



Productinformatie

Practis B.V.
Schapedrift 91
3371 JJ Hardinxveld-Giessendam
Tel: +31 (0)184 - 620 111
Fax: +31 (0)184 - 614 222

Practis Belgium N.V.
Hoogkamerstraat 42
B-9100 Sint-Niklaas
Tel: +32 (0)3 7807330
Fax: +32 (0)3 7776192

e-mail: info@practis.com
web: <http://www.practis.com>

INHOUDSOPGAVE

1. Inleiding	3
2. Kenmerken RDC	4
2.1. Basis elementen	4
2.2. Meertalige applicatie	5
2.3. RDC als IBSEN Component	5
2.3.1. IBSEN Concept	5
2.3.2. IBSEN componenten architectuur	6
3. Functionele aspecten	10
3.1. Beheer van formules	10
3.1.1. Berekeningselementen en sleutelattributen	10
3.1.2. Berekeningsdoelen en formules	11
3.1.3. Berekeningsdoelen en methoden	12
3.1.4. Mapping van berekeningselementen	12
3.2. Samenstellen van formules	14
3.3. Berekenen van formules	14
3.4. Opties en instelling configuratie	15
4. Beveiligingsaspecten	17
4.1. Autorisatie	17
4.2. Levenscyclus	17
5. Technische aspecten	19
5.1. Overdraagbaarheid	19
5.1.1. De dimensies van RDC uitwisselbaarheid	19
5.1.2. RDC uitwisselbaar: met en zonder hercompilatie.	20

1. Inleiding

Bij talloze processen, administratief of niet, binnen (pensioen)verzekeringmaatschappijen dienen berekeningen te worden gemaakt. Veelal worden deze berekeningen op diverse plaatsen in diverse informatiesystemen al of niet op dezelfde logische of technische wijze uitgevoerd. En juist deze berekeningen zijn onderworpen aan veranderingen als gevolg van wetswijzigingen, beleids- en/of productaanpassingen, etc.

Dit resulteert in een grote hoeveelheid onderhoudswerkzaamheden op de vele in gebruik zijnde informatiesystemen.

De Rules Design Component (RDC) biedt een effectieve oplossing voor het vastleggen en beheren van rekenregels die toegepast worden in informatiesystemen van levensverzekeringmaatschappijen en pensioenfondsen. De rekenregels benodigd voor de bepaling van bijvoorbeeld pensioenen, voorzieningen, winstdeling en provisie kunnen in RDC worden beheerd en berekend.

RDC kan als component geïntegreerd worden met bestaande of nieuwe informatiesystemen. Om de formules toegankelijk te maken voor informatiesystemen, zoals de offerteadministratie en de polisadministratie of beslissingsondersteunende softwarecomponenten, is een interface ontwikkeld die de communicatie verzorgt. Deze interface zorgt ervoor dat de gegevens die opgeslagen zijn in bijvoorbeeld de polisadministratie vertaald worden naar de door RDC benodigde grootheden. RDC berekent vervolgens het resultaat en koppelt dit weer terug naar het aanroepend informatiesysteem. Kortom, RDC is het hart van alle berekeningen in het verwerkingsysteem. Hierdoor wordt een zeer hoge mate van consistentie bereikt omdat alle verwerkingsystemen gebruik maken van dezelfde berekeningsmodule.

Voordelen RDC

- sluit aan op de denk- en werkwijze van de business expert
- beheert rekenregels van velerlei aard ten behoeve van meerdere toepassingen op één plaats
- vermindert softwareonderhoud; de diverse rekenregels hoeven niet langer 'hard' geprogrammeerd te worden
- vergroot flexibiliteit, verkort Time-to-Market
- biedt in combinatie met de softwarecomponent Actuarial Knowledge System (AKS) van Practis, een totaaloplossing voor alle berekeningen van levens- en pensioenverzekeraars.

2. Kenmerken RDC

Verzekeringstechnische gegevens zoals aanspraken, financiering en duren zijn in RDC getypeerd onder berekeningsdoelen. De hoogte van een aanspraak is bijvoorbeeld een doel. Om tot de bepaling van de hoogte van een aanspraak te komen moeten er vaak allerlei gegevens berekend worden, zoals de leeftijden van de verzekerde op verschillende momenten, de duur van de verzekering of premiebetaling en tarieven. Al deze gegevens zijn in RDC berekeningsdoelen. Vaak worden dezelfde regels voor verschillende doelen gebruikt. Zo kan bijvoorbeeld leeftijdbeplating voor zowel de financieringsvorm als een aanspraak gebruikt worden. Voor al deze rekenregels kan de gebruiker RDC-formules samenstellen.

Om de resultaten van een berekeningsdoel te bepalen moeten niet altijd hetzelfde formularia gebruikt worden. Afhankelijk van bijvoorbeeld de fase van de administratieve verwerking van een mutatie is de methode van berekening van de gegevens anders en moeten er andere formules gebruikt worden. Volgens welke aanpak RDC een gegeven berekent, dus welke formule gebruikt wordt, wordt bepaald aan de hand van een formuleselectie. De selectie van de formule waarmee een gegeven berekend moet worden kan door de gebruiker zelf ingericht worden. De gebruiker kan aangeven op basis van welke gegevens, zoals mutatiefase, er welke formules gebruikt moeten worden. Een gegeven waarvan de formuleselectie afhankelijk is wordt in RDC een sleutelattribuut of 'key item' genoemd en is een invoergegeven voor RDC. De waarden van de sleutelattributen tijdens de berekening bepalen welke formules uitgevoerd worden.

