INTENTIONAL™
SOFTWARE

Democratizing Software Creation

Markus Völter
Independent Consultant for ISC

Henk Kolk
CTO Financial Services, Capgemini

The Challenge

- Business needs to accelerate innovation
- Software is a main vehicle of innovation
- Innovation through software is hampered by communication between domain experts and programmers

© 2007 Intentional Software Corporation

The Key Players

Domain Expert **Programmer**

© 2007 Intentional Software Corporation

Intentional Software™ Proposition

...to fully integrate Domain Experts into software creation.

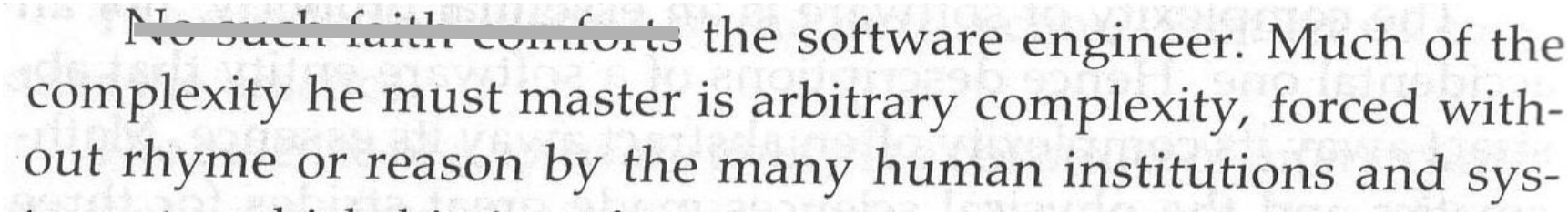
© 2007 Intentional Software Corporation

A brief history of software

A struggle to distinguish and treat separately the problem and the program

© 2007 Intentional Software Corporation

Fred Brooks: “No Silver Bullet“ ‘87



No such faith comforts the software engineer. Much of the complexity he must master is arbitrary complexity, forced without rhyme or reason by the many human institutions and sys-

