journal*, *23*(1), 16–59. https://doi.org/10.2307/785533

Hohfeld, W. N. (1917). Fundamental legal conceptions as applied in judicial reasoning [Publisher: The Yale Law Journal Company, Inc.]. *The Yale Law Journal*, *26*(8), 710–770. https://doi.org/10.2307/786270

Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., & Baesens, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, *51*(1), 141–154. https://doi.org/10.1016/j.dss.2010.12.003

Information Commissioners Office & Alan Turing Institute. (n.d.). Explaining decisions made with AI. Retrieved February 14, 2023, from https://ico.org.uk/media/for-organisations/guide-to-data-protection/key-dp-themes/explaining-decisions-made-with-artificial-intelligence-1-0.pdf

JetBrains. (2019). *MPS case study - agile law execution factory* (tech. rep.). Retrieved September 30, 2022, from https://resources.jetbrains.com/storage/products/mps/docs/MPS_DTO_Case_Study.pdf

Lacave, C., & Diez, F. J. (2004). A review of explanation methods for heuristic expert systems [Publisher: Cambridge University Press]. *The Knowledge Engineering Review*, *19*(2), 133–146. https://doi.org/10.1017/S0269888904000190

Lacave, C., & Díez, F. J. (2002). A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, *17*(2), 107–127. https://doi.org/10.1017/S026988890200019X

Liao, Q. V., Gruen, D., & Miller, S. (2020). Questioning the AI: Informing design practices for explainable AI user experiences. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3313831.3376590

Lim, B. Y., & Dey, A. K. (2009). Assessing demand for intelligibility in context-aware applications. *Proceedings of the 11th international conference on Ubiquitous computing*, 195–204. https://doi.org/10.1145/1620545.1620576

Lim, B. Y., & Dey, A. K. (2010). Toolkit to support intelligibility in context-aware applications. *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 13–22. https://doi.org/10.1145/1864349.1864353

Lim, B. Y., Dey, A. K., & Avrahami, D. (2009). Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2119–2128. https://doi.org/10.1145/1518701.1519023

Lipton, P. (1990). Contrastive explanation* [Publisher: Cambridge University Press]. *Royal Institute of Philosophy Supplements*, *27*, 247–266. https://doi.org/10.1017/S1358246100005130

Lokin, M. (2018). *Wendbaar wetgeven: De wetgever als systeembeheerder*.

Lokin, M., Ausems, A., Bulles, J., Constitutional and Administrative Law, Kooijmans Institute, & Public Contract Law. (2021). *Wetsanalyse*. Boom juridisch. Retrieved July 28, 2022, from https://research.vu.nl/en/publications/a4ceb3ea-a940-4af1-934e-58391641a8c8

Lokin, M., & van Kempen, M. (2019). Van wet naar loket: Bedrijfsregels en agile werken voor een transparante wetsuitvoering. *RegelMaat*, *34*(1), 35–57. https://doi.org/10.5553/RM/0920055X2019034001004

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38. https://doi.org/10.1016/j.artint.2018.07.007

Mueller, S. T., Hoffman, R. R., Clancey, W., Emrey, A., & Klein, G. (2019). Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI [arXiv:1902.01876 [cs]]. Retrieved October 31, 2022, from http://arxiv.org/abs/1902.01876

Muir, B. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00140139408964957]. *Ergonomics*, *37*(11), 1905–1922. https://doi.org/10.1080/00140139408964957

Neerincx, M. A., van der Waa, J., Kaptein, F., & van Diggelen, J. (2018). Using perceptual and cognitive explanations for enhanced human-agent team performance. In D. Harris (Ed.), *Engineering Psychology and Cognitive Ergonomics* (pp. 204–214). Springer International Publishing. https://doi.org/10.1007/978-3-319-91122-9_18

Neo4j. (n.d.). What is a graph database? - developer guides. Retrieved February 23, 2023, from https://neo4j.com/developer/graph-database/