2.1. *Basis elementen*

RDC wordt gekenmerkt doordat de diverse berekeningen worden uitgevoerd met gebruikmaking van een vijftal basiselementen, te weten:

- Berekeningsdoelen (objectives)
- Berekeningsmethoden
- Formules
- Berekeningselementen
- Sleutelattributen

Een berekeningsdoel bestaat uit berekeningsmethoden en formules. In een berekeningsmethode wordt bepaald onder welke omstandigheden een bepaalde formule uitgevoerd moet worden. Een berekeningsmethode is een beslissingsboom, bestaande uit verwijzingen naar formules. Als formules worden uitgevoerd onder bepaalde voorwaarden, dan wordt gebruik gemaakt van een sleutelattribuut. Formules die direct onder een

berekeningsmethode zijn geplaatst, worden altijd uitgevoerd. Formules die afhankelijk van een sleutelattribuut zijn geplaatst, worden enkel in bepaalde gevallen uitgevoerd. Sleutelattributen zijn criteria die de omstandigheden bepalen waaronder een formule wordt uitgevoerd.

Een berekeningselement is een basis item voor het componeren van formules. Er zijn twee soorten:

- Berekeningselementen van het type 'invoer', 'uitvoer' of 'invoer/ uitvoer'. Deze worden gebruikt bij het componeren van formules
 - Berekeningselementen van het type 'invoer' worden alleen gebruikt als invoer van een formule
 - Berekeningselementen van het type 'uitvoer' worden alleen gebruikt als uitvoer in een formule; in een formule kan maar één uitvoer item berekend worden
 - Berekeningselementen van het type 'invoer/ uitvoer' kunnen zowel als invoer als uitvoer gebruikt worden
- Berekeningselementen van het type 'Classificerend'. Deze worden gebruikt als sleutelattributen bij het inrichten van een berekeningsmethode.

2.2. *Meertalige applicatie*

RDC is een meertalige applicatie en beschikbaar in 4 talen (Nederlands, Engels, Frans, Spaans). Omdat RDC meertalig is opgezet is het eenvoudig om het systeem uit te breiden met een andere taal.

Per gebruiker kan de gewenste taal worden opgegeven, zodat bijvoorbeeld een Engelstalige medewerker RDC kan starten in het Engels, terwijl een Nederlandstalige gebruiker gewoon de Nederlandse versie van RDC te zien krijgt.

2.3. *RDC als IBSEN Component*

RDC is als component ontwikkeld op basis van het IBSEN (Insurance Business Engines) concept en maakt deel uit van de IBSEN architectuur van componenten.

De verschillende IBSEN componenten zijn in grote mate onafhankelijk van elkaar. Iedere component ondersteunt een bepaalde taak van de verzekeraar, van productontwerp tot polisbeheer. Hierdoor kunnen ze op een eenvoudige manier los van elkaar geïmplementeerd worden en is integratie met externe componenten mogelijk.

2.3.1. *IBSEN Concept*

De doelstelling van het IBSEN concept is dat materiedeskundigen hun kennis op een eenvoudige en doeltreffende manier kunnen inbrengen in het systeem. Dit is gerealiseerd door toepassing van de volgende basisconcepten.

Single Point of Definition (SPOD)

Het SPOD-concept beoogt een eenmalige en centrale definiëring van regels en data, die benaderbaar zijn voor c.q. uitvoerbaar zijn door alle componenten.

De volgende gegevens en regels worden gecentraliseerd:

- omgevingspecificaties (bijvoorbeeld fiscale regels);
- bedrijfsspecificaties (bijvoorbeeld commissiesystemen en winstverdeling);
- business regels (bijvoorbeeld actuariële formules en berekeningen).

Toepassing van dit concept bevordert de eenduidigheid en onderhoudbaarheid van het systeem.

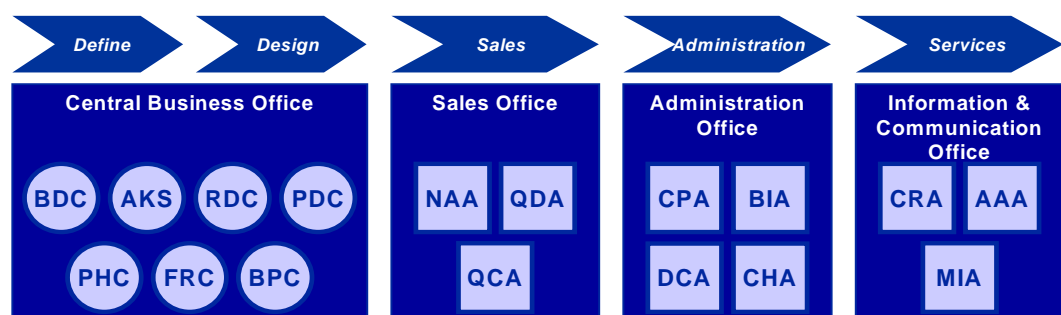
Power to the expert

Het concept 'Power to the expert' stelt materiedeskundigen in staat rechtstreekse invloed uit te oefenen op operationele informatiesystemen.