© 2007 Intentional Software Corporation

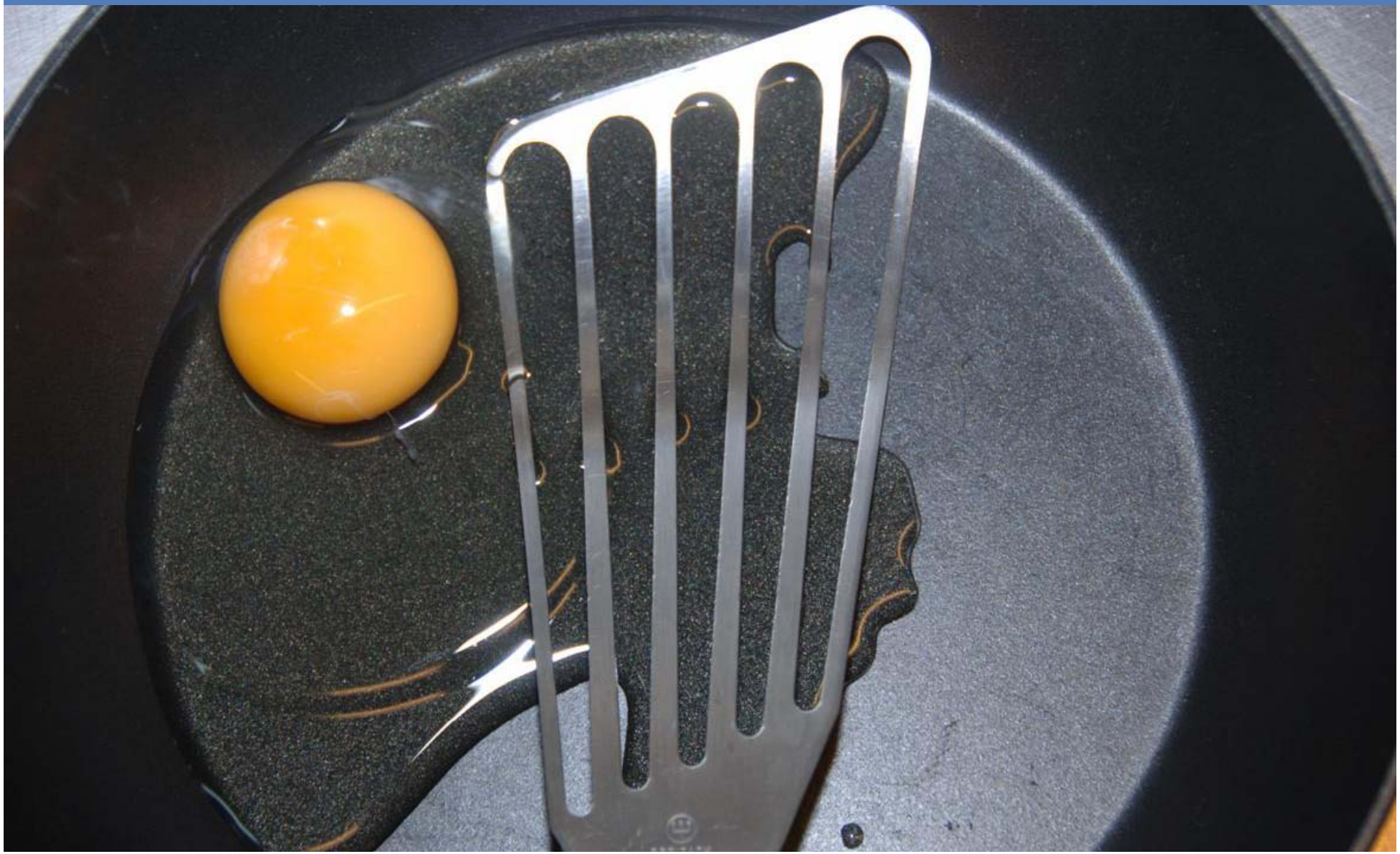
A brief history continued

- When we fail to treat separately the problem and the program
  - The problem and the program get mixed up creating the complexity we hear about
  - We get stuck with improving the resulting mess

