

# MASTER'S THESIS

## Business Rule Traceability als hulpmiddel bij softwareontwikkeling in de justitiële context

Soerokromo, M.

**Award date:**  
2022

[Link to publication](#)

### General rights

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

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

### Take down policy

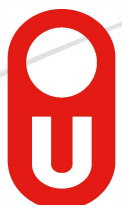
If you believe that this document breaches copyright please contact us at:

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

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 10. Dec. 2022

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



# Business Rule Traceability als hulpmiddel bij softwareontwikkeling in de justitiële context

## Business Rule Traceability as a helping mechanism for software development within the judicial context

Opleiding: Open Universiteit, faculteit Bètawetenschappen  
Masteropleiding Business Process Management & IT

Programme: Open University of the Netherlands, faculty of Science  
Master of Science Business Process Management & IT

Cursus: IM9806 Afstudeeropdracht Business Process Management and IT

Student: Mark Soerokromo

Identiteitsnummer:

Datum: 20/06/2022

Afstudeerbegeleider Dr. Lloyd Rutledge

Meelezer Dr. Ir. Ella Roubtsova

Versie nummer: 1

Status: Definitief

## Abstract

Er is onderzoek verricht naar de toepasbaarheid van een voorgestelde referentiearchitectuur voor traceerbaarheid van bedrijfsregels in informatiesystemen (RA4T) binnen de justitiële context. Een werkbaar model kan bijdragen aan het ontwerpen van systemen die ambtenaren ondersteunen bij het uitvoeren van hun werkzaamheden. Dankzij traceerbaarheid kan de output van het informatiesysteem worden onderbouwd en kunnen de regels achter de output worden weergegeven. Ook kan traceerbaarheid leiden tot concretere bedrijfsregels. Tijdens het ontwikkelen van het informatiesysteem wordt namelijk inzicht verkregen in de, mogelijk onbedoelde, resultaten van het systeem. Om de werkbaarheid van het model te evalueren, is een testmodel opgesteld op basis van de relevante wetgeving. Deze wetgeving is omgezet in bedrijfsregels die vertaald zijn naar een controlled natural language (CNL). Om een systeem met traceerbaarheid te simuleren dat geëvalueerd kan worden, is de CNL geconverteerd naar een ontologie voor het semantische web die is weergegeven in een Semantic Web Ontology Editor. Illustratieve scenario's zijn opgesteld op basis van gevoerde rechtszaken over de uitkomsten van een aanvraag voor een Verklaring Omtrent Gedrag (VOG). De eigenschappen uit de scenario's zijn in het model geladen. De traceerbaarheidsstappen zijn gevolgd, waardoor duidelijk is geworden hoe deze traceerbaarheid kan bijdragen aan de uitvoering van de werkzaamheden van de ambtenaar. Daarbij is gekeken op het niveau van de ambtenaar, de ontwikkelaar en de wetgever. Op alle niveaus zijn voordelen gevonden.

## Sleutelbegrippen

Business Rule Traceability, Controlled Natural Language, Reference architecture for business rule traceability, Rule-based systems, Business rules in software development

## Samenvatting

Automatisering binnen de overheid heeft geleid tot onbedoelde gevolgen van wetgeving. Deze automatisering komt steeds vaker voor binnen de gehele overheid. Dit verhoogt het risico op nieuwe situaties waarin wetgeving een onbedoeld effect heeft op de samenleving. De automatiseringssystemen binnen de overheid zijn ontwikkeld op basis van wetten. Een probleem is dat de vertaling van wet naar systeem niet gecontroleerd kan worden. Hierdoor is onduidelijk of de algoritmes functioneren zoals ze zijn bedoeld.

De vertaling van wetten naar IT-systemen is onder andere mogelijk door de wetgeving te vertalen naar bedrijfsregels. Bedrijfsregels zijn afspraken tussen verschillende partijen in een organisatie die bedoeld zijn om bij te dragen aan het behalen van de doelen van de organisatie. Om bedrijfsregels geschikt te maken voor een IT-systeem, kunnen de bedrijfsregels omgezet worden in een controlled natural language (CNL).

Dit rapport beschrijft het onderzoek naar de bruikbaarheid van de referentiearchitectuur voor traceerbaarheid (RA4T), ontwikkeld door Rutledge en Italiaander. Systemen die ontwikkeld zijn volgens dit model bieden eindgebruikers de mogelijkheid om de logica achter het systeem te traceren tot aan de bron-documenten. Dit model kan een oplossing bieden voor sommige problemen als gevolg van automatisering binnen de overheid.

Om de RA4T te evalueren binnen de justitiële context is een model opgesteld door wetgeving om te zetten in bedrijfsregels. De wetgeving die gebruikt is voor het model is de wetgeving rondom de Verklaring Omtrent Gedrag (VOG). Deze bedrijfsregels zijn vervolgens vertaald naar een CNL. Gekozen is voor de CNL Semantics of Business Vocabulary and Business Rules Structured English (SBVR-SE). De SBVR-SE is vervolgens met behulp van een converter getransformeerd naar een taal die verwerkbaar is voor machines. De output is daarna ingeladen in een Semantic Web Ontology Editor, wat de mogelijkheid biedt om conflicterende regels te signaleren en conclusies te genereren.

Om de bruikbaarheid van deze referentiearchitectuur aan te tonen, is het model geëvalueerd door middel van scenario's. De scenario's in dit onderzoek zijn opgesteld met gegevens uit rechtszaken die gevoerd zijn over de uitkomst van een aanvraag van een Verklaring Omtrent Gedrag (VOG).

Op basis van de onderzochte scenario's zijn voordelen gevonden voor een systeem dat gebaseerd is op de RA4T. Een dergelijk systeem kan de kennis van ambtenaren over achterliggende wetteksten vergroten. Ook kan de vertaling van wetteksten naar computertaal tijdens het ontwikkelen gecontroleerd worden door zowel de ontwikkelaar als de wetgever. De werking van de algoritmen kan door de wetgever gecontroleerd worden om onbedoelde gevolgen te voorkomen.

# Inhoudsopgave

Abstract .....	2
Sleutelbegrippen .....	2
Samenvatting .....	3
Inhoudsopgave .....	4
1.   Introductie .....	5
1.1.   Aanleiding .....	5
1.2.   Gebiedsverkenning .....	6
1.3.   Probleemstelling .....	8
1.4.   Opdrachtformulering .....	8
1.5.   Motivatie en relevantie.....	9
1.6.   Leeswijzer.....	10
2.   Theoretisch kader .....	11
2.1.   Onderzoeksaanpak.....	11
2.2.   Literatuuronderzoek .....	12
2.3.   Resultaten en conclusies.....	14
2.4.   Doel van het vervolgonderzoek .....	15
3.   Methodologie.....	16
3.1.   Onderzoeksmethode .....	16
3.2.   Technisch ontwerp: uitwerking van de methode .....	17
3.3.   Gegevensanalyse.....	18
3.4.   Validiteit, betrouwbaarheid en ethische aspecten.....	20
4.   Resultaten .....	21
4.1.   Algemeen scenario.....	21
4.2.   Scenario: ambtenaar.....	24
4.3.   Scenario: ontwikkelaar.....	28
4.4.   Scenario: wetgever .....	33
5.   Conclusie, discussie en aanbevelingen .....	37
5.1.   Conclusie .....	37
5.2.   Discussie.....	38
5.3.   Aanbevelingen voor praktijk- en vervolgonderzoek .....	38
5.4.   Reflectie .....	39
6.   Bibliography .....	40
7.   Appendix 1 .....	42

# 1. Introductie

## 1.1. Aanleiding

Er is recentelijk maatschappelijke ophef ontstaan doordat wetgeving heeft geleid tot onbedoelde gevolgen: de veelbesproken toeslagenaffaire (Donner, den Ouden, Klijnsma, Akdemir, & Gosen, 2020). Deze toeslagenaffaire heeft geleid tot de val van het kabinet Rutte III (Hendrickx, 2021). Een adviescommissie onder leiding van oud-minister Donner heeft deze toeslagenaffaire onderzocht. Het adviesrapport *Omzien in Verwondering* richt zich op de toeslagenaffaire, en probeert antwoord te geven op de vraag: *hoe heeft het zo ver kunnen komen?*

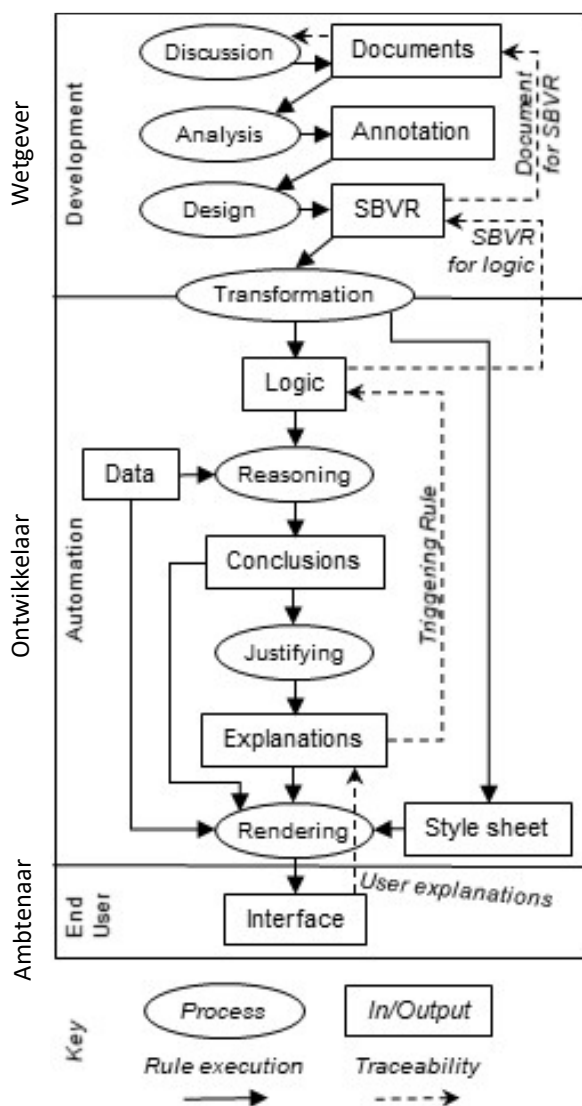
In het adviesrapport van de commissie-Donner wordt onder andere een beeld geschetst van de processtructuur binnen de Belastingdienst. Er is sprake van achterstanden, een hoge werkdruk en verlies van ervaren krachten doordat deze vervroegd met pensioen zijn gegaan. Ook wordt vermoed dat kennis van de achterliggende wetteksten en uitspraken bij ambtenaren nauwelijks aanwezig is (Donner et al., 2020). Om deze problemen op te lossen, is er een toenemende mate van automatisering, worden algoritmes gebruikt om beslissingen te nemen en werken medewerkers voornamelijk via werkinstructies en afvinklijsten (Donner et al., 2020). Tevens heeft het onderzoek uitgewezen dat is afgeweken van wettelijke regelgeving als gevolg van deze automatisering van de besluitvorming.

Deze processtructuur blijkt niet uniek voor de ambtenaren van de Belastingdienst. Uit het adviesrapport van de afdeling Advisering van de Raad van State blijkt dat geautomatiseerde besluitvorming, ofwel beslissingen op basis van algoritmen, steeds vaker voorkomt binnen overheidsprocessen (Staten-generaal, 2018). Hoewel de voordelen duidelijk zijn – een computer kan immers sneller en nauwkeuriger grote hoeveelheden gegevens verwerken dan een ambtenaar – levert het gebruik van algoritmen ook problemen op. Zo is het voor niet-ICT-specialisten niet te controleren of de vertaling van wetteksten naar programmeertaal naar machinetaal goed is gegaan. Zonder deze controle is het niet duidelijk of het algoritme functioneert zoals bedoeld. Ook is voor ambtenaren moeilijk vast te stellen op basis van welke gegevens een beslissing genomen wordt (Staten-generaal, 2018).

Met de toenemende mate van automatisering binnen de overheid neemt ook de kans toe dat er iets misgaat. Het adviesrapport schetst praktijksituaties binnen verschillende overheidsinstellingen waar algoritmes of zelflerende systemen tot resultaten komen die niet in lijn zijn met de intentie van de wet (Staten-generaal, 2018). Echter, zonder automatisering zal de werkdruk van de overheid onbeheersbaar hoog worden. Dit leidt tot de vraag hoe ervoor gezorgd kan worden dat de uitvoering van de wet in lijn is met de intentie van de wet, op een manier die werkbaar blijft voor ambtenaren.

## 1.2. Gebiedsverkenning

In dit rapport wordt de toepasbaarheid geëvalueerd van de *reference architecture for traceability*, hierna afgekort als RA4T. Deze referentiearchitectuur is ontwikkeld door Rutledge en Italiaander. Vanwege de toenemende complexiteit van software zijn softwarearchitecturen essentieel voor de ontwikkeling van een gedegen systeem. Een referentiearchitectuur is een vorm van softwarearchitectuur waarbij de software-elementen binnen de referentiearchitectuur geplaatst kunnen worden en de dataflow weergegeven kan worden (Pinheiro, 2004). De RA4T tracht ontwikkelaars te helpen bij het ontwikkelen van systemen waarbij alle onderdelen de benodigde informatie voor traceerbaarheid kunnen uitwisselen (Rutledge & Italiaander, 2021). Dit model wordt weergegeven in figuur 1.



Figuur 1. Referentiearchitectuur voor traceerbaarheid (Rutledge & Italiaander, 2021). Het niveau van de eindgebruiker is in dit onderzoek toegevoegd.

Traceerbaarheid binnen softwareontwikkeling wordt ook wel *requirements traceability* genoemd en bestaat uit *backward*, *forward* en *relationship traceability*. Requirements traceability heeft meerdere voordelen voor de ontwikkeling en het onderhoud van een systeem (International institute of Business Analysis, 2015).

De RA4T volgt de backward traceability. Via de gestippelde lijnen in figuur 1, de *traceability-links*, kunnen de achterliggende requirements gezien worden. Binnen dit onderzoek wordt traceerbaarheid dan ook gedefinieerd als de mogelijkheid om de beslissingen van het systeem te herleiden tot de bedrijfsregels achter de logica en verder terug, tot aan de originele brondocumenten die de basis vormen voor de bedrijfsregels.

De RA4T richt zich op rule-based systemen die gebruikmaken van controlled natural language (CNL) voor de logica van het systeem. De regels binnen een dergelijk systeem zijn bedrijfsregels die vertaald zijn naar CNL. Bedrijfsregels zijn niet alleen aanwezig in bedrijven, maar kunnen ook betrekking hebben op wetgeving (Gordon, Governatori, & Rotolo, 2009). Bedrijfsregels zorgen ervoor dat de eisen van het bedrijf vertaald kunnen worden naar het IT-domein.

De RA4T van Rutledge en Italiaander beschouwt het rule-based systeem op drie niveaus: ontwikkeling (*development*), automatisering (*automation*) en eindgebruiker (*end user*) zoals weergegeven in figuur 1. Ontwikkeling binnen het model faciliteert agile softwareontwikkeling: het constant verbeteren van het systeem door iteraties in tegenstelling tot periodieke updates (Dybå & Dingsøyr, 2009).

Aan de basis van de ontwikkeling staan de brondocumenten met bedrijfsregels. Het verder formaliseren van bedrijfsregels kan onder andere met behulp van CNL. Een CNL is een door regels ingeperkte vorm van natuurlijke taal. Door deze restricties worden zinnen minder complex en minder dubbelzinnig. CNL is eenvoudiger te lezen voor zowel mens als machine (Schwitter & Tilbrook, 2004).

In het model van Rutledge en Italiaander worden de bedrijfsregels in natuurlijke taal omgezet via Semantics of Business Vocabulary and Business Rules (SBVR). SBVR kan vervolgens geformaliseerd worden in een CNL die gebruikt kan worden voor de systeemlogica. Semantics of Business Vocabulary and Business Rules Structured English (SBVR-SE) en RuleSpeak zijn voorbeelden van CNL die gebruikt kunnen worden voor SBVR-formuleringen (Open Management Group, 2013). Het doel is om de bedrijfsregels, die complex verwoord kunnen zijn, te vertalen naar zinnen die eenvoudig te begrijpen zijn en die verwerkt kunnen worden door een geautomatiseerd systeem (Open Management Group, 2013).

Hiermee is het tweede niveau bereikt: automatisering. Uit de bedrijfsregels in SBVR volgen instructies in machinetaal. Het laatste niveau betreft de eindgebruiker die via een interface de uitkomst van het proces ziet.

De RA4T volgt de gestippelde traceability-links zoals weergegeven in figuur 1. Deze links bieden per niveau de mogelijkheid om terug te gaan naar een vorige stap in het proces. Zo kan de eindgebruiker de logica achter de beslissing zien via de link *user explanations* en vervolgens via *triggering rules* inzien welke regel hier achter zit. Vervolgens zou de eindgebruiker nog verder terug kunnen gaan naar de SBVR die gebruikt is voor de logica, via de link *SBVR for logic*. Als laatste stap zou de eindgebruiker de documentatie waarop de SBVR gebaseerd is kunnen bekijken via *document for SBVR* (Rutledge & Italiaander, 2021). De RA4T biedt eindgebruikers hiermee de mogelijkheid om te begrijpen waarom het systeem tot een bepaalde conclusie komt.

Deze traceerbaarheid op het gebied van rule-based systemen maakt voor eindgebruikers inzichtelijk welke beslissingen genomen zijn en op grond van welke regels deze beslissingen genomen zijn. De



RA4T streeft ernaar niet alleen te laten zien welke beslissingen er genomen worden, maar ook op basis van welke originele bedrijfsregels en brondocumenten deze beslissingen genomen worden. Hierdoor begrijpen eindgebruikers de redenering achter de beslissingen beter, wat kan leiden tot acceptatie van het systeem en verbeteringen van dit systeem (Rutledge & Italiaander, 2021).