Om dit te verwezenlijken worden tools ter beschikking gesteld waarmee materiedeskundigen snel en efficiënt data en regels kunnen vastleggen in het systeem, in een voor het betreffende expertisegebied (actuarieel, verzekeringstechniek, marketing en administratie) geëigende vorm. De communicatie tussen materiedeskundigen en IT-experts wordt hierdoor vereenvoudigd en vindt plaats op ieders expertisegebied, zodat communicatieproblemen kunnen worden voorkomen.

2.3.2. IBSEN componenten architectuur

De bedrijfsvoering van verzekeraars bestaat voor een groot deel uit informatieprocessen. Hierin kunnen vijf stadia onderscheiden worden: het beschrijven van de omgeving het werkterrein (define), het ontwerpen van producten en processen (design), het aanbieden van producten en diensten op de onderscheiden markten (sales), het uitvoeren van de administratie (administration) en de dienstverlening (services). Ter ondersteuning van deze procesketen is de volgende applicatie- en componentenarchitectuur gedefinieerd.



Hieronder worden de onderscheiden componenten per 'Office' kort toegelicht.

Central Business Office

- **BDC** (Business Define Component)
Ondersteunt het definiëren van businessregels en businessdata die relevant zijn voor de omgeving waarin de verzekeringsmaatschappij opereert.
- **FRC** (Fiscal Rules Component)
Ondersteunt de fiscale consequenties bij premie- of verplichtingvaststelling.
- **PDC** (Product Design Component)
Ondersteunt - in combinatie met AKS en RDC - het creëren, het parametriseren en beheren van specificaties die een verzekeringsproduct beschrijven.
- **PHC** (Policy Handling Component)
Bevat de kennis en de logica voor de verwerking van polismutaties en voor het uitvoeren van proforma- en offerteberekeningen.
- **AKS** (Actuarial Knowledge System)
Ondersteunt de definitie en berekening van alle actuariële formules.
- **RDC** (Rules Design Component)
Ondersteunt de definitie en berekening van alle overige, niet zuiver actuariële, berekeningen.
- **BPC** (Benefit Projection Component)
Berekent vanuit de premie de 'benefits' en vanuit 'benefits' de premie.

Sales Office

- **NAA** (Need Analysis Application)
Ondersteunt de behoeftebepaling van een klant.
- **QDA** (Quotation Delivery Application)
Ondersteunt het opstellen, uitbrengen en opvolgen van offertes voor zowel nieuwe als bestaande contracten.
- **QCA** (Quotation Calculation Application)
Ondersteunt het berekenen en beheren van offertes.

Administration Office

- **CPA** (Contract Policy Application)
Ondersteunt het opstellen en beheren van verzekeringscontracten, plannen en polissen
- **BIA** (Billing Application)
Ondersteunt - binnen de incassoprocedure - het aanmaken van verzekeringstechnische vorderingen (vervaldagberichten en uitnodigingen tot betalingen) en boekingen.
- **CHA** (Claims Handling Application)
Ondersteunt - binnen de excassoadministratie - de bepaling en de registratie van de uit te keren aanspraken per begunstigde, alsmede de prolongatie van uitkeringen ten behoeve van de betaalbaarstelling en de hieruit volgende boekingen.
- **DCA** (Distribution Channel Application)

Ondersteunt de definitie en het beheer van de distributieorganisatie (distributiekkanalen, commissiesystemen), het beheer van distributieovereenkomsten en de administratie van commissies.

Information & Communication Office

- **CRA** (Central Relation Application)
Ondersteunt het beheer van de gegevens van alle relaties van de organisatie.
- **AAA** (Actuarial Analysis Application)
Ondersteunt de actuariële verslaglegging, analyse en verantwoording, alsmede de afhandeling ten behoeve van co- en herverzekering.
- **MIA** (Management Information Application)
Ondersteunt het leveren van de nodige informatie waarop het management zich kan baseren om beslissingen te nemen; betreft zowel inhoudelijke informatie (productiecijfers, etc.) als procesmatige/besturingsinformatie (wacht- en levertijden, knelpunten).

De werking van de IBSEN-componenten wordt bepaald door programmatuur en inrichting. De inrichting wordt vormgegeven door middel van (kennis)tabellen en business regels die gedefinieerd c.q. gecomponeerd worden door middel van de componenten in de Central Business Office (bijvoorbeeld: AKS en RDC). In combinatie zorgen programmatuur en inrichting voor een, in principe, oneindige functionaliteit.

Het karakter van de componenten is verschillend. De verschillen zijn met name gelegen in:

- **Scope:** in hoeverre is een component generiek voor de gehele business, of specifiek voor een land, maatschappij, etc.

Voorbeeld RDC (Rules Design Component) wordt beschouwd als generiek voor de verzekeringsbusiness. Binnen de IBSEN-architectuur is RDC dan ook een standaardcomponent. CPA (Contract Policy Administration) daarentegen is geen standaardcomponent. Voor dergelijke procesafhankelijke Administration Office componenten geldt een framework benadering. Practis ontwerpt en ontwikkelt de kernsoftware. Deze vormt de basis voor een implementatie voor een specifieke klant. Inclusief het benodigde maatwerk ontstaat dan op efficiënte wijze een effectieve oplossing voor een klant.