© 2007 Intentional Software Corporation

Complexity of scrambled eggs

Input + Process



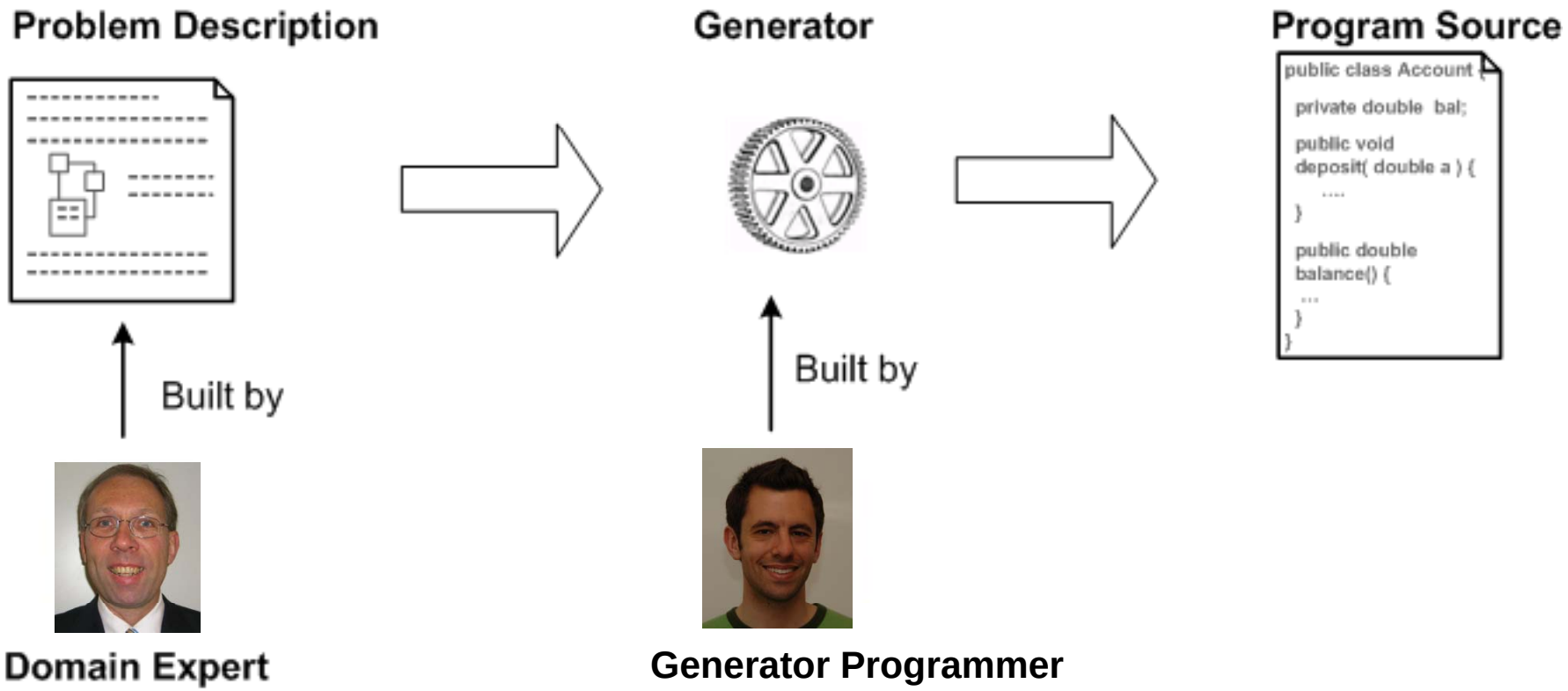
After refactoring ;-)

Domain Orientation trends:

- Domain Specific Languages (DSL)
- Domain Specific Modeling (DSM)
- Domain Driven Design (DDD)
- Model Driven Architecture (MDA)
- Model Driven (Software) Development (MD(S)D)
- Generative Programming (GP)
- Language Oriented Programming (LOP)
- Code Generation
- Meta Programming (Ruby)
- ...

© 2007 Intentional Software Corporation

DSL + Code Generation



© 2007 Intentional Software Corporation

Key benefits of domain orientation

- Domain Expert participation feasible - business knowledge isolated from technology
- Complexity is reduced – separation of concerns
- Programmers create a more valuable artifact – capture process instead of "messy" result

© 2007 Intentional Software Corporation

What prevents mainstream use?

1. Integrate Domain Experts fully
   - Matching existing notations
   - Mixing graphical/textual notations
2. Multi-domain
   - Compose independent domains
   - References between domains
3. Domain evolution
4. Groupware

© 2007 Intentional Software Corporation

Intentional Software™ Proposition

Through

Intentional Domain Workbench ™

bring the current trend of domain orientation to a new level to *fully integrate Domain Experts into software creation*.

© 2007 Intentional Software Corporation

Intentional Software™ Technology

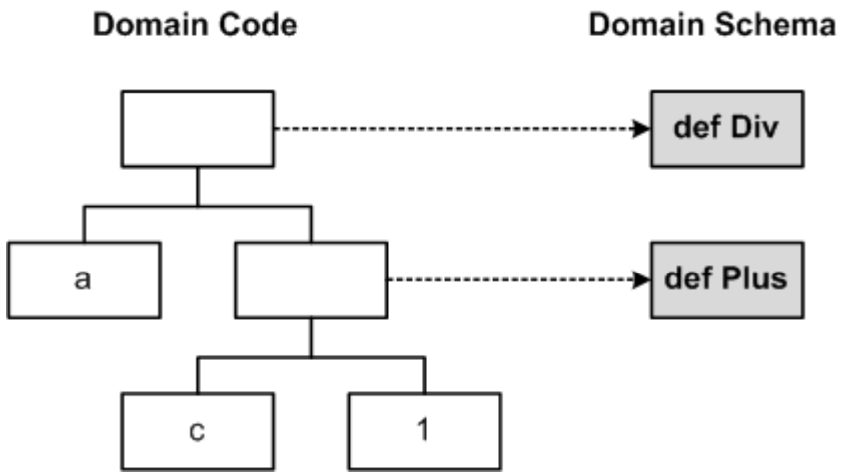
- **Intentional Domain Workbench ™**
  - Intentional Tree domain representation
  - Projectional multi-view editing