Nickerson, R. S. (1998). Confirmation Bias: A Ubiquitous Phenomenon in Many Guises [Publisher: SAGE Publications Inc]. *Review of General Psychology*, *2*(2), 175–220. https://doi.org/10.1037/1089-2680.2.2.175

Open Government Partnership. (2020). *The open government partnership's implementation plan 2020-2022* (tech. rep.). Retrieved August 30, 2022, from https://www.opengovpartnership.org/wp-content/uploads/2020/03/OGPs-Implementation-Plan-2020-2022-FINAL.pdf

Open Rules. (n.d.). Law analysis - methods. Retrieved March 3, 2023, from https://open-regels.nl/en/methoden/wetsanalyse/#the-legal-analysis-scheme

Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph databases* (Second edition) [OCLC: ocn911172345]. O'Reilly.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead [Number: 5 Publisher: Nature Publishing Group]. *Nature Machine Intelligence*, *1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K.-R. (Eds.). (2019). *Explainable AI: Interpreting, explaining and visualizing deep learning* (Vol. 11700). Springer International Publishing. https://doi.org/10.1007/978-3-030-28954-6

Schmidt, D. (2006). Guest editor's introduction: Model-driven engineering [Conference Name: Computer]. *Computer*, *39*(2), 25–31. https://doi.org/10.1109/MC.2006.58

Sørmo, F., Cassens, J., & Aamodt, A. (2005). Explanation in case-based reasoning–perspectives and goals. *Artificial Intelligence Review*, *24*(2), 109–143. https://doi.org/10.1007/s10462-005-4607-7

The High-Level Expert Group on Artificial Intelligence. (2019). Ethics guidelines for trustworthy AI. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Timmer, I., & Rietveld, R. (2019). Rule-based systems for decision support and decision-making in dutch legal practice. A brief overview of applications and implications. *Droit et Societe*, *2019*, 517–534.

Tversky, A., & Kahneman, D. (1981). The Framing of Decisions and the Psychology of Choice [Publisher: American Association for the Advancement of Science]. *Science*, *211*(4481), 453–458. https://doi.org/10.1126/science.7455683

van Deursen, A., Klint, P., & Visser, J. (2000). Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, *35*(6), 26–36. https://doi.org/10.1145/352029.352035

van Eck, M. (2018). *Geautomatiseerde ketenbesluiten & rechtsbescherming: Een onderzoek naar de praktijk van geautomatiseerde ketenbesluiten over een financieel belang in relatie tot rechtsbescherming.*

Vanthienen, J., & Wets, G. (1994). From decision tables to expert system shells. *Data & Knowledge Engineering*, *13*(3), 265–282. https://doi.org/10.1016/0169-023X(94)00020-4