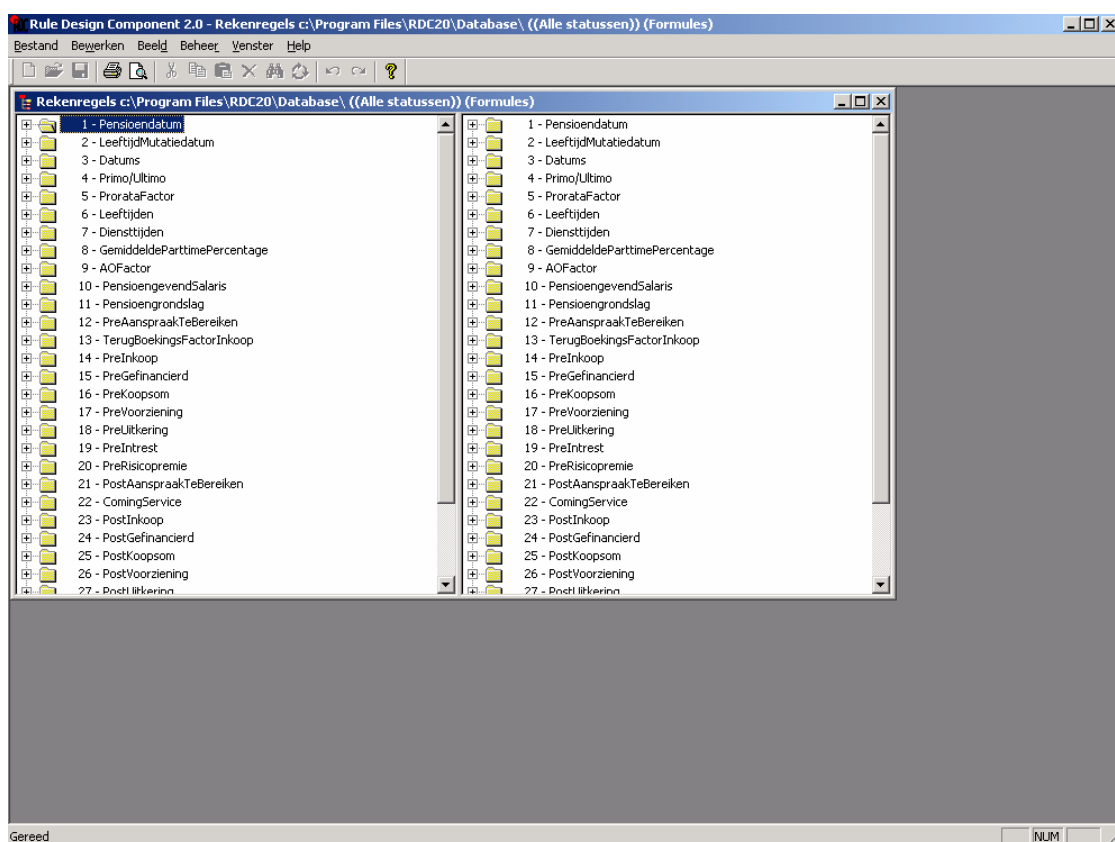
- **Gebruiksaspecten:** wie zijn de gebruikers van de component (businessdeskundige, contractbeheerder, commercieel medewerker, tussenpersoon, klant, ...)
- **Functionele aspecten:** aan welk bedrijfsproces dient de component ondersteuning te bieden
- **Technische aspecten:** mate van noodzakelijke overdraagbaarheid en (gedeeltelijk) hergebruik.

In samenhang ontstaat een totaaloplossing voor een personenverzekeraar, waarbij het gehele scala aan verzekeringsproducten (op het gebied van 'personen') voor 'front-' en 'back office' processen wordt ondersteund. Iedere verzekeringsmaatschappij heeft de mogelijkheid om een optimale keuze te maken, voor wat betreft breedte en diepgang van de IBSEN-oplossing: de uitgangssituatie en doelstelling is voor iedere verzekeringsmaatschappij verschillend. Tevens biedt de IBSEN-architectuur inzicht in de mogelijkheden om IBSEN-componenten te vervangen door externe componenten met dezelfde functionaliteit.

3. Functionele aspecten

3.1. Beheer van formules

Formules worden in RDC opgeslagen in een formuledatabase. De formuledatabase is gebaseerd op de verkennerstructuur zoals deze in Windows Verkenner wordt gebruikt. De ordening van de formules vindt plaats door ze in mappen te plaatsen. De structuur van de mappen kan de gebruiker zelf samenstellen. De formuledatabase kan er dan als volgt uitzien:



3.1.1. Berekeningselementen en sleutelattributen

Zowel de berekeningselementen als de sleutelattributen kunnen door de business-expert zelf worden gedefinieerd en opgeslagen in de database van RDC.

Rule Design Component 2.0 - [Berekeningselementen]

Bestand Bewerken Beeld Beheer Venster Help

Identificatie *	Naam	Datatype	Type	Berekeningsdoel
1	Provision_zillmer	Floating_point	Invoer/Uitvoer	Provision
2	Zillmerquote	Floating_point	Invoer/Uitvoer	Provision
3	Gross_premium	Floating_point	Invoer/Uitvoer	Premium/contribution
4	Premium_unity_net	Floating_point	Invoer	Premium/contribution
7	Age_method	Integer	Classifierend	Ages
8	Age	Floating_point	Invoer/Uitvoer	Ages
9	Date_of_birth	Date	Invoer	Ages
10	Transactiondate	Date	Invoer/Uitvoer	Ages
11	Premium_unity_gross	Floating_point	Invoer	Premium/contribution
12	Kind_of_contribution	Integer	Classifierend	Premium/contribution
13	Salary	Floating_point	Invoer	Premium/contribution
14	Contribution_percentage	Floating_point	Invoer	Premium/contribution
15	Product	Integer	Classifierend	Provision
16	Transaction	Integer	Classifierend	Provision
18	Retirement_date	Date	Invoer	Ages
21	Ax1n	Floating_point	Invoer	Provision
39	Provision	Floating_point	Invoer/Uitvoer	Provision
40	Single_premium	Floating_point	Invoer	Provision
41	Net_Premium	Floating_point	Invoer/Uitvoer	Premium/contribution
42	Annuity	Floating_point	Invoer	Provision
43	Capital	Floating_point	Invoer/Uitvoer	Provision
120	Sxn	Floating_point	Invoer	Provision

Gereed

De gebruiker kan derhalve aan de berekeningselementen en sleutelattributen een terminologie toekennen die hij gewend is.