### 1.3. Probleemstelling

Door de toenemende mate van automatisering bij de overheid nemen de risico's toe van onbedoelde gevolgen van de uitvoering van wetgeving (Staten-generaal, 2018; Donner et al., 2020). De kennis van ambtenaren over wetgeving en uitspraken lijkt af te nemen. De huidige systemen maken het lastig voor ambtenaren om inzicht te krijgen in de gegevens die gebruikt zijn om tot bepaalde beslissingen te komen (Staten-generaal, 2018). De ambtenaar ziet alleen het resultaat. De eindgebruiker van het systeem is niet meer in staat om een aantal stappen terug te gaan en de originele documenten te bekijken waarop de logica gebaseerd is. De RA4T in SBVR-gebaseerde systemen kan een uitkomst bieden voor dit probleem door eindgebruikers in staat te stellen om de logica van het systeem terug te volgen tot en met de originele bronteksten. Dit model is echter nog niet geëvalueerd.

### 1.4. Opdrachtformulering

Uit de probleemstelling volgt dat het model geëvalueerd moet worden. In dit onderzoek wordt het model getest door middel van wetteksten.

Dit leidt tot de volgende onderzoeksvraag:

*Hoe, wanneer en in hoeverre kan de referentiearchitectuur voor traceerbaarheid (RA4T) bijdragen aan het uitvoeren van ambtelijke werkzaamheden?*

Om te beoordelen hoe de RA4T een bijdrage kan leveren, wordt gekeken naar drie mogelijke voordelen die het model kan opleveren.

Het eerste voordeel betreft de *eindgebruikers* van het systeem. De eindgebruiker in dit onderzoek is de ambtenaar. De ambtenaar werkt met de interface van het systeem en ziet hier de beslissingen uit voortkomen. Door ambtenaren in staat te stellen om de logica van het systeem te herleiden tot de bedrijfsregels, zullen zij beter in staat zijn de resultaten te begrijpen. De ambtenaar opereert op de *end-user*-laag van de referentiearchitectuur.

Het tweede voordeel betreft de *ontwikkelaars* van het systeem. Tijdens het ontwikkelen van het systeem kunnen eindgebruikers namelijk feedback geven over de werking van de bedrijfsregels binnen het systeem, wat ontwikkeling vergemakkelijkt. De ontwikkelaar opereert op de *automation*-laag van de referentiearchitectuur.

Een derde voordeel zou voor de *wetgever* kunnen zijn. Onbedoelde gevolgen van wetgeving kunnen tijdens het ontwikkelproces naar voren komen, wat de mogelijkheid biedt om de wetgeving te verbeteren. De wetgever opereert op de *development*-laag van de referentiearchitectuur.

Door de voordelen op de niveaus van de verschillende betrokkenen duidelijk te maken in dit onderzoek, kan binnen het justitiële domein een weloverwogen keuze gemaakt worden met betrekking tot het soort systeem dat ontwikkeld moet worden voor de uitvoering van ambtelijke werkzaamheden.

Uit de hoofdvraag en de drie lagen van de referentiearchitectuur volgen drie deelvragen.

*Hoe, wanneer en in hoeverre helpt de RA4T...:*

1. *... eindgebruikers bij het begrijpen van de resultaten van het systeem?*
2. *...ontwikkelaars bij het ontwikkelen van het systeem?*
3. *...de wetgever bij het verbeteren van de wetten?*

Voor elke deelvraag wordt onderzocht hoe, wanneer en in hoeverre de RA4T kan helpen. Het theoretisch kader wordt gebruikt om te beschrijven hoe het model zou kunnen helpen. Wanneer en in hoeverre de RA4T kan helpen, moet blijken uit vervolgonderzoek waarin de referentiearchitectuur geëvalueerd wordt.

Door antwoord te geven op de drie deelvragen, wordt ook de hoofdvraag beantwoord. Door te analyseren op welke punten het model niet goed werkt, kunnen aanbevelingen gedaan worden om het model te verbeteren.

## 1.5. Motivatie en relevantie

Het doel van onderzoek op het gebied van Information Sciences is het verkrijgen van kennis en begrip ten aanzien van de ontwikkeling en implementatie van op technologie gebaseerde oplossingen voor relevante bedrijfsproblemen (Hevner, March, Park, & Ram, 2004). Design science research kan op drie manieren een wetenschappelijke bijdrage leveren, namelijk via de elementen *design artifact*, *foundations* en *methodologies*. Minimaal een van deze drie elementen moet aanwezig zijn binnen design science research. Dit onderzoek draagt bij aan design science door het evalueren van het design artifact dat valt binnen deze methodologieën. Door gebruik te maken van de juiste evaluatiemethoden kan worden bijgedragen aan de knowledge base (Hevner et al., 2004).

De overheid heeft in verschillende adviesrapporten gesignaleerd dat de toenemende mate van op technologie gebaseerde oplossingen tot nieuwe problemen kan leiden (Donner et al., 2020; Staten-generaal, 2018). Deze structurele ontwikkeling vraagt om structurele oplossingen, omdat het risico van onbedoelde gevolgen toeneemt met de toenemende automatisering. Een voorbeeld van een onbedoeld gevolg van automatisering is de toeslagenaffaire (Hendrickx, 2021). Een werkbare referentiearchitectuur voor traceerbaarheid zou onbedoelde gevolgen op het niveau van de eindgebruiker, de ontwikkelaar en de wetgever kunnen voorkomen. De RA4T tracht ontwikkelaars te helpen bij het ontwikkelen van software die traceren mogelijk maakt (Rutledge & Italiaander, 2021). Een ander mogelijk voordeel van traceerbaarheid is het openen van een dialoog tussen de eindgebruiker en de opstellers van de bedrijfsregels. Hierdoor zou de brontekst verbeterd kunnen worden, waardoor het voor eindgebruikers eenvoudiger wordt om de logica van het systeem te begrijpen en hun werkzaamheden uit te voeren (Rutledge & Italiaander, 2021).

## 1.6. Leeswijzer

Het vervolg van dit onderzoeksrapport is als volgt opgebouwd. In Hoofdstuk 2 wordt de bestaande literatuur geanalyseerd aan de hand van een literatuuronderzoek. Hoofdstuk 3 gaat in op de methodologie die gebruikt is voor het onderzoek. Hierbij worden de onderzoeksmethoden beschreven, maar ook de manier waarop de data verzameld zijn en hoe deze geoperationaliseerd zijn. In Hoofdstuk 4 wordt beschreven hoe het model is ontwikkeld en toegepast. Drie scenario's zijn uit de wetgeving over de VOG gehaald en verwerkt met behulp van Semantic-web-logic-tools en interface-georiënteerde software. Vervolgens wordt geanalyseerd hoe goed en in hoeverre het model werkt voor elk gekozen scenario. Hoofdstuk 5 geeft een antwoord op de onderzoeksvraag met behulp van de resultaten uit Hoofdstuk 4. Ook wordt aandacht besteed aan de aanbevelingen voor de praktijk en mogelijk vervolgonderzoek. Het hoofdstuk wordt afgesloten met een kritische blik op de kwaliteit van dit onderzoek en de houdbaarheid van de conclusies.

## 2. Theoretisch kader

In dit hoofdstuk wordt bestaande en verwante literatuur over de RA4T belicht. Op basis van de gevonden literatuur wordt een antwoord gezocht op de deelvragen. In het vervolgonderzoek wordt getoetst of de gevonden theoretische antwoorden overeenkomen met de praktijk.

### 2.1. Onderzoeksaanpak

Aan de basis van het onderzoek ligt de RA4T die is gepresenteerd in de paper *Towards a reference architecture for Traceability in SBVR-based systems* (Rutledge & Italiaander, 2021). Deze RA4T is gebaseerd op rule-based systemen die gebruikmaken van CNL voor de logica van het systeem, waarbij SBVR-SE is gekozen als CNL voor het onderzoek.

Voor het literatuuronderzoek is de volgende combinatie van methoden gebruikt: *building blocks*, *snowballing* en *berry picking*.

Building blocks is een vaak gebruikte zoekmethode voor literatuuronderzoek (Booth, 2008). Een query wordt in verschillende onderdelen verdeeld. Voor elk onderdeel worden varianten en synoniemen gezocht. Daarna worden de onderdelen in verschillende varianten en synoniemen samengevoegd als zoekterm. De gebruikte query's tijdens de start van dit onderzoek zijn: *business rule traceability*, *rule-based systems*, *Semantic Web rule language*, *reference architecture for business rule traceability*, *framework evaluation* en *traceability in software development*.

Snowballing begint met een startset van relevante publicaties, waarna door backward snowballing de referenties in deze publicaties onderzocht worden. Vervolgens wordt via forward snowballing nieuwe literatuur gevonden door te zoeken naar studies die de eerdere publicaties citeren. Op basis van de titel en het abstract wordt een eerste selectie gemaakt van relevante publicaties. Publicaties die het meest relevant lijken worden volledig gelezen. Dit proces van backward en forward snowballing wordt herhaald totdat voldoende literatuur gevonden is (Wohlin, 2014).

De startset in dit onderzoek is de publicatie van Rutledge en Italiaander uit 2021. Een enkele paper als startset is niet gebruikelijk, maar het aantal papers dat als startset dient, kan wel kleiner zijn bij een kleiner onderzoeksgebied. De referentiearchitectuur is een innovatief model waarnaar nog geen eerder onderzoek is uitgevoerd. Daarom is ervoor gekozen om toch verder te gaan met slechts een enkele publicatie als startset.

Berry picking is een dynamische meta-onderzoeksstrategie die bestaat uit zes tactieken: *backward chaining*, *forward chaining*, *journal run*, *area scanning*, *subject searches* en *author searching* (Booth, 2008). De methode heeft enkele overeenkomsten met snowballing, zoals het backward en forward chaining. Het verschil met snowballing is dat het startpunt van elke iteratie van de zoektocht veranderlijk is, wat telkens leidt tot andere resultaten. Berry picking begint met een algemene query, waarna op basis van de gevonden resultaten nieuwe query's worden opgesteld. In dit onderzoek is gebruikgemaakt van backward chaining en forward chaining.

Het literatuuronderzoek is uitgevoerd via de digitale bibliotheek van de Open Universiteit en Google Scholar.

## 2.2. Literatuuronderzoek

### Bedrijfsregels

Een bedrijfsregel is een uitspraak die een bepaald gedeelte van het bedrijf definieert of beperkt (Business Rule Group, 2000). Bedrijfsregels zijn essentieel voor bedrijfsmodellen (Ross, 2008). Volgens *best practice* zijn bedrijfsregels ondubbelzinnig en duidelijk voor eenieder die met de bedrijfsregels te maken heeft (Business Rules Group, 2003). Bedrijfsregels formaliseren afspraken tussen de verschillende partijen die betrokken zijn bij een bedrijf in de vorm van ‘wetten’ die bijdragen aan het behalen van de doelen van de organisatie (Joosten, 2007).

Bedrijfsregels worden in dit onderzoek als volgt gedefinieerd:

“A business rule is a verifiable statement that some stakeholders intent to obey, within a certain context” (Joosten, 2007).

Deze definitie wordt in dit onderzoek vertaald als “Een bedrijfsregel is een te verifiëren statement waarvan sommige stakeholders de intentie hebben zich hieraan, binnen een bepaalde context, te houden” (Joosten, 2007).

### Wetteksten als bedrijfsregels

Wetten kunnen beschouwd worden als een soort bedrijfsregels (Gordon et al., 2009). Ze schrijven voor wat er moet gebeuren, onder welke voorwaarden en met welke uitzonderingen. Wetteksten zijn geformuleerd in een zo ondubbelzinnig mogelijke taal. Toch blijkt er interpretatie mogelijk en komt de rechterlijke macht er geregeld aan te pas om een wet uit te leggen (Feteris, 2018). Voordat wetteksten gebruikt kunnen worden in een IT-systeem, moeten deze geformaliseerd worden. Alhoewel wetteksten al vrij rigide zijn geformuleerd, is verdere formalisering nodig om ze binnen een IT-systeem te kunnen gebruiken. Deze formalisering kan worden gerealiseerd met behulp van controlled natural languages (Schwitter & Tilbrook, 2004).

### Controlled Natural Language

Een CNL is een door regels ingeperkte vorm van natuurlijke taal. Door deze restricties worden zinnen minder complex en minder dubbelzinnig. Een CNL is eenvoudiger te lezen voor zowel mens als machine (Schwitter & Tilbrook, 2004). Het doel van CNL is om de bedrijfsregels die complex verwoord kunnen zijn, te vertalen naar zinnen die eenvoudig te begrijpen zijn en verwerkt kunnen worden door zowel IT-systemen als de domeinexpert in het bedrijf. Dit zorgt voor een betere communicatie tussen deze twee domeinen (Open Management Group, 2013).

Semantics of Business Vocabulary and Business Rules Structured English (SBVR-SE) en RuleSpeak zijn twee voorbeelden van CNL die gebruikt kunnen worden in een SBVR-framework (Open Management Group, 2013).

SBVR-SE gebruikt vier fontstijlen om onderscheid te maken tussen de concepten (Open Management Group, 2013).

<u>term</u>	Noun concept	Zelfstandig naamwoord
<u>Name</u>	Individual noun concept	Individu binnen een zelfstandig naamwoord
<u>verb</u>	Verb concept	Werkwoord
<u>keyword</u>	Used to construct statements	Logische samenhang van de zin

Een voorbeeld van deze notatie is in tabel 1 uitgewerkt. De originele wetteksten worden geformaliseerd naar bedrijfsregels in natuurlijke taal. Via de SBVR-SE-opmaakcodering zijn de bedrijfsregels te herleiden tot de wetteksten. Via een superscript wordt de relatie tussen de term vanuit het brondocument en de bedrijfsregel in SBVR-SE aangegeven. Het superscript is geen onderdeel van de standaard SBVR-SE-notatie.

Tabel 1. Van brondocument naar bedrijfsregels naar SBVR.

Brondocument: wetgeving
“Ten behoeve van de beoordeling van een <u>VOG-aanvraag</u> <sup>1</sup> ontvangt het COVOG alle justitiële gegevens betreffende de <u>aanvrager</u> die zijn geregistreerd in het JDS. De justitiële gegevens kunnen zowel uit Nederland als uit het buitenland afkomstig zijn. Aan de <u>aanvrager</u> <sup>2</sup> die <u>niet voorkomt in het JDS</u> <sup>3</sup> , wordt zonder meer een VOG <u>afgegeven</u> <sup>4</sup> ” (Beleidsregels VOG-NP-RP 2018, 2022).
Bedrijfsregels in natuurlijke taal
R1: Als de aanvrager geen strafblad heeft, wordt de aanvraag goedgekeurd.
Bedrijfsregels in SBVR-SE
If <u>aanvrager</u> <sup>2</sup> has geen <u>strafblad</u> <sup>3</sup> then <u>aanvraag</u> <sup>1</sup> is <u>goedgekeurd</u> <sup>4</sup>

### Traceerbaarheid in softwareontwikkeling

Bedrijfsregels dragen bij aan het bereiken van de doelen van een organisatie. Traceerbaarheid binnen softwareontwikkeling ofwel *requirements traceability* is erop gericht om te verifiëren of de eisen in de software gelinkt zijn aan een bedrijfsdoel (International institute of Business Analysis, 2015). Bedrijfsregels kunnen de tussenstap vormen tussen de gestelde eisen en de bedrijfsdoelen (Gordon et al., 2009). Requirements traceability is onder te verdelen in backward traceability, forward traceability en relationship traceability (International institute of Business Analysis, 2015). Backward traceability is te koppelen aan de traceability-links in de RA4T. Forward en relationship traceability zijn op moment geen onderdeel van de RA4T.

Traceerbaarheid binnen softwareontwikkeling stelt de ontwikkelaar onder andere in staat betrouwbaar vast te stellen welke eisen wel opgenomen zijn in de software en welke niet (International institute of Business Analysis, 2015).

Ander onderzoek wijst uit dat onderhoudstaken voor software die ontwikkeld is met traceerbaarheid in het vertalen van eisen naar code gemiddeld 24% sneller uitgevoerd kunnen worden dan dezelfde taken zonder traceerbaarheid (Mäder & Egyed, 2015). Het gebruik van een lichte vorm van traceerbaarheid kan al grote positieve gevolgen hebben voor het ontwikkelen en onderhouden van systemen (Krüger, Alkl, Berger, Leich, & Saake, 2019).

### Toepassing op wetgeving

Eerder onderzoek toont aan dat de implementatie van een rule-based systeem haalbaar is binnen de justitiële context (Bos, 2013). Hierdoor is de RA4T mogelijk ook bruikbaar binnen deze context. Binnen het domein van de wetgeving zou de traceerbaarheid kunnen bijdragen aan de oplossing voor enkele kritische punten uit het adviesrapport *Omzien in verwondering*. Ook zou traceerbaarheid enkele risico's uit het adviesrapport van de afdeling Advisering kunnen mitigeren.

Om vast te stellen hoe de RA4T een positieve bijdrage kan leveren binnen het domein van de wetgeving, wordt gekeken naar de huidige situatie en de verbetermogelijkheden die er zijn.

De volgende kritiekpunten zijn benoemd in *Omzien in verwondering* (Donner et al., 2020):

1. Er wordt afgeweken van wettelijke regelgeving als gevolg van de automatisering van de besluitvorming.
2. Ambtenaren hebben weinig kennis van achterliggende wetteksten en uitspraken.

De volgende kritiekpunten zijn benoemd in *Informatie- en communicatietechnologie (ICT) Nr. 557* (Staten-generaal, 2018):

1. De vertaling van de wettekst naar programmeertaal naar machinetaal is niet te controleren.
2. Het is onduidelijk of het algoritme functioneert zoals het is bedoeld.
3. Het is niet mogelijk voor ambtenaren om vast te stellen op basis van welke gegevens een beslissing is genomen.

Deze punten zijn in te delen in problemen van technische aard en tekortkomingen van menselijke aard. Punt 1 van *Omzien in verwondering* en punten 1, 2 en 3 van *Informatie- en communicatietechnologie (ICT) Nr. 557* zijn technische punten. Punt 2 uit *Omzien in verwondering* is een tekortkoming van menselijke aard.

De RA4T biedt theoretisch de mogelijkheid om alle genoemde kritische punten te verbeteren. Een systeem op basis van traceerbaarheid biedt ambtenaren de mogelijkheid om inzicht te krijgen in de onderliggende wetten en data waarop beslissingen worden gebaseerd. Hiermee zou punt 2 uit het adviesrapport *Omzien in verwondering* opgelost worden, namelijk het gebrek aan kennis van achterliggende wetteksten. Ook is dit een oplossing voor punt 3 in het adviesrapport *Informatie- en communicatietechnologie (ICT) Nr. 557* (Staten-generaal, 2018).