© 2007 Intentional Software Corporation

Intentional Tree

- Extendible, uniform representation
- Strong identities throughout
- No fixed meta-levels
- Versioned storage
- Separated concerns

```
return a = b / (c + 1);
```



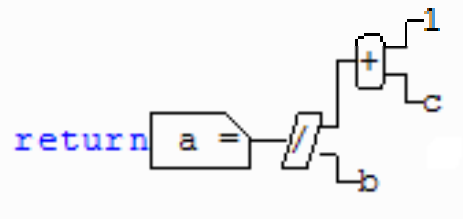

```
Return
(
   Assign
   (
      a,
      Div
      (
         b,
         Plus
         (
            c,
            1
         )
      )
   )
)
```

© 2007 Intentional Software Corporation

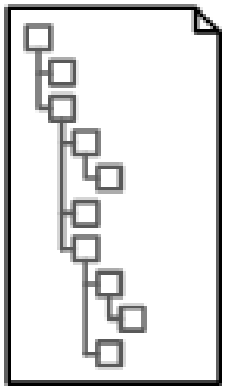
Projectional editing

```
return a = b / (c + 1);
```

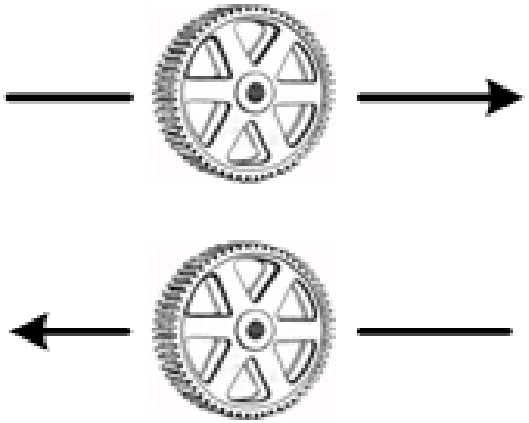
$$\text{return } a = \frac{b}{c + 1};$$



Domain Code



Projector



Display / Input

© 2007 Intentional Software Corporation

Integrate Domain Experts

- Matching existing notations
- Mixing notation graphics / text

- Projectional editor supplies appropriate notation
- Projectional editor decouples domain code from notation
  - Multi-view, embedding, extension…
- Graphics / text are treated uniformly
- Notation can change on domain or other selected boundaries

© 2007 Intentional Software Corporation

Multi domain

- Compose independent domains
- References between domains

- Tree structure accommodates composition
- Inter-domain references connecting domains

© 2007 Intentional Software Corporation

Domain evolution

- Tree storage is independent of schema – will not "break" if schema changes
- Notation can keep up with evolution
- Further parametrization is always possible
- Independent concerns can be added without interfering with others

© 2007 Intentional Software Corporation

Groupware

- Tree storage requires rethinking groupware
  - Change logs for fully general solution
  - Edit "conflicts" are a "mini domain" – integrated with notations
  - Versioning and audit trails
- Familiar metaphors – versions, branches, merges, checkins

© 2007 Intentional Software Corporation

Intentional Domain Workbench™ Status

- Technology is fully capable of handling the Domain Workbench requirements.
- We are nearing operational use in selected domains.
- We are working with selected OEM customers only, for example with Capgemini.