Aan berekeningselementen en sleutelattributen kunnen verschillende typeringen worden meegegeven b.v.

- Datum
- Integer
- Floating point
- Boolean.

Verder kan bij de berekeningselementen worden aangegeven of het een operand in een formule is (invoer) dan wel een resultaat behelst (uitvoer).

3.1.2. *Berekeningsdoelen en formules*

Door middel van een 'tree view' kunnen nu berekeningsdoelen worden gedefinieerd. Het berekeningsdoel geeft aan voor welk doel het aanroepende automatiseringssysteem een berekening door RDC wenst te laten uitvoeren, b.v. bepaling van de voorziening verzekeringsverplichting.

Aan de berekeningsdoelen kunnen formules worden toegekend. De onderscheiden formules kunnen op de voor het gedefinieerde doel op logische wijze worden gegroepeerd, b.v. alle formules die gebruikt worden bij de prospectieve methode.

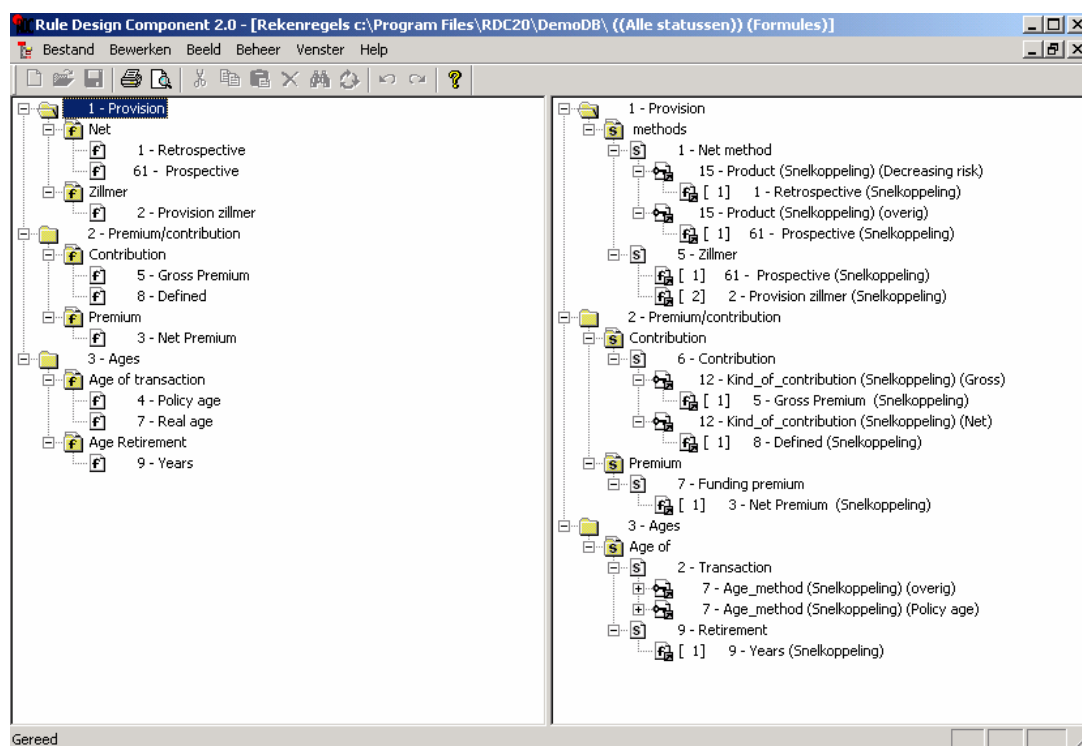
Door de 'tree view' methode kunnen ook specifieke groepen van formules worden toegekend aan b.v. de netto, Höckner of Zillmer-methode. Bedoelde groepen kunnen weer worden onderverdeeld in andere logische groepen, enz., enz.

3.1.3. Berekeningsdoelen en methoden

Teneinde het aanroepende automatiseringssysteem in staat te stellen van de geformuleerde berekeningsdoelen gebruik te maken worden de berekeningsdoelen opgedeeld in logisch bij elkaar behorende groepen van methoden.

Binnen de aldus ontstane methode kan worden aangegeven welke eerder gedefinieerde en gecomponeerde formules in achtereenvolgende wijze moeten worden uitgevoerd.

Door middel van het toekennen van een waarde aan een sleutelattribuut kan worden aangegeven welke formule al of niet dient te worden berekend.