Naast de mogelijke voordelen voor de ambtenaar kan ook tijdens het ontwikkelen al gecontroleerd worden op basis van welke gegevens de algoritmes tot een beslissing komen. De wetgever en de ontwikkelaar kunnen op deze manier bepalen of de algoritmes functioneren zoals ze zijn bedoeld. Hiermee zou punt 1 uit het adviesrapport *Omzien in verwondering* en punten 1 en 2 van het adviesrapport *Informatie- en communicatietechnologie (ICT) Nr. 557* opgelost kunnen worden.

## 2.3. Resultaten en conclusies

Aan de hand van het theoretisch kader kan een antwoord geformuleerd worden op de deelvragen:

*Hoe helpt de referentiearchitectuur voor traceerbaarheid ...:*

1. ... eindgebruikers bij het begrijpen van de resultaten van het systeem?
2. ...ontwikkelaars bij het ontwikkelen van het systeem?
3. ...de wetgever bij het verbeteren van de wetten?

### **Eindgebruikers**

Een van de genoemde voordelen van de RA4T is dat eindgebruikers in staat worden gesteld om de gegevens in te zien die door het systeem zijn gebruikt om tot een beslissing te komen (Rutledge & Italiaander, 2021). Onderzoek van de overheid heeft uitgewezen dat ambtenaren op dit moment nauwelijks inzicht hebben in de gebruikte gegevens (Donner et al 2020; Staten-generaal, 2018). Bij een systeem gebaseerd op de RA4T kunnen ambtenaren de gegevens inzien die zijn gebruikt bij het tot stand komen van een beslissing. Daarmee wordt een van de gevonden problemen opgelost.

Ook is naar voren gekomen dat ambtenaren weinig kennis hebben van de onderliggende wetgeving waarop het systeem de beslissingen baseert (Donner et al., 2020). Door het gebruik van een systeem waarbij ambtenaren de resultaten kunnen herleiden tot de onderliggende wetteksten, zou de kennis over wetgeving kunnen toenemen.

### **Ontwikkelaars**

De RA4T stelt ontwikkelaars in staat om vertegenwoordigers van eindgebruikers feedback te laten geven over de werking van het systeem ten opzichte van de brondocumenten (Rutledge & Italiaander, 2021). Hierdoor zouden discrepanties tussen brondocumenten en output naar eindgebruikers eerder ondervangen kunnen worden.

Om een verbetering te bereiken, moet er communicatie plaatsvinden tussen de IT-specialist en de domeinexpert tijdens de ontwikkeling. De referentiearchitectuur is gebaseerd op SBVR, een CNL die de communicatie tussen de IT-specialist en de domeinexpert kan vergemakkelijken (Open Management Group, 2013). De RA4T zou ontwikkelaars dus kunnen helpen in de communicatie met de domeinexpert. De referentiearchitectuur zou echter ook gebruikt kunnen worden in combinatie met een andere CNL.

Er zijn overeenkomsten met andere onderzoeken naar traceerbaarheid binnen softwareontwikkeling. Software die is ontwikkeld op basis van requirements traceability is eenvoudiger te onderhouden, wat kan leiden tot lagere kosten (Mäder & Egyed, 2015). De RA4T zou ontwikkelaars kunnen helpen in het verminderen van de onderhoudskosten voor het systeem. Ook tijdens het ontwikkelen kan traceerbaarheid al positieve gevolgen hebben (Krüger et al., 2019).

### **Wetgever**

Door toepassing van de RA4T kan de wetgever tijdens het ontwikkelen van het systeem de vertaling van wetteksten naar machinetaal en output volgen en feedback geven aan de ontwikkelaar (Rutledge & Italiaander, 2021). Tijdens het ontwikkelen kan al gecontroleerd worden op basis van welke gegevens de algoritmes tot beslissingen komen. De wetgever en de ontwikkelaar kunnen op deze manier bepalen of de algoritmes functioneren zoals ze bedoeld zijn. Hiermee zou punt 2 van het adviesrapport *Informatie- en communicatietechnologie (ICT) Nr. 557* opgelost kunnen worden.

## 2.4. Doel van het vervolgonderzoek

Het doel van het vervolgonderzoek is het evalueren van de werkbaarheid van de RA4T. Dit theoretische model dient geëvalueerd te worden om tot een werkbaar model te komen. Op basis van de theorie uit de literatuur zou een systeem dat gebaseerd is op de RA4T moeten kunnen bijdragen aan een verbeterde ambtelijke uitvoering. Het vervolgonderzoek moet uitwijzen of deze theoretische voordelen zich ook laten vertalen in de praktijk.



### 3. Methodologie

In dit hoofdstuk wordt de methode beschreven waarmee de RA4T in de praktijk is geëvalueerd.

#### 3.1. Onderzoeksmethode

In het framework van Hevner et al. (2004) valt information science research tussen het *environment* en de *knowledge base*. Information science richt zich op het ontwikkelen van theorieën en artefacten en het verifiëren en evalueren van deze theorieën en artefacten (Hevner et al., 2004). Artefacten zijn structuren, modellen en methoden die ontwikkeld zijn voor de bouw en het gebruik van informatiesystemen (Hevner et al., 2004). De RA4T is een model dat bedoeld is om ontwikkelaars te helpen tijdens de ontwikkeling van informatiesystemen (Rutledge & Italiaander, 2021). De RA4T kan daarom worden beschouwd als een artefact.

Een van de richtlijnen voor design science research is dat de bruikbaarheid, kwaliteit en effectiviteit van een artefact strikt aangetoond moeten worden (Hevner et al., 2004). Om dit mogelijk te maken, onderscheidt Hevner vijf categorieën designevaluatiemethoden: *observational*, *analytical*, *experimental*, *testing* en *descriptive* (Hevner et al., 2004). Elke methode kan in subcategorieën worden onderverdeeld.

Observationele methoden bestaan uit casestudies en veldstudies. Deze methoden zijn op dit moment niet bruikbaar voor het te onderzoeken artefact omdat het nog niet in praktijksituaties wordt gebruikt.

Analytische methoden bestuderen een artefact op verschillende manieren aan de hand van meetbare kwaliteiten. Omdat de referentiearchitectuur RA4T nog een theoretisch model is, kunnen er nog geen metingen worden verricht.

Experimentele methoden onderzoeken een artefact in een gecontroleerde omgeving of op basis van artificiële data. Er is nog geen software ontwikkeld op basis van de referentiearchitectuur. Een experimentele onderzoeksmethode is dan ook niet mogelijk zonder eerst een werkend systeem te ontwikkelen. Dit ligt buiten de scope van dit onderzoek.

Testmethoden betreffen het testen van een werkend systeem. Dit is in de huidige situatie ook niet mogelijk.

Beschrijvende (descriptive) methoden moeten alleen gebruikt worden voor innovatieve artefacten waarvoor geen andere evaluatiemethode haalbaar is (Hevner et al., 2004). De RA4T voldoet aan deze eis.

De beschrijvende evaluatiemethoden kunnen worden onderverdeeld in *informed arguments* en scenario's.

Informed arguments gebruiken bestaande literatuur om het nut van een artefact te beargumenteren. Scenario's zijn gedetailleerde scenario's rondom het artefact om daarmee de bruikbaarheid aan te tonen. Uit het literatuuronderzoek is gebleken dat de bestaande kennis minimaal is. Om die reden zijn voor dit onderzoek scenario's gebruikt om daarmee de bruikbaarheid van de referentiearchitectuur te onderzoeken.

Een meta-analyse van de meest gebruikte design-evaluation-methoden toont aan dat *technical experiments* en *illustrative scenario's* de meest gangbare vormen zijn voor het evalueren van artefacten, omdat deze hiervoor het meest geschikt zijn (Peffer, Rothenberger, Tuunanen, & Vaczi, 2012). De keuze voor scenario's is hiermee gebaseerd op de bestaande literatuur.

## 3.2. Technisch ontwerp: uitwerking van de methode

Om de referentiearchitectuur te evalueren, zijn scenario's gebruikt. De scenario's komen voort uit de wetsartikelen waarop de VOG is gebaseerd. Deze wetsartikelen zijn vrij verkrijgbaar via het ministerie van Justitie (Wet justitiële en strafvorderlijke gegevens, 2002).

De scenario's zijn gebaseerd op praktijksituaties waarin twijfel bestaat over de interpretatie van de wetsartikelen. Scenario's moeten goed gedefinieerd zijn om een artefact te kunnen evalueren (Hevner et al., 2004). Relevante scenario's voldoen in dit onderzoek aan de volgende criteria:

- Het scenario moet duidelijk gebaseerd zijn op de wetsartikelen omtrent de VOG.
- De interpretatie van de wetsartikelen moet ter discussie staan.
- Minimaal één van de drie beoogde partijen die voordeel kan hebben van de referentiearchitectuur voor traceerbaarheid moet bij het scenario betrokken zijn.
- Het eindresultaat moet zo goed als mogelijk onbetwist zijn.

Door elk scenario aan deze vier voorwaarden te toetsen, kunnen de uitkomsten van de scenario's met elkaar vergeleken worden en kan antwoord gegeven worden op de onderzoeksvraag.

De scenario's zijn gekozen uit rechtszaken die met de VOG te maken hebben. Rechtszaken voldoen aan de vier criteria voor scenario's. Zo worden de relevante wetsartikelen genoemd in de samenvatting van een rechtszaak, waarmee aan het eerste criterium is voldaan. Daarnaast geeft het voeren van rechtszaak zelf al aan dat er een conflict is tussen partijen, waardoor de regels ter discussie staan. Ook aan de derde voorwaarde is voldaan, omdat de rechtszaken zijn aangespannen doordat een partij het niet eens is met de uitkomst van het aanvraagproces. De uitkomst van het proces is opgemaakt door de *eindgebruiker*, maar wanneer de onderliggende wetteksten onduidelijk zijn, is ook de *wetgever* hierbij betrokken. Wanneer aan de werking van het systeem wordt getwijfeld, kunnen *ontwikkelaars* hier kennis van opdoen.

Door alle scenario's op rechtszaken te baseren, kan hetzelfde model gebruikt worden met telkens andere variabelen uit de desbetreffende rechtszaak. Zo kunnen de resultaten uit de scenario's met elkaar vergeleken worden. Ook is het onderzoek op deze manier herhaalbaar in de toekomst. De wet is aan interpretatie onderhevig en daarom is alleen gezocht naar rechtszaken waarin de hoogste rechterlijke bestuurlijke macht in Nederland een uitspraak heeft gedaan, om eventuele tegenstrijdigheden in het vervolgonderzoek zoveel mogelijk tegen te gaan.

Om de bruikbaarheid van de referentiearchitectuur in verschillende scenario's te testen, zijn rechtszaken gezocht die passen bij de gebruikers van het systeem. Voor elk type gebruiker is een rechtszaak onderzocht waarin de uitkomst van het systeem eerst bevestigd en daarna gewijzigd is. Bevestiging betekent dat de originele beoordeling door een ambtenaar of de aangevochten uitspraak van een lagere rechtbank alsnog bekrachtigd is. Een wijziging houdt in dat een aangevochten uitspraak verworpen is door de Raad van State.

Elk rechtszaak is vertaald naar eigenschappen in het model. Voorbeelden hiervan in natuurlijke taal zijn: *heeft een strafblad*, *heeft reeds een VOG*, en *functieprofiel*. De volledige lijst van eigenschappen is opgesteld door het analyseren van de wetgeving en rechtszaken en het vertalen naar SBVR.

### 3.3. Gegevensanalyse

Om de voorgestelde referentiearchitectuur zo nauwkeurig mogelijk te evalueren, moeten bij de ontwikkeling van het validatiemodel alle stappen gevolgd worden die zijn beschreven in de paper van Rutledge en Italiaander. Deze paragraaf beschrijft op welke manier het model voor het evalueren van de traceerbaarheidsarchitectuur is opgesteld. Eerst is een basis-VOG-aanvraagmodel opgesteld door de relevante wetgeving te vertalen naar Web Ontology Language (OWL) 2 via drie stappen in fase 1.

#### Fase 1: Vorbereiden van het model

1. Relevante wetgeving is vertaald naar SBVR.
2. SBVR is geconverteerd naar technische code. De SBVR is in deze stap omgezet naar OWL-2-code via s2o.
3. Het model is geïmporteerd en getest. De OWL-2-code is hierbij ingelezen in Protégé.

Vervolgens zijn scenario's toegevoegd aan het basis-VOG-aanvraagmodel via de hierna beschreven stappen van fase 2.

#### Fase 2: Evalueren van het scenario

1. De CNL-input is voorbereid. Op basis van een scenario zijn relevante bedrijfsregels en een vocabulaire opgesteld in SBVR.
2. De werking van het model is getest door de vocabulaire uit het scenario toe te voegen aan het basismodel en het resultaat te vergelijken met de werkelijkheid. Een test is geslaagd wanneer de toegevoegde data dezelfde uitkomst geven als de werkelijke casus.
3. De referentiearchitectuur is geëvalueerd. In deze stap zijn de traceability-links gevolgd via de werkwijze van de ambtenaar.
4. Als laatste stap is onderzocht of de werkzaamheden van de ambtenaar worden beïnvloed door de mogelijkheid om de gebruikte data te herleiden tot de brondocumenten. Ook is onderzocht hoe de referentiearchitectuur uitgebreid kan worden om verder bij te dragen aan het domein van de wetgeving.

#### Vorbereiden van het model

Het model uit fase 1 vormt de basis van het onderzoek. De eerste stap van fase 1 is het analyseren van de wetsartikelen en bijbehorende beleidsregels. Op basis van deze documenten zijn bedrijfsregels opgesteld. Deze zijn daarna geformaliseerd in een CNL. Als CNL is SBVR-SE gekozen, omdat deze taal ook gebruikt werd in de paper van Rutledge en Italiaander, hoewel de RA4T ook te gebruiken is met een andere CNL (Rutledge & Italiaander, 2021).

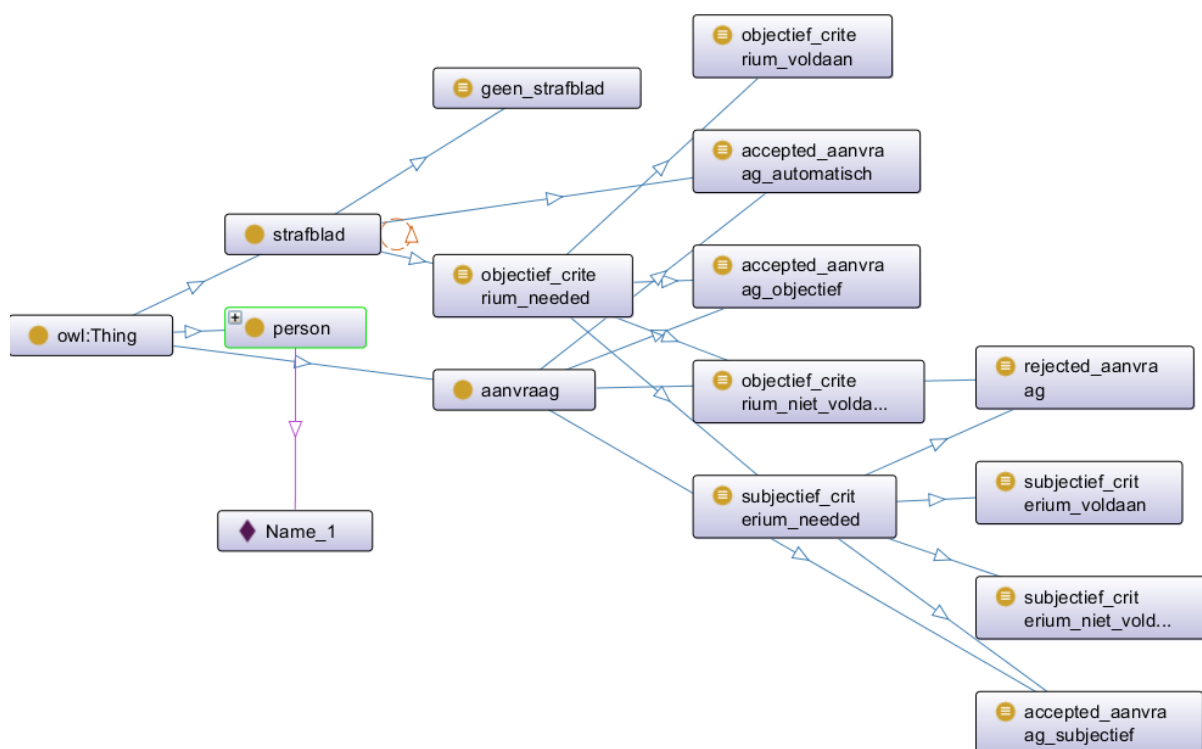
De bedrijfsregels in CNL zijn vervolgens getransformeerd naar een taal die verwerkbaar is door machines. Hiervoor is s2o gebruikt, een converter die SBVR omzet naar OWL 2 (Karpovič, 2014). Deze convertor is een demonstratietool voor de mogelijkheid om SBVR en SBVR-SE automatisch om te zetten naar OWL 2. Het is echter ook mogelijk de SBVR handmatig te converteren naar OWL 2, bijvoorbeeld met het programma Protégé.

De keuze voor s2o is gemaakt omdat dit programma ook door Rutledge en Italiaander is gebruikt (Rutledge & Italiaander, 2021). Daarnaast laat het gebruik van een geautomatiseerde tool beter de mogelijkheden zien van een werkend systeem. Alle entiteiten uit de opgestelde bedrijfsregels zijn als SBVR Business Vocabulary toegevoegd aan s2o. Ook alle opgestelde bedrijfsregels in SBVR zijn als

bedrijfsregels toegevoegd aan s2o. De output is een ontologie in het OWL-2-format. Een ontologie definieert de concepten en relaties binnen het Semantic Web (W3C, 2022).