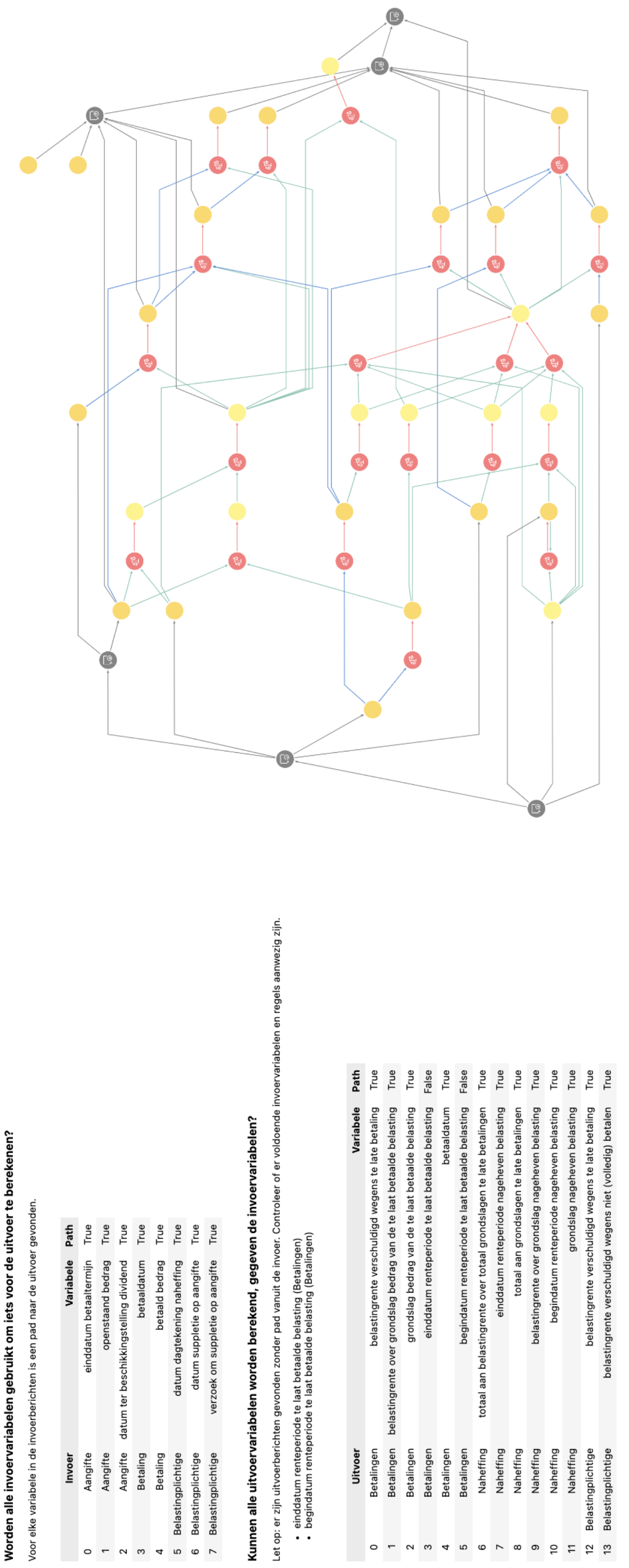
Vilone, G., & Longo, L. (2020). Explainable artificial intelligence: A systematic review [arXiv:2006.00093 [cs]]. Retrieved September 15, 2022, from http://arxiv.org/abs/2006.00093

Völter, M. (Ed.). (2013). *DSL engineering: Designing, implementing and using domain-specific languages*. CreateSpace Independent Publishing Platform.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the general data protection regulation. https://doi.org/10.2139/SSRN.2903469

Weiser, M., & Brown, J. S. (1995). Designing calm technology.

Expanded Figures of Explanation Results



Worden alle invoervariabelen gebruikt om iets voor de uitvoer te berekenen?

Voor elke variabele in de invoerberichten is een pad naar de uitvoer gevonden.

	Invoer	Variabele	Path
0	Aangifte	einddatum betaaltermijn	True
1	Aangifte	openstaand bedrag	True
2	Aangifte	datum ter beschikkingstelling dividend	True
3	Betaling	betaaldatum	True
4	Betaling	betaald bedrag	True
5	Belastingplichtige	datum dagtekening naheffing	True
6	Belastingplichtige	datum suppletie op aangifte	True
7	Belastingplichtige	verzoek om suppletie op aangifte	True

Kunnen alle uitvoervariabelen worden berekend, gegeven de invoervariabelen?

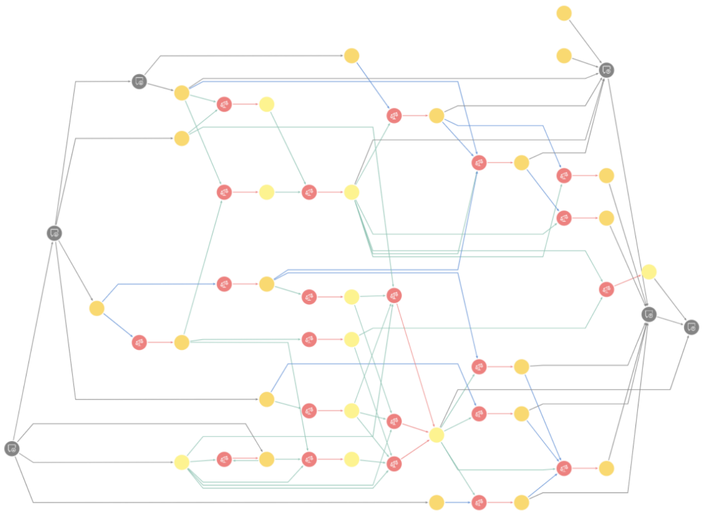
Let op: er zijn uitvoerberichten gevonden zonder pad vanuit de invoer. Controleer of er voldoende invoervariabelen en regels aanwezig zijn.