Zo zal binnen de methodegroep 'prospectief' gekozen worden voor de netto-methode, afhankelijk van de waarde van het sleutelattribuut 'premiebetalende' wordt gekozen voor de formule 'CW-lasten' minus 'CW-baten' dan wel de formule 'CW-lasten'

3.1.4. Mapping van berekeningselementen

Om RDC te kunnen gebruiken binnen een informatiesysteem, dient er overeenstemming te zijn tussen client-IS en RDC voor wat betreft de definitie van een set van berekeningselementen. Dit wil zeggen dat de data-elementjes die binnen RDC en binnen client-IS gekoppeld zijn aan een gegeven Berekeningselement-ID van hetzelfde datatype zijn en een gelijke interpretatie hebben, en dat elk berekeningselement dat binnen een client-IS in gebruik is ook binnen RDC bestaat. Deze situatie moet gehandhaafd blijven,

terwijl men in de loop der tijd zowel in RDC als in het IS berekeningselementen toevoegd of verwijderd.

Binnen RDC doorlopen berekeningselementen een levenscyclus. Deze bestaat uit de fases “Aangemaakt”, “In Productie”, “Geblokkeerd” en “Uit Productie”. Deze fases kunnen alleen in deze volgorde doorlopen worden; er is geen weg terug. Alleen berekeningselementen met status “Aangemaakt” en “Uit Productie” kunnen worden verwijderd. Overgang naar status “Uit Productie” vereist dat het berekeningselement niet meer in gebruik is in client-IS'en. (RDC is uiteraard niet in staat om deze controle uit te voeren; deze controle berust derhalve bij een daartoe bevoegde gebruiker, bijvoorbeeld de applicatiebeheerder.)

Een IS kan berekeningselement-definitie uit RDC ‘importeren’, d.w.z. dat een berekeningselement-ID wordt gekoppeld aan een data-element (zoals een tabelkolom of een tijdelijke waarde) met datatype en interpretatie die overeenkomen met de definitie binnen RDC. Voorwaarde hierbij is dat het betreffende berekeningselement (volgens RDC) in de fase “In Productie” is.

Bewaking van deze laatste voorwaarde kan op twee manieren plaatsvinden:

- ‘Handmatig’. Programmeurs van een client-IS kunnen binnen RDC een lijst van beschikbare berekeningselementen (met status “In Productie”) opvragen, en de ID's van de gewenste elementen op de een of andere manier koppelen aan een data-element (hetzij door een dergelijke koppeling ‘hard’ in de programmatuur in te bouwen, hetzij door ‘flexibel’ een mapping-tabel in te richten.)
- ‘Automatisch’. RDC kent API functies voor het opvragen van definitie van beschikbare berekeningselementen. Op basis hiervan kan men binnen een client-IS een beheertool bouwen voor het inrichten van een mapping-tabel.¹

Doordat uitsluitend berekeningselementen met status “In Productie” worden geïmporteerd in het client-IS, en door de restricties op de levenscyclus van berekeningselementen wordt gegarandeerd dat RDC en client-IS'en overeen blijven stemmen m.b.t. de definitie van berekeningselementen.

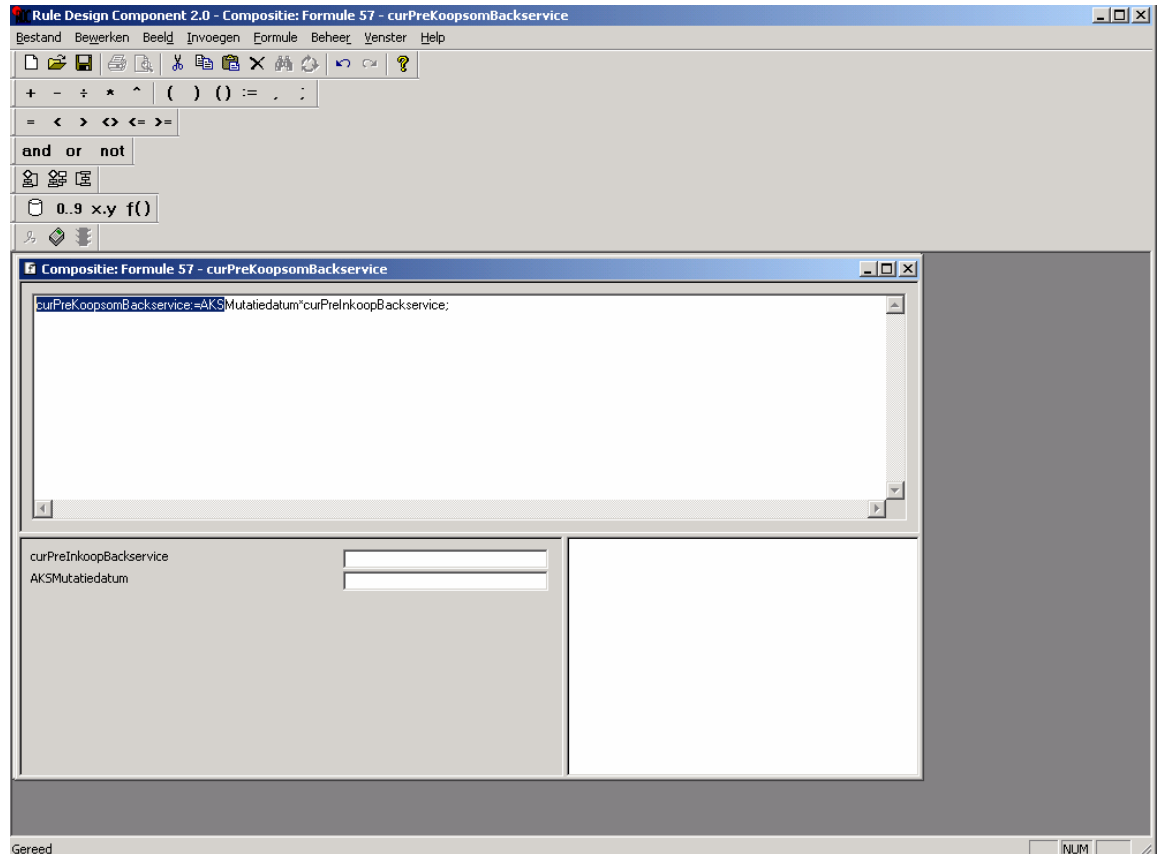
Er dient ook een mapping te worden gemaakt tussen de berekeningsdoelen en –methodes en client-IS.