Deze OWL-2-code is daarna verwerkt in de Semantic Web Ontology Editor in Protégé. Dezelfde editor is ook gebruikt door Rutledge en Italiaander, al zou vergelijkbare software ook gebruikt kunnen worden. Protégé is gekozen omdat dit opensourcesoftware betreft die door de wetenschappelijke gemeenschap veel gebruikt wordt (Cardoso, 2007). Met Protégé kunnen de scenario's getest worden door het toewijzen van *data properties* uit de scenario's aan bepaalde objecten. Dit gebeurt met behulp van de Pellet Reasoner. Deze ingebouwde functie van Protégé genereert conclusies (*inferencing*) gebaseerd op de SBVR-code, met als input de bestaande data properties. Protégé kan via inferencing de conclusies trekken die horen bij de aanwezige properties. Deze inferences worden in Protégé weergegeven als geel gemarkeerde velden. Hiermee kan de uitkomst van een aanvraag gesimuleerd worden. Dit format is voor alle scenario's gebruikt.

De basis-ontologie is opgebouwd rondom drie elementen: aanvraag, strafblad en persoon. Een persoon kan een strafblad hebben. Een aanvraag voor een persoon zonder strafblad wordt automatisch goedgekeurd zoals beschreven in tabel 1. Bij een aanvraag voor een persoon met een strafblad wordt gekeken naar een objectief en een subjectief criterium. De relaties tussen deze objecten wordt weergegeven in figuur 2.



Figuur 2. Objectrelaties gevisualiseerd in de OntoGraf-tab van Protégé.

### Evaluëren van het scenario

Om de RA4T te evalueren zijn per scenario specifieke data uit een scenario gehaald. De data zijn als data properties toegevoegd aan een *instance* in het model. Hiermee wordt een interface gesimuleerd met andere bronnen. Ook vergemakkelijkt dit het wijzigen van de data properties per *individual* om zo de logica achter het systeem te testen. Waar de data in dit onderzoek binair worden weergegeven als 0 of 1, biedt deze wijze ook de mogelijkheid tot het toekennen van zwaarte

aan bepaalde properties, om zo een model te ontwikkelen dat beter overeenkomt de werkelijkheid. Op deze manier is de referentiearchitectuur voor verschillende scenario's geëvalueerd.

Voor de evaluatie van het model is eerst getest of het model werkt zoals verwacht. De nieuwe data properties uit het scenario zijn toegevoegd en er is getest of het model het verwachte resultaat geeft. De resultaten zijn afgeleid door inferencing van Protegé via de Pellet Reasoner.

Na het succesvol testen zijn de traceability-links gevolgd. Ook is nagegaan hoe de traceerbaarheid kan bijdragen aan het uitvoeren van de werkzaamheden, het ontwikkelen van de systemen en het verbeteren van wetteksten.

### 3.4. Validiteit, betrouwbaarheid en ethische aspecten

#### **Validiteit en betrouwbaarheid**

Onderzoeksnaauwkeurigheid is een van de eisen die gesteld worden aan design science research (Hevner et al., 2004). Het is belangrijk dat een onderzoek zowel nauwkeurig als relevant is. Betrouwbaar onderzoek komt voort uit het effectief gebruikmaken van de knowledge base en het gebruik van de juiste methodologie, om het juiste artefact te evalueren. Hier is rekening mee gehouden door het artefact in te delen naar de richtlijnen van Hevner et al. (2004). Een onderzoek is relevant wanneer het bijdraagt aan een oplossing voor een probleem in de organisatie. De RA4T zou verschillende problemen binnen de overheid kunnen oplossen.

Verder is design science iteratief (Hevner et al., 2004). Door continue bijdragen van nieuwe onderzoeken wordt de beschikbare kennis voortdurend vergroot. Dit onderzoek richt zich specifiek op een bepaalde situatie binnen een casus. Met dezelfde methodologie zouden ook andere situaties binnen de casus onderzocht kunnen worden. Ook zou eenzelfde situatie binnen een andere casus volgens dezelfde methodologie uitgevoerd kunnen worden. Door de methodologie binnen dit onderzoek herhaalbaar te maken, kan vervolgonderzoek uitwijzen of dezelfde resultaten gevonden kunnen worden.

#### **Ethische aspecten**

Een ethisch aspect van dit onderzoek is het gebruiken van informatie uit rechtszaken, waarbij geen expliciete toestemming gevraagd is aan de betrokkenen. Door alle informatie uit publieke beschikbare bronnen te halen en geen namen van de betrokkenen in dit onderzoek te gebruiken, wordt zo goed mogelijk rekening gehouden met de privacy. Tevens is de informatie uit de publieke bronnen al geanonimiseerd (Commissie Communicatie Webvoorzieningen, 2021).

## 4. Resultaten

Om de onderzoeksvraag te beantwoorden, is de RA4T geëvalueerd met behulp van scenario's. De evaluatie van de referentiearchitectuur is in dit onderzoek uitgevoerd met de software Protégé.

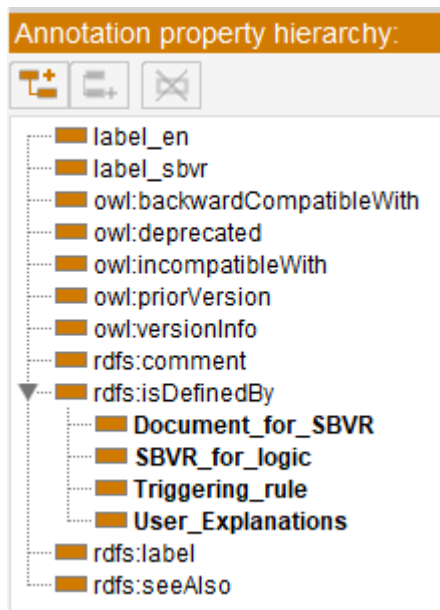
### 4.1. Algemeen scenario

In dit onderzoek is een model gemaakt waarmee de echte software wordt gesimuleerd. Het is daarom van belang om de juiste eigenschappen te gebruiken, zodat ook de echte traceerbaarheid wordt gesimuleerd. Dit onderzoek simuleert traceerbaarheid door in Protégé gebruik te maken van `rdfs:isDefinedBy`. Deze eigenschap kan gebruikt worden om de bron van een onderwerp weer te geven (W3C, 2022). `rdfs:isDefinedBy` wordt gedefinieerd als de definitie van het onderwerp. Dit sluit aan bij de traceability-links vanuit de RA4T, waarmee terug wordt gegaan naar de regels die het onderdeel definiëren. Hiermee wordt de werking van een systeem dat gebaseerd is op traceerbaarheid gesimuleerd in Protégé conform de RA4T. Andere opties binnen Protégé zoals `rdfs:comment`, een door mensen leesbare beschrijving van het onderwerp, en `rdfs:seeAlso`, mogelijk relevante informatie over het onderwerp, zijn minder strikt gebonden aan het onderwerp dan `rdfs:isDefinedBy` en zijn daarom niet gekozen.

De RA4T biedt traceerbaarheid op de volgende manieren:

- User explanations
- Triggering rule
- SBVR for logic
- Document for SBVR

De verschillende traceability-links zijn als sub-property toegevoegd binnen `rdfs:isDefinedBy`, zoals is getoond in figuur 3. De OWL-2-code die hierbij hoort is te zien in tabel 2. Voor elke laag van traceerbaarheid is de relevante informatie toegevoegd als annotatie bij één van de sub-properties in Protégé. De volledige OWL-2-code is te vinden in Appendix 1.



Figuur 3. Traceability-links als sub-property van `rdfs:isDefinedBy`.

De toegevoegde annotaties die behoren bij een automatisch goedgekeurde aanvraag worden in figuur 4 weergegeven.

[Document\\_for\\_SBVR](#) @ x o

Ten behoeve van de beoordeling van een VOG-aanvraag ontvangt het COVOG alle justitiële gegevens betreffende de aanvrager die zijn geregistreerd in het JDS. De justitiële gegevens kunnen zowel uit Nederland als uit het buitenland afkomstig zijn. Aan de aanvrager die niet voorkomt in het JDS, wordt zonder meer een VOG afgegeven.

Bron: Beleidsregels VOG 2018

<https://wetten.overheid.nl/BWBR0040253/2018-01-01>

[SBVR\\_for\\_logic](#) @ x o

accepted\_aanvraag\_automatisch  
 Definition: strafblad that has feiten less\_than\_or\_equal\_to 0  
 General\_concept: aanvraag

[Triggering\\_rule](#) @ x o

[geen\\_strafblad](#)

[User\\_Explanations](#) @ x o

Een VOG aanvraag wordt automatisch goedgekeurd wanneer geen feiten in het JDS gevonden zijn.

Figuur 4. Traceability-links als sub-properties van Rdfs:isDefinedBy voor een automatisch goedgekeurde aanvraag.

Elk scenario heeft dezelfde usercasestory als startpunt. Per scenario worden de afwijkingen toegevoegd aan de usercasestory.

*Algemene usercasestory:*

Een ambtenaar gebruikt het systeem om een VOG-aanvraag te verwerken. Op basis van de persoonsgegevens vult het systeem in dat de aanvrager een vermelding in het Justitieel Documentatie Systeem (JDS) heeft. De vermelding in het JDS wordt door de ambtenaar bekeken en op basis van het objectieve en subjectieve criterium wordt de aanvraag geweigerd.

Een verkorte versie van de SBVR-code wordt weergegeven in tabel 2. De volledige SBVR-code is te vinden in Appendix 1.

Tabel 2. SBVR input in s2o voor het algemene scenario.

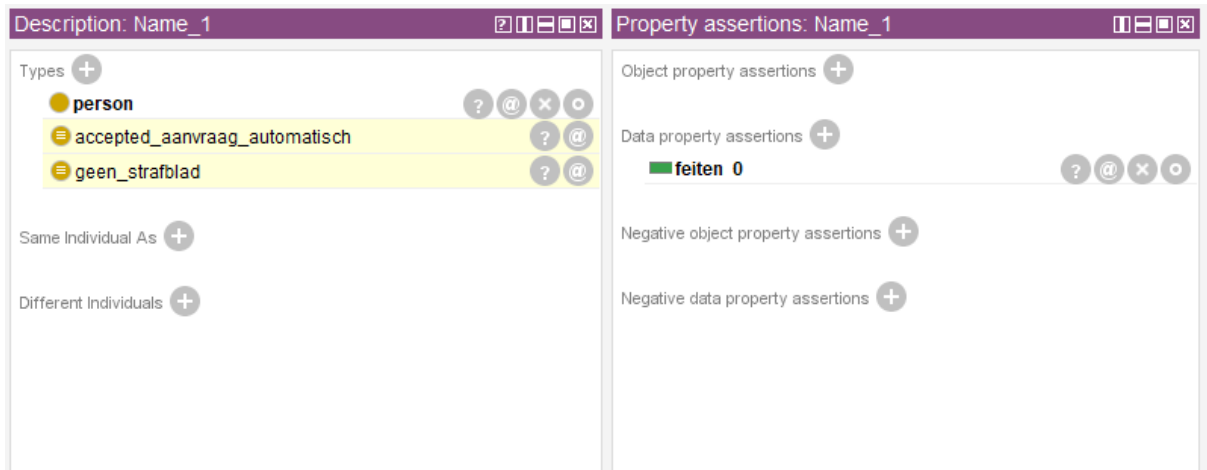
SBVR in s2o: Vocabulary
person strafblad person has strafblad Concept_type: property_association
feiten Concept_type: role General_concept: number
strafblad has feiten Concept_type: property_association
geen_strafblad Definition: strafblad that has feiten less_than_or_equal_to 0 General_concept: strafblad
objectief_criterium_needed Definition: strafblad that has feiten greater_than 0 General_concept: strafblad
aanvraag
accepted_aanvraag_subjectief

Definition: subjectief_criterium_needed that has subjectief less_than_or_equal_to 0 General_concept: aanvraag  accepted_aanvraag_objectief Definition: objectief_criterium_needed that has objectief less_than_or_equal_to 0 General_concept: aanvraag  accepted_aanvraag_automatisch Definition: strafblad that has feiten less_than_or_equal_to 0 General_concept: aanvraag  rejected_aanvraag Definition: subjectief_criterium_needed that has subjectief greater_than 0 General_concept: aanvraag
<b>Additional OWL 2</b>
# Annotation Property: :Document_for_SBVR (:Document_for_SBVR) SubAnnotationPropertyOf(:Document_for_SBVR rdfs:isDefinedBy)  # Annotation Property: :SBVR_for_logic (:SBVR_for_logic) SubAnnotationPropertyOf(:SBVR_for_logic rdfs:isDefinedBy)  # Annotation Property: :Triggering_rule (:Triggering_rule) SubAnnotationPropertyOf(:Triggering_rule rdfs:isDefinedBy)  # Annotation Property: :User_explanations (:User_explanations) SubAnnotationPropertyOf(:User_explanations rdfs:isDefinedBy)

Het algemene model is eerst getest. De test is geslaagd als het model in staat is een aanvraag op de volgende manieren te beoordelen:

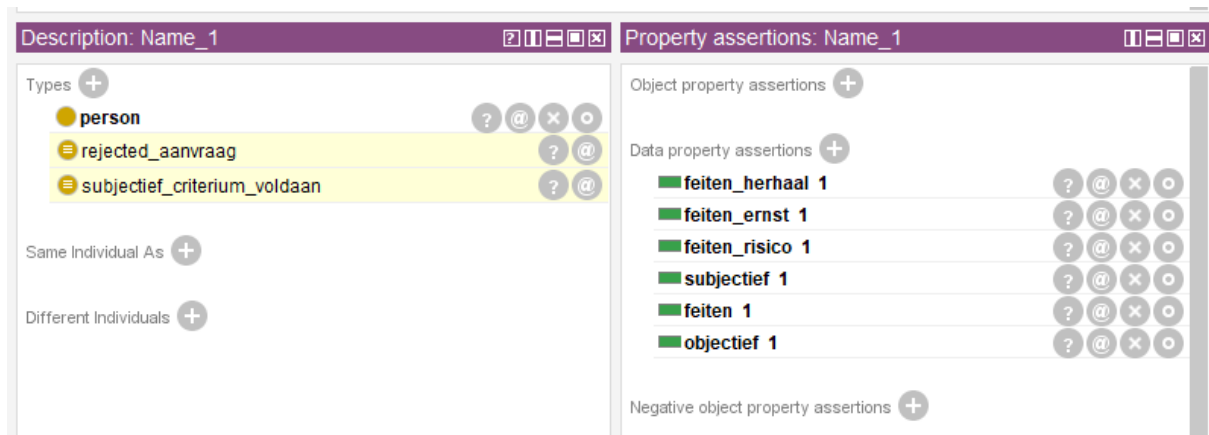
- goedkeuren bij afwezigheid van een strafblad;
- afkeuren wanneer aan het subjectieve criterium is voldaan.

Via de inferences is in figuur 5 te zien dat bij het ontbreken van een strafblad, gedefinieerd als *feiten = 0*, de aanvrager geen strafblad heeft. Dit is zichtbaar als de geel gemarkeerde inferentie *geen\_strafblad*. Ook is de volgende inferentie te zien, *accepted\_aanvraag\_automatisch*. Dit betekent dat de aanvraag automatisch goedgekeurd is. In figuur 6 is te zien dat wanneer er wel feiten zijn gevonden en aan het objectieve en subjectieve criterium wordt voldaan, de aanvraag wordt afgekeurd. Dit is te zien aan de inferentie *rejected\_aanvraag*. De data properties die hiervoor zijn gebruikt, zijn: *feiten = 1*, *feiten\_ernst = 1*, *feiten\_risico = 1*, *feiten\_herhaal = 1*, *objectief = 1*, *subjectief = 1*. Het model gedraagt zich zoals verwacht en kan gebruikt worden als basis voor alle scenario's.



Figuur 5. Bij het ontbreken van een strafblad wordt de aanvraag automatisch goedgekeurd.





Figuur 6. Wanneer aan zowel het objectieve als subjectieve criterium is voldaan, wordt de aanvraag geweigerd.

## 4.2. Scenario: ambtenaar

In dit scenario is onderzocht hoe een systeem gebaseerd op de RA4T kan bijdragen aan de praktische uitvoering van de werkzaamheden van de ambtenaar. Het doel om antwoord te geven op de vragen hoe, wanneer en in hoeverre de referentiearchitectuur eindgebruikers kan helpen bij het begrijpen van de resultaten van het systeem.

### 4.2.1 Voorbereiden van de CNL-input

*Ambtenaar scenario:*

De Raad van State beslist dat de weigering van de VOG onterecht was, omdat één van de vermeldingen in een hoger beroep is vernietigd.

Om het basismodel aan te passen aan het scenario, is de code uit tabel 3 toegevoegd aan het model. Een nieuwe data property voor het testen van het model is: *bijzonderheden = 1*. Met deze property *bijzonderheden = 1* wordt aangegeven dat bij het verwerken van de aanvraag de ambtenaar te zien krijgt dat er mogelijk relevante zaken spelen binnen het dossier. Deze relevante informatie wordt toegevoegd als annotatie aan bijzonderheden via een zelf aangemaakte sub-property van *rdfs:comment*, genaamd *Link\_to\_JDS*, zoals te zien in figuur 9. Hiermee is een interface gesimuleerd naar een annotatie binnen een ander systeem binnen de overheid. Daarnaast is de relevante wetgeving toegevoegd als annotatie aan de data property *bijzonderheden* via *rdfs:isDefinedBy*, om zo de traceability-link *Document for SBVR* te simuleren. Dit is weergegeven in figuur 10.

Tabel 3. Toevoegingen aan het algemeen model voor scenario ambtenaar.