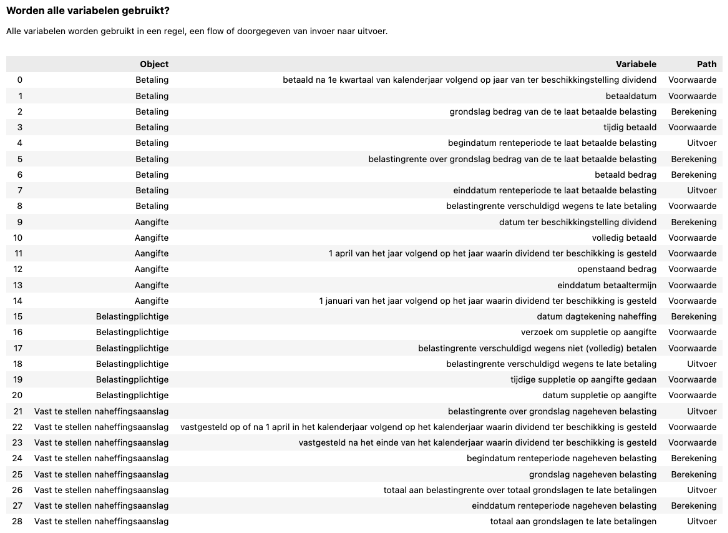
- einddatum renteperiode te laat betaalde belasting (Betalingen)
- begindatum renteperiode te laat betaalde belasting (Betalingen)

	Uitvoer	Variabele	Path
0	Betalingen	belastingrente verschuldigd wegens te late betaling	True
1	Betalingen	belastingrente over grondslag bedrag van de te laat betaalde belasting	True
2	Betalingen	grondslag bedrag van de te laat betaalde belasting	True
3	Betalingen	einddatum renteperiode te laat betaalde belasting	False
4	Betalingen	betaaldatum	True
5	Betalingen	begindatum renteperiode te laat betaalde belasting	False
6	Naheffing	totaal aan belastingrente over totaal grondslagen te late betalingen	True
7	Naheffing	einddatum renteperiode nageheven belasting	True
8	Naheffing	totaal aan grondslagen te late betalingen	True
9	Naheffing	belastingrente over grondslag nageheven belasting	True
10	Naheffing	begindatum renteperiode nageheven belasting	True
11	Naheffing	grondslag nageheven belasting	True
12	Belastingplichtige	belastingrente verschuldigd wegens te late betaling	True
13	Belastingplichtige	belastingrente verschuldigd wegens niet (volledig) betalen	True

Figure 17

Part of the model expert explanation and answer to two questions, namely "Is all input used to create the output?" and "Can all output be created, given the input?" (see Figure 12)