¹ Vanwege de verscheidenheid van mogelijke client-IS'en, en het feit dat de structuur van zo'n mapping-tabel niet voor elke toepassing gelijk hoeft te zijn, biedt RDC geen kant-en-klare oplossing voor een dergelijke beheertool.

3.2. *Samenstellen van formules*

Het samenstellen van formules gebeurt met behulp van een z.g. werkblad. De formule wordt samengesteld uit operanden, de gedefinieerde berekeningselementen, operatoren en logica, die op een werkbalk zijn weergegeven.

De businessdeskundige kan op die manier met bekende terminologie formules samenstellen.



Tijdens het samenstellen kan, naast de automatische syntax controle, validatie van de formule plaatsvinden omdat de operanden onder het werkblad worden weergegeven en nadat hieraan waarden worden toegekend wordt de berekening van de formule uitgevoerd, zodat de rekenkundige uitkomst van de formule op eenvoudige wijze kan worden gecontroleerd.

Indien validatie heeft plaatsgevonden kan de formule worden opgeslagen in de formuleredatabase.

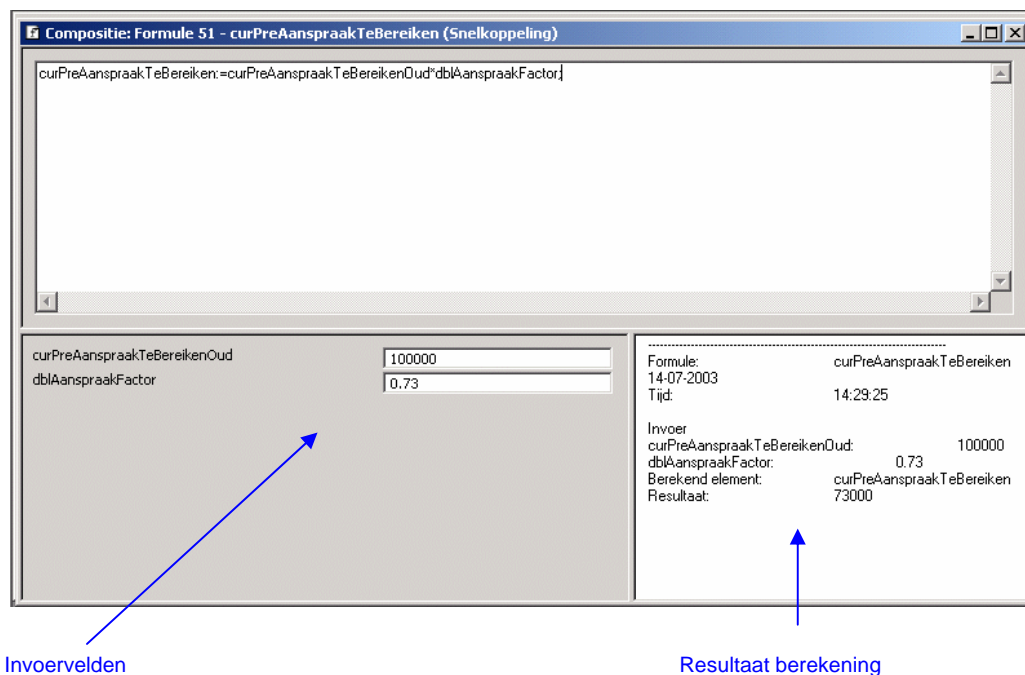
3.3. *Berekenen van formules*

Gedurende een bewerking in een automatiseringssysteem moet een berekening worden uitgevoerd. Het aanroepende systeem zal dan RDC aanroepen voor de berekening van een bepaald berekeningsdoel, hetwelk onderdeel is van het uit te voeren algoritme. De

berekening zal moeten geschieden volgens een bepaalde methode die gedefinieerd staat in de bij het aanroepende systeem gedefinieerde gegevens, b.v. contractgegevens. Als doel en methode voor RDC bekend zijn kan RDC bepalen welke *sleutelgegevens* relevant zijn en zal deze bij het aanroepende systeem opvragen. Nadat deze *sleutelgegevens* bekend zijn gemaakt kan RDC bepalen welke formule(s) moet(en) worden uitgevoerd. RDC zal hiertoe aan het aanroepende systeem de benodigde rekenelementen (operanden) vragen. Nadat de beoogde berekening is uitgevoerd stelt RDC het resultaat beschikbaar aan het aanroepende systeem.

De verschillende berekeningen worden dus niet uitgevoerd door het bepalen van parameters, maar RDC vraagt de benodigde gegevens (GET) en stelt vervolgens resultaten ter beschikking (SET). Een en ander heeft een hoge mate van flexibiliteit in benodigde berekeningen tot gevolg, aangezien de businessdeskundige zelf invloed uitoefent op de hoedanigheid van de berekeningen.