Brondocument: wetgeving
<p><i>"De minister heeft aan de weigering van een VOG onder meer ten grondslag gelegd dat [appellant] in verband staat tot medeplichtigheid aan oplichting. De minister heeft dat verband uitsluitend gebaseerd op het feit dat in het JDS vermeld stond dat [appellant] veroordeeld was wegens medeplichtigheid aan oplichting. Aangezien inmiddels die veroordeling is vernietigd en [appellant] is vrijgesproken van medeplichtigheid aan oplichting, mocht, achteraf bezien, zonder nadere motivering dat delict niet bij de beoordeling worden betrokken. De minister heeft geen nadere motivering gegeven"</i> (Raad van State, 2019).</p>
Scenario SBVR in s2o: Vocabulary
<p>bijzonderheden_aanvraag          Definition: strafblad that has bijzonderheden greater_than 0          General_concept: aanvraag</p>
Additional OWL 2

```

<ns:s2o#feiten> 1 ;
<ns:s2o#feiten_ernst> 1 ;
<ns:s2o#feiten_herhaal> 1 ;
<ns:s2o#feiten_risico> 1 ;
<ns:s2o#bijzonderheden> 1 ;
<ns:s2o#objectief> 1 ;
<ns:s2o#subjectief> 1 ;

```

DataPropertyAssertion(Annotation(rdfs:Link\_to\_JDS "Vermelding in het JDS wordt aangevochten in hoger beroep. Rechtszaak nog bezig."))

# Data Property: <ns:s2o#bijzonderheden> (bijzonderheden)

AnnotationAssertion(:Document\_for\_SBVR <ns:s2o#bijzonderheden> "Het subjectieve criterium ziet op omstandigheden van het geval die ertoe kunnen leiden dat de objectieve vaststelling van een risico voor de samenleving ten aanzien van deze aanvrager niet zou moeten leiden tot een weigering van de afgifte van de VOG.

Omstandigheden van het geval die altijd in de beoordeling worden betrokken zijn:

- de afdoening van de strafzaak;
- het tijdsverloop;
- de hoeveelheid antecedenten.

Indien de aanvrager ten tijde van het plegen van een strafbaar feit minderjarig was, betreft het COVOG dit in de beoordeling van de aanvraag.

Ten behoeve van een goede oordeelsvorming is het COVOG bevoegd inlichtingen in te winnen bij het Openbaar Ministerie en de reclassering."

#### 4.2.2 Testen van het model

Het model is eerst getest. De test is geslaagd wanneer het model in staat is, naast de testeisen uit het basismodel, het volgende weer te geven:

- een melding geven wanneer bijzonderheden aanwezig zijn.

De Pallet Reasoner is gevolgd vanaf *individu Name\_1*. Aan de inference *rejected\_aanvraag* is te zien dat de VOG-aanvraag afgewezen is, omdat aan het subjectieve criterium is voldaan (zie figuur 7). Daarnaast is zichtbaar dat bijzonderheden aanwezig zijn via de inference *bijzonderheden\_aanvraag*.

The screenshot shows a reasoning tool interface with two main panels. The left panel, titled 'Description: Name\_1', shows a list of types under 'Types'. The types listed are 'person', 'bijzonderheden\_aanvraag', 'rejected\_aanvraag', and 'subjectief\_criterium\_voldaan'. The right panel, titled 'Property assertions: Name\_1', shows 'Object property assertions' and 'Data property assertions'. Under 'Data property assertions', there are seven entries, each with a green bar and a value of '1': 'feiten\_herhaal 1', 'feiten\_ernst 1', 'feiten\_risico 1', 'subjectief 1', 'feiten 1', 'bijzonderheden 1', and 'objectief 1'. The 'Object property assertions' section is currently empty.

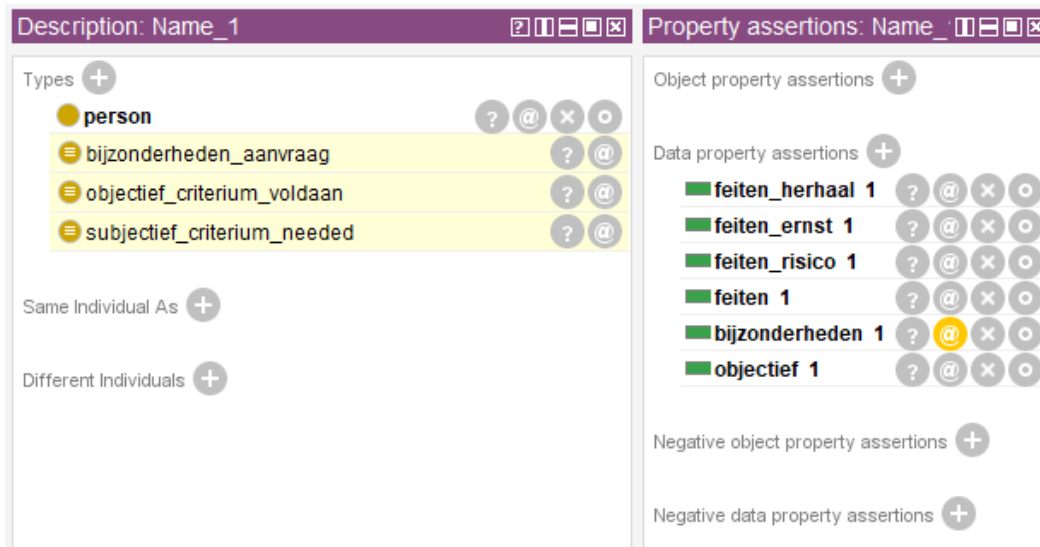
Figuur 7. De VOG-aanvraag is afgekeurd op basis van het subjectieve criterium.

Het model is succesvol getest. Het invullen van de data uit het scenario heeft hetzelfde resultaat opgeleverd als in werkelijkheid. Het model gedraagt zich zoals verwacht. Vervolgens is onderzocht of traceerbaarheid invloed heeft op de beslissingen van de ambtenaar.

#### 4.2.3 Evalueren van de referentiearchitectuur

De data properties voor het testen van het model zijn: *feiten = 1*, *feiten\_ernst = 1*, *feiten\_risico = 1*, *feiten\_herhaal = 1*, *objectief = 1*, *subjectief = 1* en *bijzonderheden = 1*. Deze zijn stapsgewijs toegevoegd om zo het proces van een ambtenaar te simuleren.

De Pallet Reasoner is gevolgd vanaf individu *Name\_1* tot het punt waar het systeem aangeeft dat er bijzonderheden zijn via de inference *bijzonderheden\_aanvraag*, zoals te zien in figuur 8.



Figuur 8. Het systeem herkent dat bijzonderheden aanwezig zijn via de inference *bijzonderheden\_aanvraag*.

De vervolgstappen van de ambtenaar zijn hierna uitgewerkt.

1. Door de annotatie bij *bijzonderheden* te volgen via de geel gemarkeerde @, vindt de ambtenaar de relevante context voor de vermelding in het strafblad zoals te zien in figuur 9.



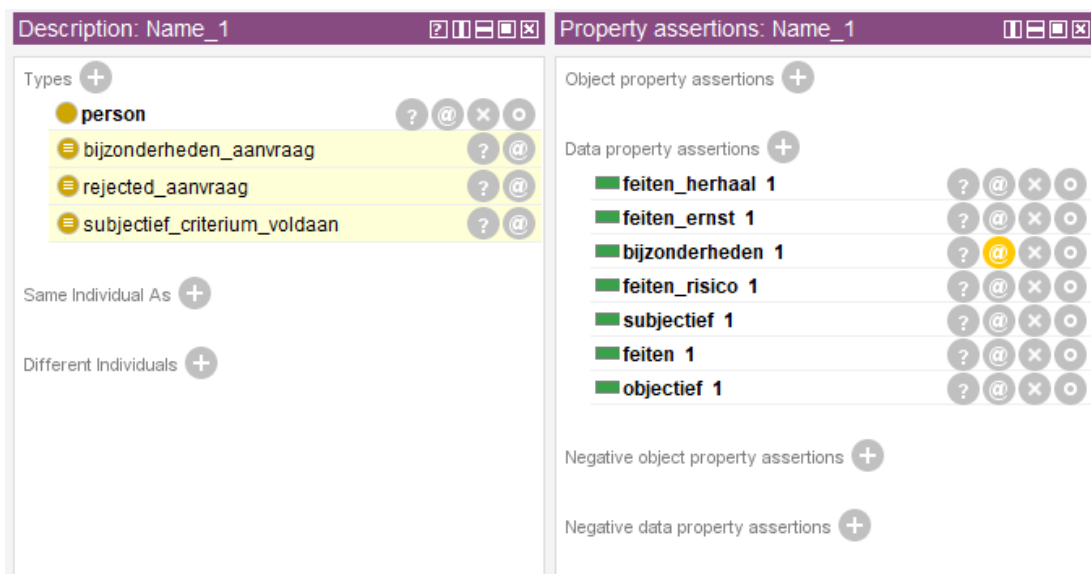
Figuur 9. Bijzonderheden bij de aanvraag, als annotatie van een sub-property van *rdfs:comment*

2. Door de data property *bijzonderheden* te volgen vindt de ambtenaar de relevante wetgeving voor bijzonderheden, zoals te zien in figuur 10.



Figuur 10. Achterliggende wetteksten voor bijzonderheden, als annotaties van sub-properties van `rdfs:isDefinedBy`.

3. De ambtenaar heeft nu alle feiten. De vermelding in het strafblad is aangevochten in een hoger beroep. Uit de relevante wetgeving blijkt dat de afdoening van de strafzaak meegenomen moet worden in de beslissing van de ambtenaar bij het subjectieve criterium. Het hoger beroep is nog gaande. De vermelding zoals nu in het strafblad is nog valide.
4. De ambtenaar beoordeelt dat aan het subjectieve criterium voldaan wordt. De data property `subjectief = 1` wordt toegevoegd aan de aanvraag.
5. De aanvraag wordt afgekeurd zoals te zien aan de inferentie `rejected_aanvraag` in figuur 11.



Figuur 11. De VOG aanvraag is afgekeurd, ondanks de bijzonderheden.

## Conclusie

Traceerbaarheid heeft in dit geval de uitkomst van de aanvraag niet veranderd. Traceerbaarheid heeft er wel voor gezorgd dat de ambtenaar een geïnformeerde keuze heeft kunnen maken. Kijkend naar het moment was de keuze gerechtvaardigd. De uitleg vanuit de annotatie bij de bijzonderheden heeft de ambtenaar getriggert om de lopende casus te bekijken. Dit kan ook gegeneraliseerd worden: om tot een betere kennis over de wetgeving te komen, zou het behulpzaam kunnen zijn voor een ambtenaar om relevante rechtszaken in te kunnen zien als toevoeging aan de originele

brondocumenten. Waar dit in dit scenario gesimuleerd is met de data property *bijzonderheden*, zou dit in een op traceerbaarheid gebaseerd systeem gelinkt kunnen zijn aan de brondocumenten op het niveau van de wetgever binnen de RA4T.

### 4.3. Scenario: ontwikkelaar

In dit scenario is onderzocht wat de voordelen zijn voor ontwikkelaars van een systeem gebaseerd op de RA4T bij de ontwikkeling van systemen die gebruikt worden voor de werkzaamheden van de ambtenaar.

#### 4.3.1 Voorbereiden van de CNL-input

*Ontwikkelaar scenario:*

De Raad van State beslist later dat de weigering onterecht was, omdat reeds eerder een VOG afgegeven was op basis van dezelfde feiten en er onvoldoende onderbouwd was waarom de VOG bij gelijkblijvende feiten nu niet afgegeven kon worden.

Om het basismodel aan te passen aan het scenario is de code uit tabel 4 toegevoegd aan het model.

Tabel 4. Toevoegingen aan het algemeen model voor scenario ontwikkelaar.

Brondocument: wetgeving
<p><i>“Voorts heeft de staatssecretaris bij de beoordeling of zich een risico voor de samenleving voordoet onvoldoende betrokken dat [appellant] in 2007 op basis van dezelfde feiten wel een VOG is afgegeven en hij sindsdien geen strafbare feiten heeft gepleegd. Dat het beleid ten aanzien van zedendelicten in de Beleidsregels 2008 is aangescherpt, maakt niet dat de staatssecretaris niet hoefde te motiveren waarom afgifte van de VOG, gelet op het risico van de samenleving, in 2007 wel verantwoord was, maar dat dit risico thans in onvoldoende mate is afgenomen” (Raad van State, 2013).</i></p>
Bedrijfsregel in natuurlijke taal
R1. Wanneer reeds een VOG afgegeven is op basis van dezelfde feiten, moet er extra onderbouwing zijn bij het subjectieve criterium om de aanvraag af te wijzen.
Scenario SBVR in s2o: Vocabulary
<pre> feiten_extra_informatie   Concept_type: role   General_concept: number   General_concept: feiten  feiten_VOG   Concept_type: role   General_concept: number   General_concept: feiten  strafblad has feiten_VOG   Concept_type: property_association   General_concept: strafblad has feiten  strafblad has feiten_extra_informatie   Concept_type: property_association   General_concept: strafblad has feiten  Subjectief_criterium_voldaan_reeds_VOG   Definition: subjectief_criterium_voldaan that has feiten_VOG greater_than 0   General_concept: subjectief_criterium_voldaan </pre>

```
subjectief_criterium_voldaan_reeds_VOG_voldaan
Definition: subjectief_criterium_voldaan_reeds_VOG that has extra_informatie greater_than 0
General_concept: subjectief_criterium_voldaan_reeds_VOG
```

#### 4.3.2 Testen van het model

Alvorens de referentiearchitectuur te evalueren, is het model getest met de nieuwe eigenschappen. De nieuwe data properties voor het testen van het model zijn *feiten\_VOG* en *feiten\_extra\_informatie*. Met de data property *feiten\_VOG = 1* wordt aangegeven dat reeds eerder een VOG afgegeven is voor dezelfde feiten. Met de data property *feiten\_extra\_informatie = 1* wordt gesimuleerd dat wanneer om extra informatie gevraagd wordt, deze onvoldoende ingevuld wordt om de aanvraag beargumenteerd af te wijzen.

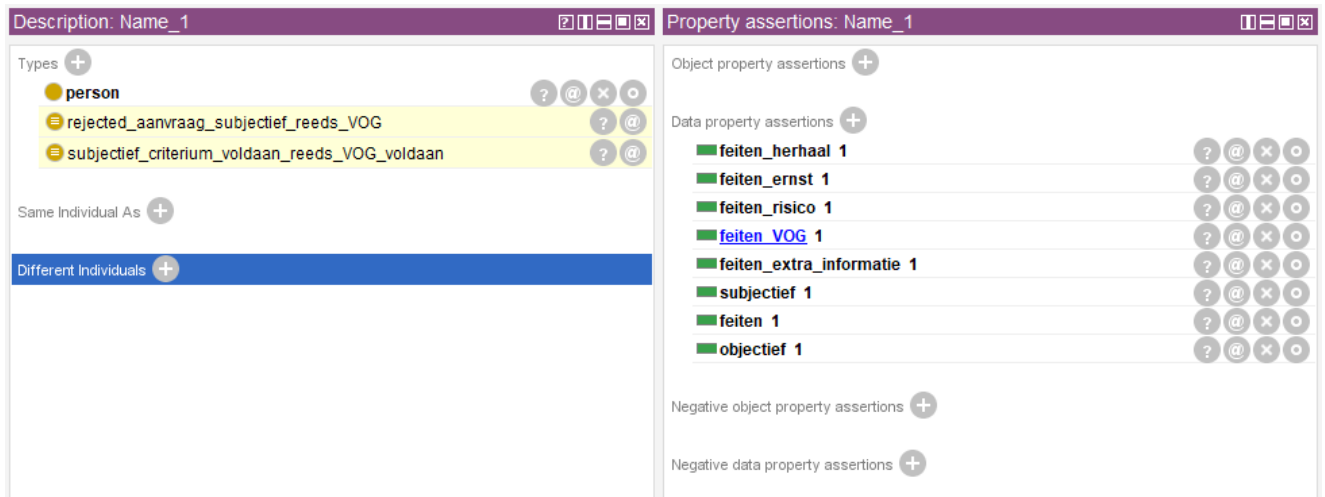
Het model is eerst getest. De test is geslaagd wanneer het model in staat is, naast de testeisen uit het basismodel, het volgende weer te geven:

- een melding geven wanneer er reeds een VOG afgegeven is voor dezelfde feiten;
- de aanvraag afkeuren wanneer er voldoende informatie is om de aanvraag beargumenteerd af te wijzen.

Dee Pallet Reasoner is gevolgd vanaf individu Name\_1. De eerste test is te zien in figuur 12. Na het beoordelen van de aanvraag op het objectieve en subjectieve criterium geeft het systeem aan dat er reeds eerder een VOG is afgegeven met de inference *subjectief\_criterium\_voldaan\_reeds\_VOG*. De tweede test is te zien in figuur 13. Aan de inference *rejected\_aanvraag\_subjectief\_reeds\_VOG* is te zien dat na het invullen van de extra benodigde informatie de aanvraag wordt afgekeurd. Het model gedraagt zich zoals verwacht. Hierna is onderzocht of een systeem met traceerbaarheid invloed heeft op de beslissingen van de ambtenaar.

The screenshot displays the Pallet Reasoner interface. The left pane, titled 'Description: Name\_1', shows a list of types: 'person' and 'subjectief\_criterium\_voldaan\_reeds\_VOG'. The right pane, titled 'Property assertions: Name\_1', shows a list of data property assertions for 'Name\_1'. The assertions are: 'feiten\_herhaal 1', 'feiten\_ernst 1', 'feiten\_risico 1', 'feiten\_VOG 1', 'subjectief 1', 'feiten 1', and 'objectief 1'. Each assertion has a set of control icons (question mark, at-sign, cross, circle) to its right.

Figuur 12. Het systeem geeft een melding dat er reeds een VOG is afgegeven voor dezelfde feiten.



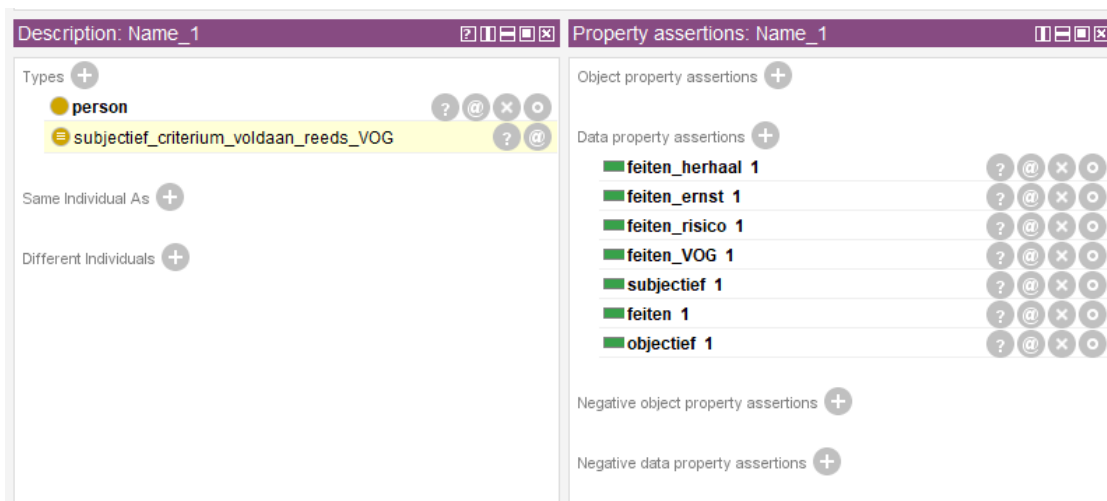
Figuur 13. Het systeem keurt een aanvraag af nadat er voldoende extra informatie wordt gegeven.