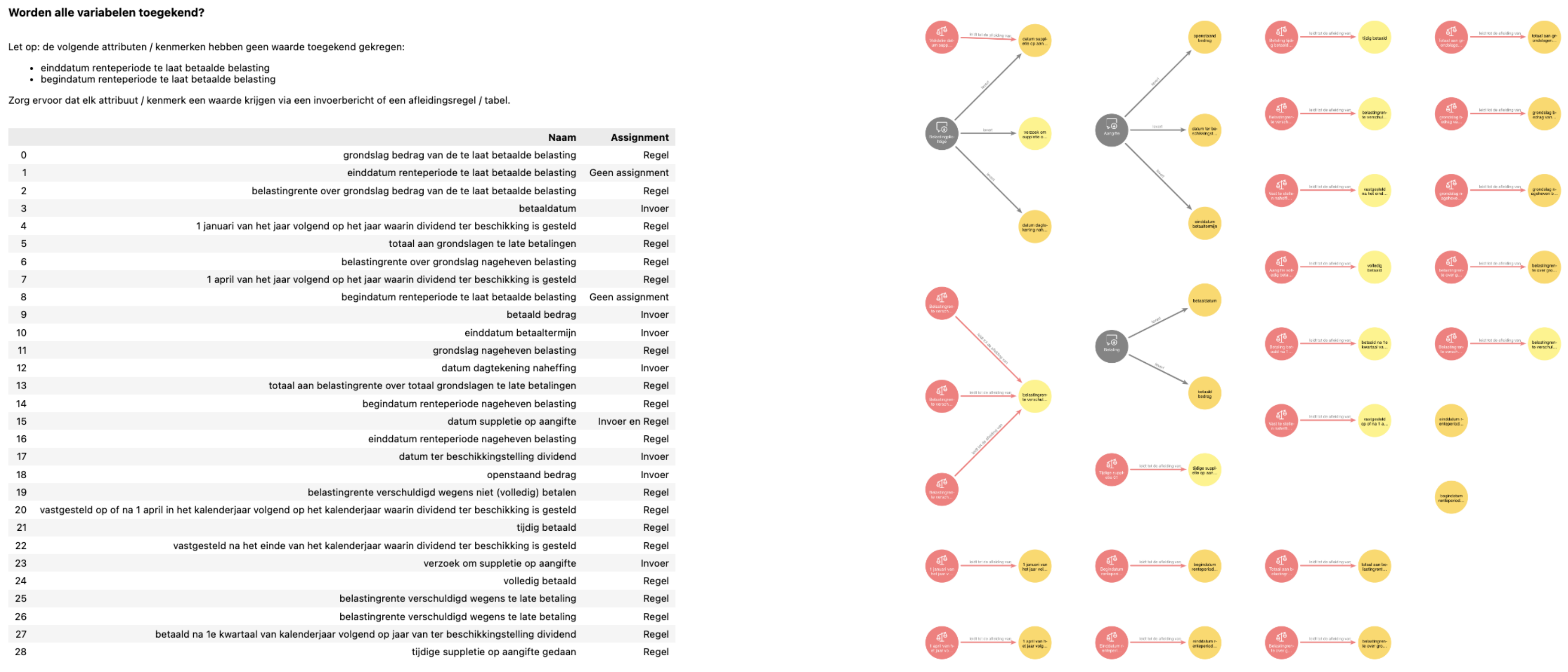
Worden alle variabelen gebruikt?

Alle variabelen worden gebruikt in een regel, een flow of doorgegeven van invoer naar uitvoer.