Via de Graphical User Interface van RDC kunnen ook berekeningen worden uitgevoerd. De benodigde invoergegevens moeten worden gevuld. De in te vullen velden zijn afhankelijk van de formule. Het resultaat van de berekening verschijnt direct op het scherm.



3.4. *Opties en instelling configuratie*

RDC biedt een aantal opties die door de gebruiker ingesteld kunnen worden. Als voorbeelden kunnen genoemd worden:

- Nummer opties voor het weergeven en genereren van formules, berekeningselementen, berekeningsmethoden en berekeningsdoelen;
- Locatie van de formuledatabase;

- Wijze van afronding

4. Beveiligingsaspecten

De beveiliging van RDC gebeurt door middel van autorisatie die verleend wordt aan gebruikers. De beveiliging van formules vindt plaats door een levenscyclus toe te passen op formules.

4.1. *Autorisatie*

Om RDC te gebruiken is een identificatiecode en een wachtwoord nodig. De combinatie van de identificatiecode en het wachtwoord bepalen de rechten die aan de gebruiker worden toegekend. De applicatiebeheerder van RDC kan gebruikersgroepen definiëren. Voor elke gebruikersgroep wordt bepaald waartoe de betreffende gebruikers geautoriseerd zijn. In RDC kan dit per functie worden bepaald. Op basis van de autorisatietabellen en aan de hand van de ingegeven gebruikersnaam en wachtwoord worden de gebruikersrechten bepaald en de taal waarin RDC zich zal presenteren.

4.2. *Levenscyclus*

Een levenscyclus is een reeks fasen die een in RDC gedefinieerd gegeven, zoals een formule of grondslag, doorloopt.

De levenscyclus stelt de gebruiker in staat om formules, grondslagen en opslagen eerst te testen voordat ze daadwerkelijk worden gebruikt. Afhankelijk van de fase waarin een formule, grondslag of opslag zich bevindt kan deze worden gewijzigd of verwijderd. De volgende fasen zijn gedefinieerd in RDC:

- Aangemaakt
De formule, grondslag of opslag is gecomponeerd, maar nog niet voor test gereed
- In test
De formule, grondslag of opslag is gereed om getest te worden
- Gefiatteerd
De formule, grondslag of opslag is gefiatteerd door een expert en kan nu overgeheveld worden naar de productieomgeving
- In productie
De formule, grondslag of opslag kan worden gebruikt door andere gebruikers of andere systemen
- Geblokkeerd

De formule, grondslag of opslag kan nog worden gebruikt, maar niet meer in nieuwe composities

- Verwijderd

De formule, grondslag of opslag kan/ mag niet meer worden gebruikt.

5. Technische aspecten

5.1. Overdraagbaarheid

De Rules Design Component (RDC) is ontwikkeld voor meerdere platforms:

- Windows
- Unix of omgevingen zoals Unix (bijvoorbeeld Dec Alpha)
- Mainframe omgevingen zoals IBM mainframe en MVS.

Om na te gaan hoe RDC te gebruiken is als een rekenmodule in uw eigen applicaties zoals de polisadministratie verwijzen we u naar de technische documentatie waarin informatie is opgenomen over de Application Programming Interface (API) van RDC.

5.1.1. De dimensies van RDC uitwisselbaarheid

Het RDC systeem is een applicatie die onder verschillende operating systems kan draaien:

- Windows 95;
- Windows 98;
- Windows 2000;
- Windows NT workstation version 4.0;
- Windows NT server version 3.5 – 5.x;
- Unix Operating Systems of Unix-like environments zoals Linux;
- Mainframe environments zoals IBM mainframe;
- OS/2.

Installatie op een Microsoft Windows cliënt of server kan volledig automatisch plaatsvinden door middel van een installatie programma.

Voor de Unix en Mainframe omgeving is er geen 'install and go' methode voor implementatie. De applicatie moet opnieuw gecompileerd worden en geïntegreerd in de applicatie van waaruit RDC aangeroepen wordt, zoals de polisadministratie.

5.1.2. *RDC uitwisselbaar: met en zonder hercompilatie.*

De architectuur van RDC is zo opgezet dat RDC in verschillende soorten omgevingen gebruikt kan worden. Eenvoudig weergegeven bestaat RDC uit drie hoofddelen: de grafische user interface (GUI) component, de berekeningskern en de database.

De GUI-component kan gebruikt worden om RDC formules te componeren met behulp van de grafische formule editor. Deze component is alleen ontwikkeld voor Microsoft Windows en is niet-uitwisselbaar met andere platformen. De GUI-component roept de rekenmodule aan door gebruik te maken van het blackbox concept, zodat formules berekend kunnen worden in de GUI. De GUI-component is uitwisselbaar met alle Windows platformen zonder dat opnieuw gecompileerd moet worden.

De rekenmodule is geschreven in standaard ANSI C++ code. Deze module is geschikt voor verschillende platformen. Als er een C++ compiler aanwezig is kan RDC op ieder platform draaien. Voor Windows is de rekenmodule beschikbaar zonder dat er opnieuw gecompileerd moet worden (wordt geleverd als DLL). Voor alle andere platformen moet het bronbestand van de rekenmodule getransfereerd worden naar het specifieke platform en moeten de bronnen gecompileerd worden naar een executable of een bibliotheek die kan worden aangeroepen door andere programma's.