#### 4.3.3 Evalueren van de referentiearchitectuur

De data properties voor het testen van het model zijn:  $feiten = 1$ ,  $feiten_ernst = 1$ ,  $feiten_risico = 1$ ,  $feiten_herhaal = 1$ ,  $objectief = 1$ ,  $subjectief = 1$ ,  $feiten_VOG = 1$ ,  $feiten_extra_informatie = 1$ .

Met de data property  $feiten_VOG = 1$  wordt aangegeven dat er eerder een VOG afgegeven is. Met de data property  $feiten_extra_informatie = 0$  wordt gesimuleerd dat wanneer om extra informatie wordt gevraagd, deze onvoldoende ingevuld wordt om de aanvraag beargumenteerd af te wijzen.

De Pallet Reasoner is gevolgd vanaf individu *Name\_1* tot het punt waar het systeem aangeeft dat, hoewel wordt voldaan aan het subjectieve criterium, er reeds eerder een VOG is afgegeven. In figuur 14 is dit te zien aan de inference *subjectief\_criterium\_voldaan\_reeds\_VOG*.



Figuur 14. Het systeem geeft aan dat er reeds eerder een VOG afgegeven is.

De ambtenaar volgt deze melding via de inference *subjectief\_criterium\_voldaan\_reeds\_VOG* en ziet de user explanations, triggering rule, SBVR-logica en brondocumentatie, zoals weergegeven in figuur 15. Door de casus te vergelijken met de brondocumenten waarop de logica is gebaseerd, in dit geval een uitspraak van de Raad van State, ziet de ambtenaar de overeenkomsten tussen de casussen en weet dat extra motivatie nodig is om de aanvraag af te wijzen.

**Document\_for\_SBVR** @ x o

"Voorts heeft de staatssecretaris bij de beoordeling of zich een risico voor de samenleving voordoet onvoldoende betrokken dat [appellant] in 2007 op basis van dezelfde feiten wel een VOG is afgegeven en hij sindsdien geen strafbare feiten heeft gepleegd. Dat het beleid ten aanzien van zedendelicten in de Beleidsregels 2008 is aangescherpt, maakt niet dat de staatssecretaris niet hoefde te motiveren waarom afgifte van de VOG, gelet op het risico van de samenleving, in 2007 wel verantwoord was, maar dat dit risico thans in onvoldoende mate is afgenomen." (Raad van State, 2013)

---

**SBVR\_for\_logic** @ x o

subjectief\_criterium\_voldaan\_reeds\_VOG\_voldaan  
 Definition: subjectief\_criterium\_voldaan\_reeds\_VOG that has extra\_informatie greater\_than 0  
 General\_concept: subjectief\_criterium\_voldaan\_reeds\_VOG

---

**Triggering\_rule** @ x o

■ [feiten\\_extra\\_informatie](#)

---

**User\_Explanations** @ x o

Wanneer reeds een VOG afgegeven is, op basis van dezelfde feiten, moet er extra onderbouwing zijn bij het subjectieve criteria om de aanvraag af te wijzen.

Figuur 15. De achterliggende logica voor het aanleveren van extra informatie, als annotaties van sub-properties van *rdfs:isDefinedBy*.

De ambtenaar bepaalt dat er onvoldoende extra informatie is en vult de data property *feiten\_extra\_informatie* = 0 in. Dit leidt tot het accepteren van de VOG-aanvraag zoals te zien in figuur 16 aan de inferentie *accepted\_aanvraag\_subjectief\_reeds\_VOG*.

The screenshot shows a logic editor interface with two main panels. The left panel, titled 'Description: Name\_1', shows a list of types under a '+' icon. The types listed are 'person', 'accepted\_aanvraag\_subjectief\_reeds\_VOG', and 'subjectief\_criterium\_voldaan\_reeds\_VOG\_niet\_voldaa'. The right panel, titled 'Property assertions: Name\_1', shows a list of data property assertions under a '+' icon. The assertions listed are: 'feiten\_herhaal 1', 'feiten\_ernst 1', 'feiten\_risico 1', 'feiten\_VOG 1', 'feiten\_extra\_informatie 0', 'subjectief 1', 'feiten 1', and 'objectief 1'. Each assertion has a green square icon to its left and a set of control icons (question mark, at-sign, x, o) to its right. Below the data property assertions, there are sections for 'Negative object property assertions' and 'Negative data property assertions', both with '+' icons.

Figuur 16. De VOG aanvraag wordt goedgekeurd.

## Conclusie

Geconcludeerd kan worden dat de mogelijkheid om eerdere uitspraken in te zien die relevant zijn voor de aanvraag, het werk van de ambtenaar kunnen vergemakkelijken. De ambtenaar is beter in staat de bestaande wetgeving te combineren met recente uitspraken, om mogelijk toekomstige rechtszaken te voorkomen. Wanneer uit zulke uitspraken bedrijfsregels naar voren komen, kan de ontwikkelaar deze toevoegen aan de bestaande logica.



Voortbordurend op de bevindingen uit het eerste scenario, waar relevante rechtszaken kunnen bijdragen aan de kennis van de ambtenaar, wordt voorgesteld om additionele informatie via specifieke sub-properties van `rdfs:comment` te implementeren. Waar `rdfs:isDefinedBy` de bron weergeeft waarmee het onderwerp gedefinieerd wordt, kan `rdfs:comment` gebruikt worden om extra toelichting en voorbeelden te geven. In het kader van wetgeving zouden dit verwante uitspraken kunnen zijn. Hierdoor kan de wetgeving geplaatst worden in de context van praktijksituaties met uitspraken. Dit vergroot de kennis van de ambtenaar over de wetgeving en verlaagt het risico van het afwijzen van een aanvraag op basis van eerder verworpen logica.

Om aan de aanbevelingen te voldoen, moet de ontwikkelaar in staat zijn om in updates van de software recente uitspraken mee te nemen en deze te linken aan alle bestaande logica. Hiervoor is het belangrijk dat de infrastructuur voor traceerbaarheid goed is opgezet. Een systeem dat is opgezet volgens de RA4T zou aan deze eisen kunnen voldoen.

## 4.4. Scenario: wetgever

In dit scenario is onderzocht hoe een systeem gebaseerd op de RA4T de wetgever kan helpen bij het ontwikkelen en bijhouden van de wetten die de basis vormen van de logica achter het systeem.

### 4.4.1 Voorbereiden van de CNL-input

*Wetgever scenario:*

De Raad van State beslist later dat de weigering onterecht was, omdat het aangehouden screeningsprofiel voor de functie onvoldoende passend was, gezien de inhoud van de functie en de eisen die aan de functie worden gesteld.

Om het basismodel aan te passen aan het scenario is de code uit tabel 5 toegevoegd aan het model. Uit de uitspraak van de Raad van State volgt een nieuwe bedrijfsregel.

Tabel 5. Toevoegingen aan het algemeen model voor scenario wetgever.

Brondocument: wetgeving
<p><i>“De rechtbank heeft ten onrechte niet ingezien dat [appellant] een VOG voor de functie ambieert omdat hij door zijn opdrachtgevers wordt gevraagd vanwege zijn ervaring als gedetineerde. Hij zal binnen DJI ook werkzaam zijn als ervaringsdeskundige. Het gaat hier dus niet om een ‘gewone’ penitentiair medewerker, waarvan wordt verwacht dat deze aan hogere integriteitseisen voldoet.</i></p> <p>...</p> <p><i>En hoewel een functie bij de DJI hoge integriteitseisen kent, zoals de minister ter zitting ook een aantal keer heeft benadrukt, gaat het in de situatie van [appellant] nu juist om zijn expertise als (ex) gedetineerde. Ook op dat punt heeft de minister geen motivering in zijn besluit gegeven.”</i> (Raad van State, 2021).</p>
Bedrijfsregel in natuurlijke taal
R1. VOG-aanvraag is niet geldig wanneer het screeningsprofiel niet past bij de uit te oefenen functie.
Scenario SBVR in s2o: Vocabulary
<p>functie              Concept_type: role              General_concept: number</p> <p>aanvraag has functie              Concept_type: property_association</p> <p>screeningsprofiel              Concept_type: role              General_concept: number              General_concept: functie</p> <p>aanvraag has screeningsprofiel              Concept_type: property_association              General_concept: aanvraag has functie</p> <p>invalid_aanvraag_profiel              Definition: aanvraag that has screeningsprofiel greater_than 0 and has functie less_than_or_equal_to 0              General_concept: aanvraag</p>
Additional OWL 2

```

<ns:s2o#feiten> 1 ;
<ns:s2o#feiten_VOG> 1 ;
<ns:s2o#feiten_ernst> 1 ;
<ns:s2o#feiten_extra_informatie> 0 ;
<ns:s2o#feiten_herhaal> 1 ;
<ns:s2o#feiten_risico> 1 ;
<ns:s2o#objectief> 1 ;
<ns:s2o#subjectief> 1 ;

```

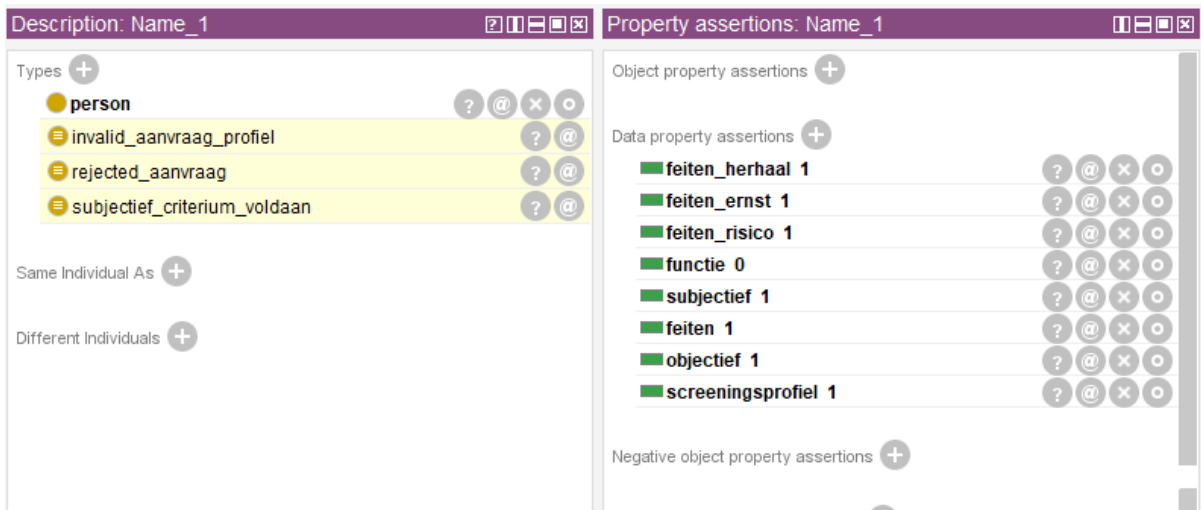
#### 4.4.2 Testen van het model

Alvorens de referentiearchitectuur te evalueren, is het model getest met de nieuwe eigenschappen. De nieuwe data properties voor het testen van het model zijn functie en screeningsprofiel. Met *functie = 1* wordt aangegeven dat de aanvraag voor functie 1 is. Met de data property *screeningsprofiel = 1* wordt gesimuleerd dat het geldende screeningsprofiel voor deze functie 1 is. Het model is eerst getest. De test is geslaagd wanneer het model in staat is, naast de testeisen uit het basismodel, het volgende weer te geven:

- als functie en functieprofiel gelijk zijn, is de aanvraag valide en wordt de aanvraag afgekeurd;
- als functie en functieprofiel van elkaar afwijken, is de aanvraag niet valide.

De Pallet Reasoner is gevolgd vanaf individu *Name\_1*. Voor de eerste test is de data property *functie = 1* toegevoegd. Voor de tweede test is de data property *functie = 0* toegevoegd. De uitkomsten hiervan zijn te zien in figuur 17 en figuur 18. Als de functie en het screeningsprofiel met elkaar overeenkomen, is te zien aan de inference *rejected\_aanvraag* dat de aanvraag wordt afgekeurd. Als de functie afwijkt van het functieprofiel geeft de inference *invalid\_aanvraag\_profiel* weer dat de aanvraag niet geldig is. Het model gedraagt zich in beide situaties zoals verwacht. Vervolgens is onderzocht of een systeem met traceerbaarheid invloed heeft op de beslissingen van de ambtenaar.

Figuur 17. De VOG-aanvraag wordt terecht afgekeurd bij een gelijke functie en screeningsprofiel.

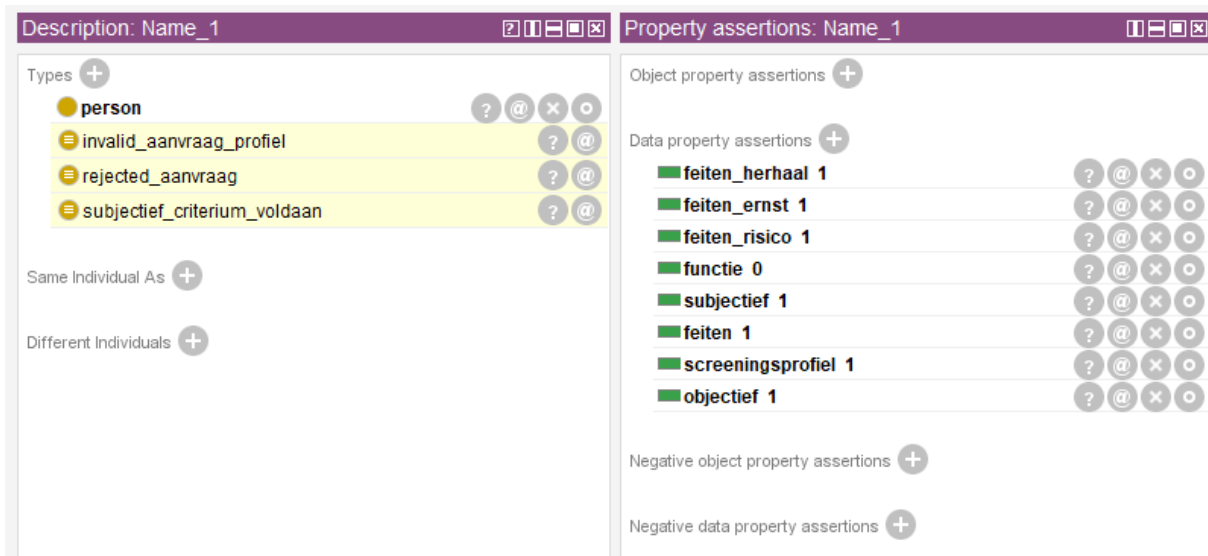


Figuur 18. De VOG aanvraag is niet valide bij een afwijkende functie en screeningsprofiel, ook wanneer aan de overige criteria is voldaan.

#### 4.4.3 Evalueren van de referentiearchitectuur

De data properties voor het testen van het model zijn: *feiten* = 1, *feiten\_ernst* = 1, *feiten\_risiko* = 1, *feiten\_herhaal* = 1, *objectief* = 1, *subjectief* = 1, *functie* = 0 en *screeningsprofiel* = 1.

De Pallet Reasoner is gevolgd vanaf individu *Name\_1* tot het punt waar het systeem aangeeft dat de aanvraag niet geldig is, hoewel er aan het subjectieve criterium wordt voldaan. Dit wordt weergegeven in figuur 19 met de inference *invalid\_aanvraag\_profiel*.



Figuur 19. De VOG aanvraag is niet valide.

De ambtenaar volgt de inference *invalid\_aanvraag\_profiel* en ziet de user explanations, triggering rule, SBVR-logica en brondocumentatie zoals te zien in figuur 20. Door de casus te vergelijken met de brondocumenten waarop de logica is gebaseerd, in dit geval een uitspraak van de Raad van State, ziet de ambtenaar dat wanneer het screeningsprofiel niet overeenkomt met de functie, de aanvraag niet geweigerd kan worden.

#### Document\_for\_SBVR



"De rechtbank heeft ten onrechte niet ingezien dat [appellant] een VOG voor de functie ambieert omdat hij door zijn opdrachtgevers wordt gevraagd vanwege zijn ervaring als gedetineerde. Hij zal binnen DJI ook werkzaam zijn als ervaringsdeskundige. Het gaat hier dus niet om een 'gewone' penitentiair medewerker, waarvan wordt verwacht dat deze aan hogere integriteitseisen voldoet. ... En hoewel een functie bij de DJI hoge integriteitseisen kent, zoals de minister ter zitting ook een aantal keer heeft benadrukt, gaat het in de situatie van [appellant] nu juist om zijn expertise als (ex) gedetineerde. Ook op dat punt heeft de minister geen motivering in zijn besluit gegeven." (Raad van State, 2021)

#### SBVR\_for\_logic



invalid\_aanvraag\_profiel

Definition: aanvraag that has screeningsprofiel greater\_than 0 and has functie less\_than\_or\_equal\_to 0

General\_concept: aanvraag

#### Triggering\_rule



aanvraag

and (functie some xsd:decimal[<= 0])

and (screeningsprofiel some xsd:decimal[> 0])

#### User\_explanations



VOG-aanvraag is niet geldig wanneer het screeningsprofiel niet past bij de uit te oefenen functie.