	Object	Variabele	Path
0	Betaling	betaald na 1e kwartaal van kalenderjaar volgend op jaar van ter beschikkingstelling dividend	Voorwaarde
1	Betaling	betaaldatum	Voorwaarde
2	Betaling	grondslag bedrag van de te laat betaalde belasting	Berekening
3	Betaling	tijdig betaald	Voorwaarde
4	Betaling	begindatum renteperiode te laat betaalde belasting	Uitvoer
5	Betaling	belastingrente over grondslag bedrag van de te laat betaalde belasting	Berekening
6	Betaling	betaald bedrag	Berekening
7	Betaling	einddatum renteperiode te laat betaalde belasting	Uitvoer
8	Betaling	belastingrente verschuldigd wegens te late betaling	Voorwaarde
9	Aangifte	datum ter beschikkingstelling dividend	Berekening
10	Aangifte	volledig betaald	Voorwaarde
11	Aangifte	1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Voorwaarde
12	Aangifte	openstaand bedrag	Voorwaarde
13	Aangifte	einddatum betaaltermijn	Voorwaarde
14	Aangifte	1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Voorwaarde
15	Belastingplichtige	datum dagtekening naheffing	Berekening
16	Belastingplichtige	verzoek om suppletie op aangifte	Voorwaarde
17	Belastingplichtige	belastingrente verschuldigd wegens niet (volledig) betalen	Voorwaarde
18	Belastingplichtige	belastingrente verschuldigd wegens te late betaling	Uitvoer
19	Belastingplichtige	tijdige suppletie op aangifte gedaan	Voorwaarde
20	Belastingplichtige	datum suppletie op aangifte	Voorwaarde
21	Vast te stellen naheffingsaanslag	belastingrente over grondslag nageheven belasting	Uitvoer
22	Vast te stellen naheffingsaanslag	vastgesteld op of na 1 april in het kalenderjaar volgend op het kalenderjaar waarin dividend ter beschikking is gesteld	Voorwaarde
23	Vast te stellen naheffingsaanslag	vastgesteld na het einde van het kalenderjaar waarin dividend ter beschikking is gesteld	Voorwaarde
24	Vast te stellen naheffingsaanslag	begindatum renteperiode nageheven belasting	Berekening
25	Vast te stellen naheffingsaanslag	grondslag nageheven belasting	Berekening
26	Vast te stellen naheffingsaanslag	totaal aan belastingrente over totaal grondslagen te late betalingen	Uitvoer
27	Vast te stellen naheffingsaanslag	einddatum renteperiode nageheven belasting	Berekening
28	Vast te stellen naheffingsaanslag	totaal aan grondslagen te late betalingen	Uitvoer

Figure 18

Part of the model expert explanation and answer to Is each variable used?" (see Figure 13)



Worden alle variabelen toegekend?

Let op: de volgende attributen / kenmerken hebben geen waarde toegekend gekregen:

- einddatum renteperiode te laat betaalde belasting
- begindatum renteperiode te laat betaalde belasting

Zorg ervoor dat elk attribuut / kenmerk een waarde krijgen via een invoerbericht of een afleidingsregel / tabel.

	Naam	Assignment
0	grondslag bedrag van de te laat betaalde belasting	Regel
1	einddatum renteperiode te laat betaalde belasting	Geen assignment
2	belastingrente over grondslag bedrag van de te laat betaalde belasting	Regel
3	betaaldatum	Invoer
4	1 januari van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Regel
5	totaal aan grondslagen te late betalingen	Regel
6	belastingrente over grondslag nageheven belasting	Regel
7	1 april van het jaar volgend op het jaar waarin dividend ter beschikking is gesteld	Regel
8	begindatum renteperiode te laat betaalde belasting	Geen assignment
9	betaald bedrag	Invoer
10	einddatum betaaltermijn	Invoer
11	grondslag nageheven belasting	Regel
12	datum dagtekening naheffing	Invoer
13	totaal aan belastingrente over totaal grondslagen te late betalingen	Regel
14	begindatum renteperiode nageheven belasting	Regel
15	datum suppletie op aangifte	Invoer en Regel
16	einddatum renteperiode nageheven belasting	Regel
17	datum ter beschikkingstelling dividend	Invoer
18	openstaand bedrag	Invoer
19	belastingrente verschuldigd wegens niet (volledig) betalen	Regel
20	vastgesteld op of na 1 april in het kalenderjaar volgend op het kalenderjaar waarin dividend ter beschikking is gesteld	Regel
21	tijdig betaald	Regel
22	vastgesteld na het einde van het kalenderjaar waarin dividend ter beschikking is gesteld	Regel
23	verzoek om suppletie op aangifte	Invoer
24	volledig betaald	Regel
25	belastingrente verschuldigd wegens te late betaling	Regel
26	belastingrente verschuldigd wegens te late betaling	Regel
27	betaald na 1e kwartaal van kalenderjaar volgend op jaar van ter beschikkingstelling dividend	Regel
28	tijdige suppletie op aangifte gedaan	Regel

Figure 19

Part of the model expert explanation and answer to "Are all variables assigned?" (see Figure 13)