© 2007 Intentional Software Corporation

Henk Kolk

CTO Financial Services Capgemini

Problems for Pension Companies

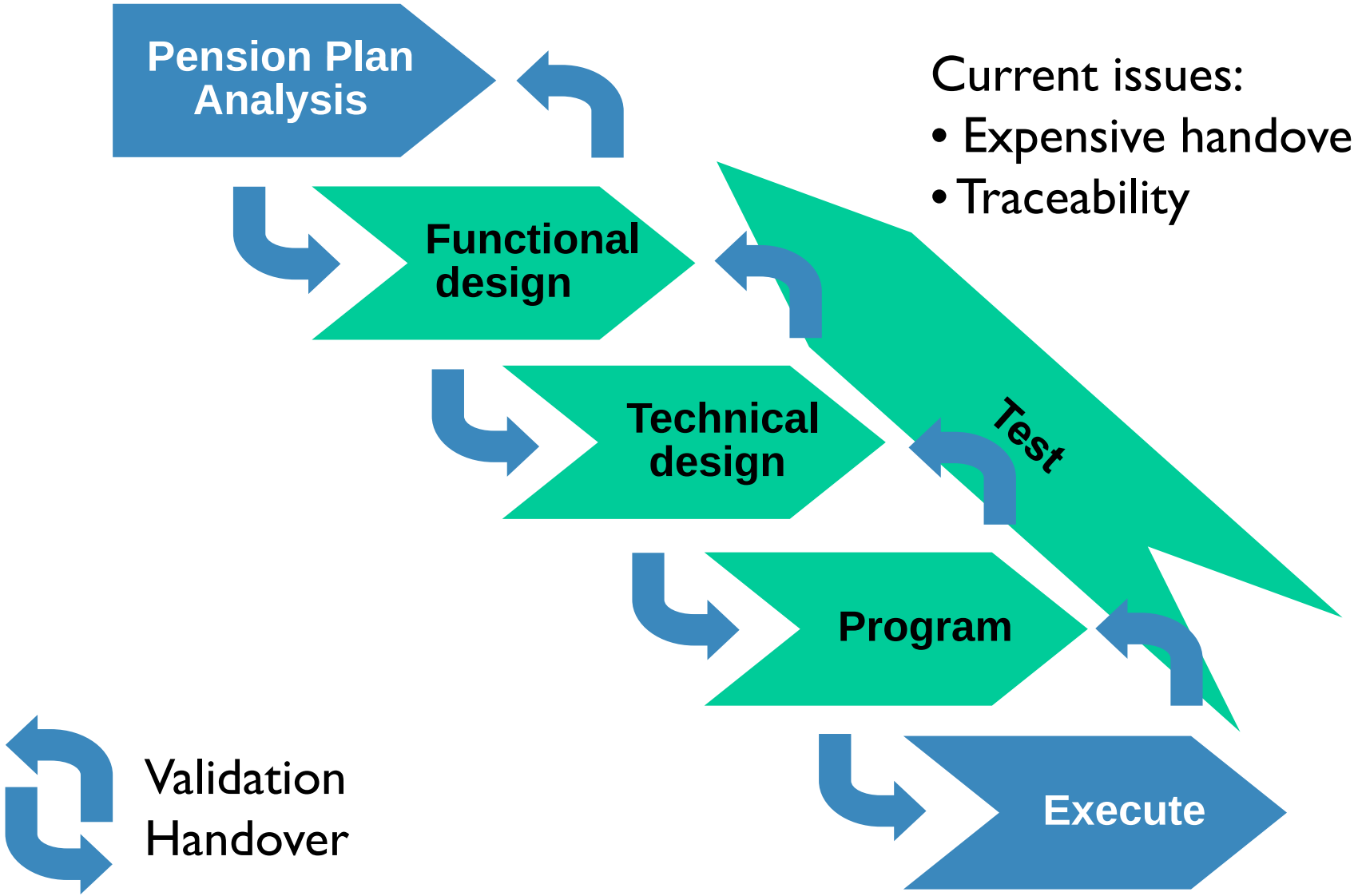
Need for pension product innovation

- Governmental interest
  - New Pension Laws