*Figuur 20. De achterliggende logica voor een invalide aanvraag, als annotatie van sub-properties van `rdfs:isDefinedBy`.*

De ambtenaar kan de aanvraag nu terugsturen naar de aanvrager met het verzoek om het juiste screeningsprofiel in te stellen. Dit verkleint de kans dat een aanvraag ten onrechte afgekeurd wordt, waardoor toekomstige disputen voorkomen kunnen worden.

## Conclusie

Een systeem gebaseerd op traceerbaarheid waar nieuwe uitspraken dan wel bedrijfsregels aan toegevoegd kunnen worden, kan de wetgever helpen iteratief tot steeds betere wetten te komen, die verwerkt kunnen worden in het systeem. Met elke toegevoegde uitspraak kunnen toekomstige rechtszaken voorkomen worden, doordat het systeem de situatie herkent en de ambtenaar hierover informeert, en omdat de beslissingen die volgen uit de logica van het systeem steeds accurater worden. Dit kan bijdragen aan het verlagen van de werkdruk in andere lagen van de overheid en de rechterlijke macht. Waar het tweede scenario tot voordelen voor de ontwikkelaar leidde, kan nu ook geconcludeerd worden dat er voordelen zijn op het niveau van de wetgever binnen de RA4T.

## 5. Conclusie, discussie en aanbevelingen

In dit hoofdstuk worden conclusies getrokken over de RA4T op basis van de uitgewerkte illustratieve scenario's in Protégé. Naast specifieke conclusies binnen de justitiële context wordt ook de generaliseerbaarheid van de conclusies besproken. Daarnaast worden aanbevelingen voor vervolgonderzoek gedaan. Ten slotte wordt op de kwaliteit van dit onderzoek gereflecteerd.

### 5.1. Conclusie

Centraal in dit onderzoek staan de vragen hoe, wanneer en in hoeverre de referentiearchitectuur voor traceerbaarheid kan bijdragen aan het uitvoeren van ambtelijke werkzaamheden op het niveau van de wetgever, de ontwikkelaar en de ambtenaar. Deze vragen zijn beantwoord met behulp van illustratieve scenario's.

#### **Hoe**

Geconcludeerd kan worden dat een systeem gebaseerd op de RA4T op drie niveaus een positieve bijdrage kan leveren aan de uitvoering van ambtelijke werkzaamheden, namelijk:

- het uitvoeringsniveau van de ambtenaar;
- het ontwikkelniveau van de ontwikkelaar;
- het wetgevingsniveau van de wetgever.

Een op traceerbaarheid gebaseerd systeem kan de kennis over wetgeving van ambtenaren vergroten. De ambtenaar is hiermee beter in staat om geïnformeerde beslissingen te nemen die met name betrekking hebben op subjectieve criteria. Voor zowel de ontwikkelaar als de wetgever zijn er voordelen tijdens de ontwikkeling van de software, doordat de werking van de algoritmen gecontroleerd kan worden.

#### **Wanneer**

Een systeem gebaseerd op de RA4T kan in verschillende fasen positief bijdragen aan de werkzaamheden van verschillende gebruikers. Voor de ambtenaar is dit tijdens het uitvoeren van de handelingen, door meer inzicht te krijgen in de achterliggende wetgeving. Voordelen voor de ontwikkelaar en de wetgever bevinden zich zowel in de ontwikkelfase als de onderhoudsfase. Door tijdens het ontwikkelen de resultaten terug te volgen naar de achterliggende brondocumenten, kan door de ontwikkelaar en de wetgever gecontroleerd worden of de algoritmen werken zoals ze zijn bedoeld. Wanneer tijdens het onderhoud van het systeem nieuwe uitspraken via bedrijfsregels toegevoegd kunnen worden, kan het systeem meegroeien met de lezing en ontwikkeling van de wetten. De resultaten van het systeem kunnen op deze manier gevolgd worden naar de verschillende achterliggende regels. Dit kan ertoe leiden dat de wetgever bepaalde wetten moet herzien om onbedoelde gevolgen te voorkomen.

#### **In hoeverre**

Op basis van dit onderzoek blijkt dat traceerbaarheid een oplossing kan zijn voor enkele kritiekpunten die geïdentificeerd zijn door adviescommissies binnen de overheid. Deze punten zijn in te delen in technische en menselijke punten. Een systeem dat gebaseerd is op de RA4T kan bijdragen aan een verbetering van beide soorten. Zo kan traceerbaarheid bijdragen aan het controleren van de uitkomsten van een systeem met de wettelijke regelgeving die erachter zit, door inzichtelijk te maken op basis van welke data, bedrijfsregels en brondocumenten het systeem tot een conclusie is gekomen. Hiermee wordt duidelijk of een algoritme functioneert zoals het is bedoeld. Ook is inzichtelijk hoe de wetteksten vertaald zijn naar computertaal, doordat de SBVR-regels door zowel

mens als machine gelezen kunnen worden. Voor ambtenaren is het mogelijk om in te zien op basis van welke data de beslissing is genomen door het volgen van de traceability-links. Ook neemt de kennis van ambtenaren over achterliggende wetteksten en uitspraken toe, wanneer deze ingezien kunnen worden door de traceability-links te volgen vanaf de beslissing.

## 5.2. Discussie

De referentiearchitectuur voor traceerbaarheid zoals gepresenteerd door Rutledge en Italiaander beschrijft mogelijke voordelen voor de drie lagen binnen de architectuur: de eindgebruiker, de ontwikkelaar en de brontekstschrijver. Voor alle drie de lagen zijn voordelen gevonden in dit onderzoek. De resultaten uit dit onderzoek zijn echter wel beïnvloed door bepaalde keuzes die gemaakt zijn.

De RA4T is in dit onderzoek geëvalueerd op basis van illustratieve scenario's binnen een zelfontwikkeld model dat een gedeelte van de werkelijkheid nabootst, binnen een bepaald domein met één casus. De conclusies die getrokken zijn in dit onderzoek kunnen niet zonder meer gegeneraliseerd worden, omdat deze gebaseerd zijn op de resultaten binnen één domein met één casus. Om de conclusies uit dit onderzoek te generaliseren, moeten deze vergeleken worden met andere onderzoeken die de RA4T evalueren. Binnen de afstudeerkring wordt de RA4T geëvalueerd binnen andere domeinen met casussen. Een meta-analyse van de resultaten uit die onderzoeken zou kunnen leiden tot algemene conclusies en richtingen voor vervolgonderzoek.

De scenario's in dit onderzoek zijn gebaseerd op rechterlijke uitspraken. Echter, slechts delen van deze rechtspraken zijn verwerkt in het model. Voor het valideren van de resultaten uit dit onderzoek kan het interessant zijn om een volledige rechtszaak te verwerken in een model en dit te evalueren. Hierdoor kunnen de handelingen van de ambtenaar beter gesimuleerd worden.

Binnen dit onderzoek is gebruikgemaakt van s2O. Hoewel deze tool ook gebruikt wordt in de paper waarin de referentiearchitectuur wordt gepresenteerd, zijn er ook alternatieven beschikbaar.

S2O is een *web-based conversion tool* die gecreëerd is als onderdeel van een PhD-onderzoek van Jaroslav Karpovič. Het betreft een prototype, waar niet alle SBVR- of OWL-2-concepten in zitten. Een onderzoek waarbij gebruik wordt gemaakt van de volledige mogelijkheden van SBVR of OWL 2 zou tot andere resultaten kunnen leiden.

## 5.3. Aanbevelingen voor praktijk- en vervolgonderzoek

Tijdens het ontwikkelen van het model zijn meerdere mogelijkheden gevonden die buiten de scope van dit onderzoek lagen. Om de resultaten uit dit onderzoek robuuster te maken, zouden deze mogelijkheden in een vervolgonderzoek toegepast kunnen worden. Zo zou een bepaalde zwaarte meegegeven kunnen worden aan de feiten in het strafblad, om hiermee een meer realistische en complexere situatie na te bootsen om het handelen van ambtenaren beter te simuleren. Op die manier kan onderzoek gedaan worden naar de haalbaarheid van een praktische implementatie van het systeem.

Daarnaast zijn enkele mogelijkheden voorgesteld waarmee de RA4T beter zou aansluiten op het onderzochte domein. Om de context rondom wetten te vergroten, zou de RA4T naast traceerbaarheid uitgebreid kunnen worden met voorbeelden uit de praktijk. Binnen een evaluatiemodel dat is opgesteld in OWL 2 zou dit kunnen door het toevoegen van specifieke sub-properties van `rdfs:comment` als optie om relevante scenario's toe te voegen.

## 5.4. Reflectie

In dit onderzoek is gekozen voor de onderzoeksmethode design science research. De richtlijnen voor deze methode vereisen dat de bruikbaarheid, kwaliteit en effectiviteit strikt aangetoond moeten worden (Hevner et al., 2004). Hoewel dit onderzoek de bruikbaarheid, kwaliteit en effectiviteit van een systeem gebaseerd op de RA4T aantoont, zijn de conclusies domein-afhankelijk en niet altijd te generaliseren zonder vervolgonderzoek. Er is gekozen voor een beschrijvende evaluatiemethode met behulp van scenario's. Scenario's zijn in dit onderzoek verwerkt tot een werkbaar model. Er zit een overlap tussen een beschrijvende methode en een experimentele methode. Voor deze laatste methode is niet gekozen, omdat dit buiten de scope van het onderzoek ligt. Voor een vervolgonderzoek zou een volledig werkend systeem gebouwd kunnen worden volgens dezelfde stappen die in dit onderzoek zijn gevolgd. Dit zou tot hardere conclusies kunnen leiden binnen het domein.



## 6. Bibliography

- Beleidsregels VOG-NP-RP 2018. (2022, 03 07). Opgehaald van Beleidsregels VOG-NP-RP 2018: <https://wetten.overheid.nl/BWBR0040253/2018-01-01>
- Booth, A. (2008). Unpacking your literature search toolbox: On search styles and tactics. *Health Information and Libraries Journal*.
- Bos, P. (2013). *Bedrijfsregels in verschillende vormen*.
- Business Rule Group. (2000). Principles of the business rule approach. *International Journal of Information Management*.
- Business Rules Group. (2003). *The Business Rules Manifesto*.
- Cardoso, J. (2007). The Semantic Web Vision: Where are We? *IEEE Intelligent Systems*.
- Commissie Communicatie Webvoorzieningen. (2021). *Anonimiseringsrichtlijnen*. Opgehaald van <https://www.rechtspraak.nl/Uitspraken/Paginas/Anonimiseringsrichtlijnen.aspx>
- Donner, J., den Ouden, W., Klijnsma, J., Akdemir, G., & Gosen, C. (2020). *Omzien in verwondering 2. Eindadvies*.
- Dybå, T., & Dingsøy, T. (2009). What do we know about Agile Software. *IEEE Software*.
- Feteris, M. (2018). Signalen van de Hoge Raad naar de wetgever. *RegelMaat*.
- Gordon, T. F., Governatori, G., & Rotolo, A. (2009). *Rules and norms: Requirements for rule interchange languages in the legal domain*.
- Hendrickx, F. (2021, January 15). *De Volkskrant*. Opgehaald van Rutte III treedt af: 'We hebben gefaald in de bescherming van de zwakken': <https://www.volkskrant.nl/nieuws-achtergrond/rutte-iii-treedt-af-we-hebben-gefaald-in-de-bescherming-van-de-zwakken~b2e128de/?referrer=https%3A%2F%2Fwww.google.com%2F>
- Hevner, A., March, S., Park, J., & Ram, S. (2004). *Design Science in IS Research*. MIS Quarterly.
- International institute of Business Analysis. (2015). *A guide to the business analysis body of knowledge*.
- International institute of Business Analysis. (2015). *A guide to the business analysis body of knowledge*. Toronto: IBA.
- Joosten, S. (2007). *Rule based design*.
- Karpovič, J. (2014). *s2o: SBVR to OWL 2 Converter - description*. Opgehaald van s2o: SBVR to OWL 2 Converter: <https://s2o.isd.ktu.lt/about.php>
- Krüger, J., Alkl, G., Berger, T., Leich, T., & Saake, G. (2019). Effects of explicit feature traceability on program comprehension. *ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*.
- Mäder, P., & Egyed, A. (2015). Do developers benefit from requirements traceability when evolving and maintaining a software system? *Empirical Software Engineering*.

- Open Management Group. (2013). *Semantics of Business Vocabulary and Business Rules (SBVR), V1.2 Annex G - EU-Rent Example*.
- Peffers, K., Rothenberger, M., Tuunanen, T., & Vaczi, R. (2012). *Design Science Research in Information Systems. Advances in Theory and Practice* (Vol. 7286). Opgehaald van <http://www.springerlink.com/index/10.1007/978-3-642-29863-9>
- Pinheiro, F. (2004). Requirements Traceability. *The Springer International Series in Engineering and Computer Science*.
- Raad van State. (2013). *ECLI:NL:RVS:2013:13*. Opgehaald van de Rechtspraak: <https://uitspraken.rechtspraak.nl/inziendocument?id=ECLI:NL:RVS:2013:13>
- Raad van State. (2019). *ECLI : NL : RVS : 2019 : 2539*. Opgehaald van de Rechtspraak: <https://uitspraken.rechtspraak.nl/inziendocument?id=ECLI:NL:RVS:2019:2539>
- Raad van State. (2021). *ECLI : NL : RVS : 2021 : 1855*. Opgehaald van de Rechtspraak: <https://uitspraken.rechtspraak.nl/inziendocument?id=ECLI:NL:RVS:2021:1855>
- Ross, R. (2008). Business Rules from A –Z: What You Need to Know. . *Business Rules Journal*.
- Rutledge, L., & Italiaander, R. (2021). Toward a Reference Architecture for Traceability in SBVR-based systems. *Proceedings of the Seventh International Workshop on Controlled Natural Language*.
- Schwitter, R., & Tilbrook, M. (2004). Controlled natural language meets the semantic web. *Proceedings of the Australasian Language Technology Workshop*.
- Staten-generaal, T. (2018). *Advies Afdeling advisering Raad van State en Nader rapport inzake de effecten van digitalisering voor de rechtsstatelijke verhoudingen*. Den Haag.
- W3C. (2022, 31 05). *RDF Schema 1.1*. Opgehaald van W3: <https://www.w3.org/TR/rdf-schema/>
- W3C. (2022). *Vocabularies*. Opgehaald van W3C: <https://www.w3.org/standards/semanticweb/ontology>
- Wet justitiële en strafvorderlijke gegevens. (2002). *Wet justitiële en strafvorderlijke gegevens 2002*.
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *ACM International Conference Proceeding Series*.

## 7. Appendix 1

SBVR in s2o: Vocabulary voor het basismodel
person
Name_1 General_concept: person
Name_2 General_concept: person
strafblad
person has strafblad Concept_type: property_association
strafblad contains strafblad Concept_type: partitive_verb_concept Concept_type: transitive_verb_concept
feiten
Concept_type: role General_concept: number
strafblad has feiten Concept_type: property_association
feiten_ernst
Concept_type: role General_concept: number General_concept: feiten
feiten_herhaal
Concept_type: role General_concept: number General_concept: feiten
feiten_risico
Concept_type: role General_concept: number General_concept: feiten
subjectief
Concept_type: role General_concept: number General_concept: feiten
objectief
Concept_type: role General_concept: number General_concept: feiten
strafblad has feiten_ernst Concept_type: property_association General_concept: strafblad has feiten
strafblad has subjectief Concept_type: property_association General_concept: strafblad has feiten
strafblad has objectief Concept_type: property_association General_concept: strafblad has feiten
strafblad has feiten_herhaal Concept_type: property_association General_concept: strafblad has feiten
strafblad has feiten_risico Concept_type: property_association

General\_concept: strafblad has feiten

geen\_strafblad

Definition: strafblad that has feiten less\_than\_or\_equal\_to 0

General\_concept: strafblad

objectief\_criterium\_needed

Definition: strafblad that has feiten greater\_than 0

General\_concept: strafblad

objectief\_criterium\_niet\_voldaan

Definition: objectief\_criterium\_needed that has objectief less\_than\_or\_equal\_to 0

General\_concept: objectief\_criterium\_needed

objectief\_criterium\_voldaan

Definition: objectief\_criterium\_needed that has objectief greater\_than 0

General\_concept: objectief\_criterium\_needed

subjectief\_criterium\_needed

Definition: objectief\_criterium\_needed that has objectief greater\_than 0

General\_concept: objectief\_criterium\_needed

subjectief\_criterium\_niet\_voldaan

Definition: subjectief\_criterium\_needed that has subjectief less\_than\_or\_equal\_to 0

General\_concept: subjectief\_criterium\_needed

subjectief\_criterium\_voldaan

Definition: subjectief\_criterium\_needed that has subjectief greater\_than 0

General\_concept: subjectief\_criterium\_needed

aanvraag

accepted\_aanvraag\_subjectief

Definition: subjectief\_criterium\_needed that has subjectief less\_than\_or\_equal\_to 0

General\_concept: aanvraag

accepted\_aanvraag\_objectief

Definition: objectief\_criterium\_needed that has objectief less\_than\_or\_equal\_to 0

General\_concept: aanvraag

accepted\_aanvraag\_automatisch

Definition: strafblad that has feiten less\_than\_or\_equal\_to 0

General\_concept: aanvraag

rejected\_aanvraag

Definition: subjectief\_criterium\_needed that has subjectief greater\_than 0

General\_concept: aanvraag

## OWL 2

Prefix(=<http://isd.ktu.lt/semantika/s2o#>)

Prefix(ns:=<http://isd.ktu.lt/semantika/>)

Prefix(owl:=<http://www.w3.org/2002/07/owl#>)

Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)

Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)

Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)

Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://isd.ktu.lt/semantika/s2o>

Declaration(Class(<ns:s2o#aanvraag>))

Declaration(Class(<ns:s2o#accepted\_aanvraag\_automatisch>))

Declaration(Class(<ns:s2o#accepted\_aanvraag\_objectief>))

Declaration(Class(<ns:s2o#accepted\_aanvraag\_subjectief>))

Declaration(Class(<ns:s2o#geen\_strafblad>))

Declaration(Class(<ns:s2o#objectief\_criterium\_needed>))

Declaration(Class(<ns:s2o#objectief\_criterium\_niet\_voldaan>))

Declaration(Class(<ns:s2o#objectief\_criterium\_voldaan>))

Declaration(Class(<ns:s2o#person>))

Declaration(Class(<ns:s2o#rejected\_aanvraag>))

Declaration(Class(<ns:s2o#strafblad>))

Declaration(Class(<ns:s2o#subjectief\_criterium\_needed>))

Declaration(Class(<ns:s2o#subjectief\_criterium\_niet\_voldaan>))

Declaration(Class(<ns:s2o#subjectief\_criterium\_voldaan>))

```

Declaration(ObjectProperty(<ns:s2o#contains__strafblad>))
Declaration(ObjectProperty(<ns:s2o#partitive_object_property>))
Declaration(DataProperty(<ns:s2o#feiten>))
Declaration(DataProperty(<ns:s2o#feiten_ernst>))
Declaration(DataProperty(<ns:s2o#feiten_herhaal>))
Declaration(DataProperty(<ns:s2o#feiten_risico>))
Declaration(DataProperty(<ns:s2o#objectief>))
Declaration(DataProperty(<ns:s2o#strafblad>))
Declaration(DataProperty(<ns:s2o#subjectief>))
Declaration(NamedIndividual(<ns:s2o#Name_1>))
Declaration(NamedIndividual(<ns:s2o#Name_2>))
Declaration(AnnotationProperty(:Document_for_SBVR))
Declaration(AnnotationProperty(:SBVR_for_logic))
Declaration(AnnotationProperty(:Triggering_rule))
Declaration(AnnotationProperty(:User_Explanations))
Declaration(AnnotationProperty(<ns:s2o#label_en>))
Declaration(AnnotationProperty(<ns:s2o#label_sbvr>))
#####
# Annotation Properties
#####

# Annotation Property: :Document_for_SBVR (:Document_for_SBVR)

SubAnnotationPropertyOf(:Document_for_SBVR rdfs:isDefinedBy)

# Annotation Property: :SBVR_for_logic (:SBVR_for_logic)

SubAnnotationPropertyOf(:SBVR_for_logic rdfs:isDefinedBy)

# Annotation Property: :Triggering_rule (:Triggering_rule)

SubAnnotationPropertyOf(:Triggering_rule rdfs:isDefinedBy)

# Annotation Property: :User_Explanations (:User_Explanations)

SubAnnotationPropertyOf(:User_Explanations rdfs:isDefinedBy)

#####
# Object Properties
#####

# Object Property: <ns:s2o#contains__strafblad> (strafblad contains strafblad)

AnnotationAssertion(rdfs:label <ns:s2o#contains__strafblad> "strafblad contains strafblad"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#contains__strafblad> "strafblad contains strafblad")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#contains__strafblad> "strafblad contains strafblad"@en)
SubObjectPropertyOf(<ns:s2o#contains__strafblad> <ns:s2o#partitive_object_property>)
TransitiveObjectProperty(<ns:s2o#contains__strafblad>)
ObjectPropertyDomain(<ns:s2o#contains__strafblad> <ns:s2o#strafblad>)
ObjectPropertyRange(<ns:s2o#contains__strafblad> <ns:s2o#strafblad>)

# Object Property: <ns:s2o#partitive_object_property> (partitive object property)

AnnotationAssertion(rdfs:label <ns:s2o#partitive_object_property> "partitive object property"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#partitive_object_property> "partitive object property")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#partitive_object_property> "partitive_object_property"@en)

#####
# Data Properties
#####

# Data Property: <ns:s2o#feiten> (feiten)

AnnotationAssertion(rdfs:label <ns:s2o#feiten> "feiten"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#feiten> "feiten")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#feiten> "feiten"@en)
DataPropertyDomain(<ns:s2o#feiten> <ns:s2o#strafblad>)
DataPropertyRange(<ns:s2o#feiten> xsd:decimal)

```

```

# Data Property: <ns:s2o#feiten_ernst> (feiten ernst)

AnnotationAssertion(rdfs:label <ns:s2o#feiten_ernst> "feiten ernst"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#feiten_ernst> "feiten ernst")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#feiten_ernst> "feiten_ernst"@en)
SubDataPropertyOf(<ns:s2o#feiten_ernst> <ns:s2o#feiten>)
DataPropertyDomain(<ns:s2o#feiten_ernst> <ns:s2o#strafblad>)
DataPropertyRange(<ns:s2o#feiten_ernst> xsd:decimal)

# Data Property: <ns:s2o#feiten_herhaal> (feiten herhaal)

AnnotationAssertion(rdfs:label <ns:s2o#feiten_herhaal> "feiten herhaal"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#feiten_herhaal> "feiten herhaal")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#feiten_herhaal> "feiten_herhaal"@en)
SubDataPropertyOf(<ns:s2o#feiten_herhaal> <ns:s2o#feiten>)
DataPropertyDomain(<ns:s2o#feiten_herhaal> <ns:s2o#strafblad>)
DataPropertyRange(<ns:s2o#feiten_herhaal> xsd:decimal)

# Data Property: <ns:s2o#feiten_risico> (feiten risico)

AnnotationAssertion(rdfs:label <ns:s2o#feiten_risico> "feiten risico"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#feiten_risico> "feiten risico")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#feiten_risico> "feiten_risico"@en)
SubDataPropertyOf(<ns:s2o#feiten_risico> <ns:s2o#feiten>)
DataPropertyDomain(<ns:s2o#feiten_risico> <ns:s2o#strafblad>)
DataPropertyRange(<ns:s2o#feiten_risico> xsd:decimal)

# Data Property: <ns:s2o#objectief> (objectief)

AnnotationAssertion(:Document_for_SBVR <ns:s2o#objectief> "Paragraaf 3.2. Het objectieve criterium

De afgifte van de VOG wordt in beginsel geweigerd indien wordt voldaan aan het objectieve criterium. Het objectieve criterium betreft
de beoordeling of de justitiële gegevens die ten aanzien van de aanvrager zijn aangetroffen, indien herhaald, gelet op het risico voor de
samenleving, een belemmering vormen voor een behoorlijke uitoefening van de functie/taak/bezigheid waarvoor de VOG is
aangevraagd.")
AnnotationAssertion(rdfs:label <ns:s2o#objectief> "objectief"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#objectief> "objectief")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#objectief> "objectief"@en)
SubDataPropertyOf(<ns:s2o#objectief> <ns:s2o#feiten>)
DataPropertyDomain(<ns:s2o#objectief> <ns:s2o#strafblad>)
DataPropertyRange(<ns:s2o#objectief> xsd:decimal)

# Data Property: <ns:s2o#strafblad> (strafblad)

AnnotationAssertion(rdfs:label <ns:s2o#strafblad> "strafblad"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#strafblad> "strafblad")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#strafblad> "strafblad"@en)
DataPropertyDomain(<ns:s2o#strafblad> <ns:s2o#person>)
DataPropertyRange(<ns:s2o#strafblad> xsd:string)

# Data Property: <ns:s2o#subjectief> (subjectief)

AnnotationAssertion(:Document_for_SBVR <ns:s2o#subjectief> "Paragraaf 3.3. Het subjectieve criterium

Op grond van het subjectieve criterium kan worden geoordeeld dat het belang dat een aanvrager heeft bij het verstrekken van de VOG
zwaarder weegt dan het belang van de samenleving bij bescherming tegen het door middel van het objectieve criterium vastgestelde
risico voor de samenleving. In dat geval wordt de VOG afgegeven ondanks dat wordt voldaan aan het objectieve criterium.")
AnnotationAssertion(rdfs:label <ns:s2o#subjectief> "subjectief"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#subjectief> "subjectief")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#subjectief> "subjectief"@en)
SubDataPropertyOf(<ns:s2o#subjectief> <ns:s2o#feiten>)
DataPropertyDomain(<ns:s2o#subjectief> <ns:s2o#strafblad>)
DataPropertyRange(<ns:s2o#subjectief> xsd:decimal)

#####
# Classes
#####

# Class: <ns:s2o#aanvraag> (aanvraag)

```

```

AnnotationAssertion(rdfs:label <ns:s2o#aanvraag> "aanvraag"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#aanvraag> "aanvraag")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#aanvraag> "aanvraag"@en)

# Class: <ns:s2o#accepted_aanvraag_automatisch> (accepted aanvraag automatisch)

AnnotationAssertion(:Document_for_SBVR <ns:s2o#accepted_aanvraag_automatisch> "Ten behoeve van de beoordeling van een
VOG-aanvraag ontvangt het COVOG alle justitiële gegevens betreffende de aanvrager die zijn geregistreerd in het JDS. De justitiële
gegevens kunnen zowel uit Nederland als uit het buitenland afkomstig zijn. Aan de aanvrager die niet voorkomt in het JDS, wordt
zonder meer een VOG afgegeven.

Bron: Beleidsregels VOG 2018

https://wetten.overheid.nl/BWBR0040253/2018-01-01")
AnnotationAssertion(:SBVR_for_logic <ns:s2o#accepted_aanvraag_automatisch> "accepted aanvraag automatisch
  Definition: strafblad that has feiten less_than_or_equal_to 0
  General_concept: aanvraag")
AnnotationAssertion(:Triggering_rule <ns:s2o#accepted_aanvraag_automatisch> <ns:s2o#geen_strafblad>)
AnnotationAssertion(:User_Explanations <ns:s2o#accepted_aanvraag_automatisch> "Een VOG aanvraag wordt automatisch
goedgekeurd wanneer geen feiten in het JDS gevonden zijn.")
AnnotationAssertion(rdfs:label <ns:s2o#accepted_aanvraag_automatisch> "accepted aanvraag automatisch"@en)
EquivalentClasses(<ns:s2o#accepted_aanvraag_automatisch> ObjectIntersectionOf(<ns:s2o#strafblad>
DataSomeValuesFrom(<ns:s2o#feiten> DatatypeRestriction(xsd:decimal xsd:maxInclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#accepted_aanvraag_automatisch> <ns:s2o#aanvraag>)

# Class: <ns:s2o#accepted_aanvraag_objectief> (accepted aanvraag objectief)

AnnotationAssertion(rdfs:label <ns:s2o#accepted_aanvraag_objectief> "accepted aanvraag objectief"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#accepted_aanvraag_objectief> "accepted aanvraag objectief")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#accepted_aanvraag_objectief> "accepted aanvraag objectief"@en)
EquivalentClasses(<ns:s2o#accepted_aanvraag_objectief> ObjectIntersectionOf(<ns:s2o#objectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#objectief> DatatypeRestriction(xsd:decimal xsd:maxInclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#accepted_aanvraag_objectief> <ns:s2o#aanvraag>)

# Class: <ns:s2o#accepted_aanvraag_subjectief> (accepted aanvraag subjectief)

AnnotationAssertion(rdfs:label <ns:s2o#accepted_aanvraag_subjectief> "accepted aanvraag subjectief"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#accepted_aanvraag_subjectief> "accepted aanvraag subjectief")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#accepted_aanvraag_subjectief> "accepted aanvraag subjectief"@en)
EquivalentClasses(<ns:s2o#accepted_aanvraag_subjectief> ObjectIntersectionOf(<ns:s2o#subjectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#subjectief> DatatypeRestriction(xsd:decimal xsd:maxInclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#accepted_aanvraag_subjectief> <ns:s2o#aanvraag>)

# Class: <ns:s2o#geen_strafblad> (geen strafblad)

AnnotationAssertion(rdfs:label <ns:s2o#geen_strafblad> "geen strafblad"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#geen_strafblad> "geen strafblad")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#geen_strafblad> "geen strafblad"@en)
EquivalentClasses(<ns:s2o#geen_strafblad> ObjectIntersectionOf(<ns:s2o#strafblad> DataSomeValuesFrom(<ns:s2o#feiten>
DatatypeRestriction(xsd:decimal xsd:maxInclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#geen_strafblad> <ns:s2o#strafblad>)

# Class: <ns:s2o#objectief_criterium_needed> (objectief criterium needed)

AnnotationAssertion(rdfs:label <ns:s2o#objectief_criterium_needed> "objectief criterium needed"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#objectief_criterium_needed> "objectief criterium needed")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#objectief_criterium_needed> "objectief_criterium_needed"@en)
EquivalentClasses(<ns:s2o#objectief_criterium_needed> ObjectIntersectionOf(<ns:s2o#strafblad>
DataSomeValuesFrom(<ns:s2o#feiten> DatatypeRestriction(xsd:decimal xsd:minExclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#objectief_criterium_needed> <ns:s2o#strafblad>)

# Class: <ns:s2o#objectief_criterium_niet_voldaan> (objectief criterium niet voldaan)

AnnotationAssertion(rdfs:label <ns:s2o#objectief_criterium_niet_voldaan> "objectief criterium niet voldaan"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#objectief_criterium_niet_voldaan> "objectief criterium niet voldaan")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#objectief_criterium_niet_voldaan> "objectief_criterium_niet_voldaan"@en)
EquivalentClasses(<ns:s2o#objectief_criterium_niet_voldaan> ObjectIntersectionOf(<ns:s2o#objectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#objectief> DatatypeRestriction(xsd:decimal xsd:maxInclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#objectief_criterium_niet_voldaan> <ns:s2o#objectief_criterium_needed>)

```

```

# Class: <ns:s2o#objectief_criterium_voldaan> (objectief criterium voldaan)

AnnotationAssertion(rdfs:label <ns:s2o#objectief_criterium_voldaan> "objectief criterium voldaan"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#objectief_criterium_voldaan> "objectief criterium voldaan")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#objectief_criterium_voldaan> "objectief_criterium_voldaan"@en)
EquivalentClasses(<ns:s2o#objectief_criterium_voldaan> ObjectIntersectionOf(<ns:s2o#objectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#objectief> DatatypeRestriction(xsd:decimal xsd:minExclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#objectief_criterium_voldaan> <ns:s2o#objectief_criterium_needed>)

# Class: <ns:s2o#person> (person)

AnnotationAssertion(rdfs:label <ns:s2o#person> "person"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#person> "person")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#person> "person"@en)

# Class: <ns:s2o#rejected_aanvraag> (rejected aanvraag)

AnnotationAssertion(rdfs:label <ns:s2o#rejected_aanvraag> "rejected aanvraag"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#rejected_aanvraag> "rejected aanvraag")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#rejected_aanvraag> "rejected_aanvraag"@en)
EquivalentClasses(<ns:s2o#rejected_aanvraag> ObjectIntersectionOf(<ns:s2o#subjectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#subjectief> DatatypeRestriction(xsd:decimal xsd:minExclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#rejected_aanvraag> <ns:s2o#aanvraag>)

# Class: <ns:s2o#subjectief_criterium_needed> (subjectief criterium needed)

AnnotationAssertion(rdfs:label <ns:s2o#subjectief_criterium_needed> "subjectief criterium needed"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#subjectief_criterium_needed> "subjectief criterium needed")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#subjectief_criterium_needed> "subjectief_criterium_needed"@en)
EquivalentClasses(<ns:s2o#subjectief_criterium_needed> ObjectIntersectionOf(<ns:s2o#objectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#objectief> DatatypeRestriction(xsd:decimal xsd:minExclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#subjectief_criterium_needed> <ns:s2o#objectief_criterium_needed>)

# Class: <ns:s2o#subjectief_criterium_niet_voldaan> (subjectief criterium niet voldaan)

AnnotationAssertion(rdfs:label <ns:s2o#subjectief_criterium_niet_voldaan> "subjectief criterium niet voldaan"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#subjectief_criterium_niet_voldaan> "subjectief criterium niet voldaan")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#subjectief_criterium_niet_voldaan> "subjectief_criterium_niet_voldaan"@en)
EquivalentClasses(<ns:s2o#subjectief_criterium_niet_voldaan> ObjectIntersectionOf(<ns:s2o#subjectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#subjectief> DatatypeRestriction(xsd:decimal xsd:maxInclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#subjectief_criterium_niet_voldaan> <ns:s2o#subjectief_criterium_needed>)

# Class: <ns:s2o#subjectief_criterium_voldaan> (subjectief criterium voldaan)

AnnotationAssertion(rdfs:label <ns:s2o#subjectief_criterium_voldaan> "subjectief criterium voldaan"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#subjectief_criterium_voldaan> "subjectief criterium voldaan")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#subjectief_criterium_voldaan> "subjectief_criterium_voldaan"@en)
EquivalentClasses(<ns:s2o#subjectief_criterium_voldaan> ObjectIntersectionOf(<ns:s2o#subjectief_criterium_needed>
DataSomeValuesFrom(<ns:s2o#subjectief> DatatypeRestriction(xsd:decimal xsd:minExclusive "0"^^xsd:integer))))
SubClassOf(<ns:s2o#subjectief_criterium_voldaan> <ns:s2o#subjectief_criterium_needed>)

#####
# Named Individuals
#####

# Individual: <ns:s2o#Name_1> (Name 1)

AnnotationAssertion(rdfs:label <ns:s2o#Name_1> "Name 1"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#Name_1> "Name 1")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#Name_1> "Name_1"@en)
ClassAssertion(<ns:s2o#person> <ns:s2o#Name_1>)
DataPropertyAssertion(<ns:s2o#feiten> <ns:s2o#Name_1> "1"^^xsd:integer)
DataPropertyAssertion(<ns:s2o#feiten_ernst> <ns:s2o#Name_1> "1"^^xsd:integer)
DataPropertyAssertion(<ns:s2o#feiten_herhaal> <ns:s2o#Name_1> "1"^^xsd:integer)
DataPropertyAssertion(<ns:s2o#feiten_risico> <ns:s2o#Name_1> "1"^^xsd:integer)
DataPropertyAssertion(<ns:s2o#objectief> <ns:s2o#Name_1> "1"^^xsd:integer)
DataPropertyAssertion(<ns:s2o#subjectief> <ns:s2o#Name_1> "1"^^xsd:integer)

# Individual: <ns:s2o#Name_2> (Name 2)

```



```
AnnotationAssertion(rdfs:label <ns:s2o#Name_2> "Name 2"@en)
AnnotationAssertion(<ns:s2o#label_en> <ns:s2o#Name_2> "Name 2")
AnnotationAssertion(<ns:s2o#label_sbvr> <ns:s2o#Name_2> "Name 2"@en)
ClassAssertion(<ns:s2o#person> <ns:s2o#Name_2>)
```

```
)
```