- Mergers
- Transparency

Problems

- Time to market
- Abstract product models
- Ensuring quality

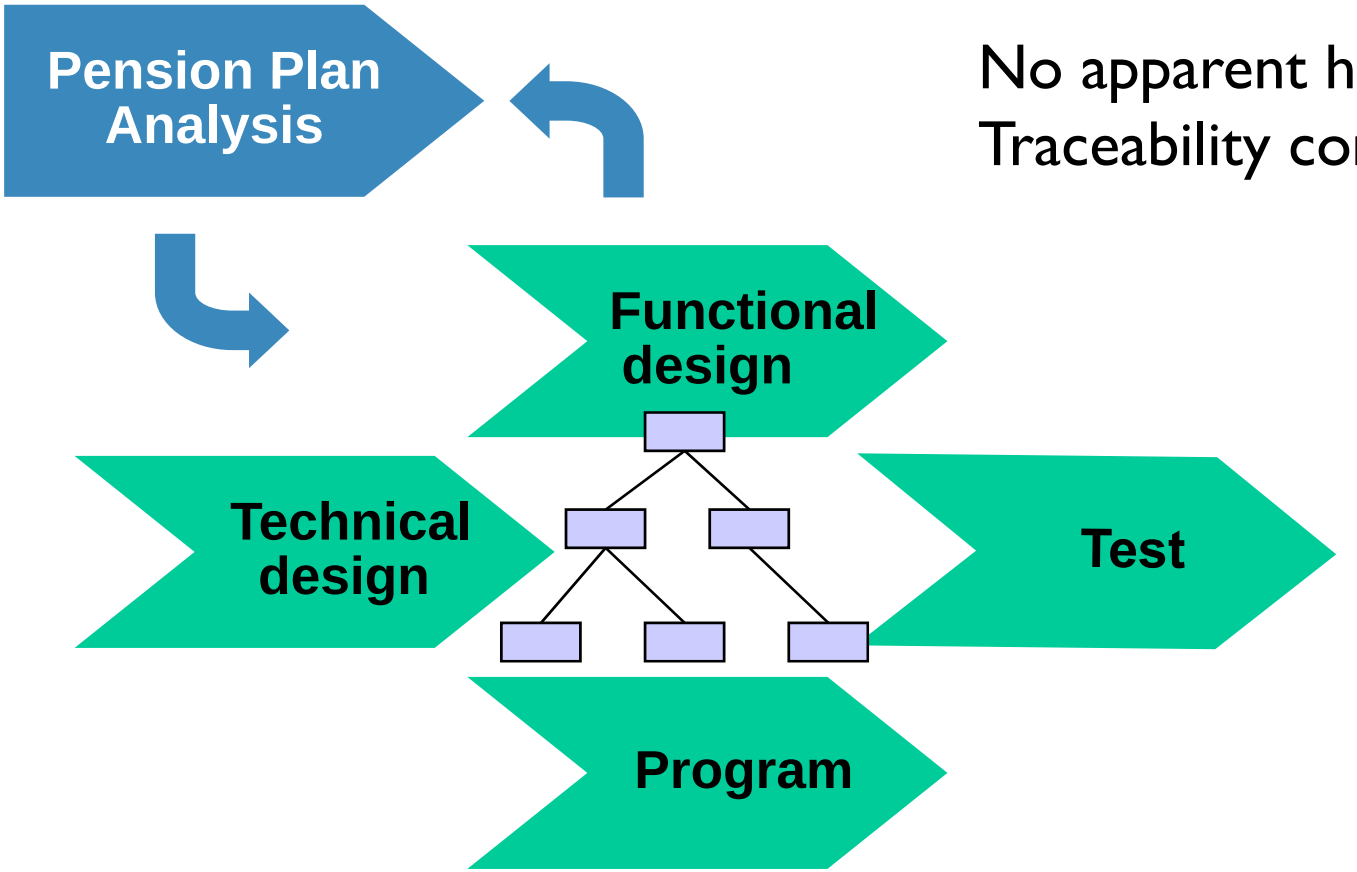
Old way: disconnected domains



Current issues:

- Expensive handovers
- Traceability

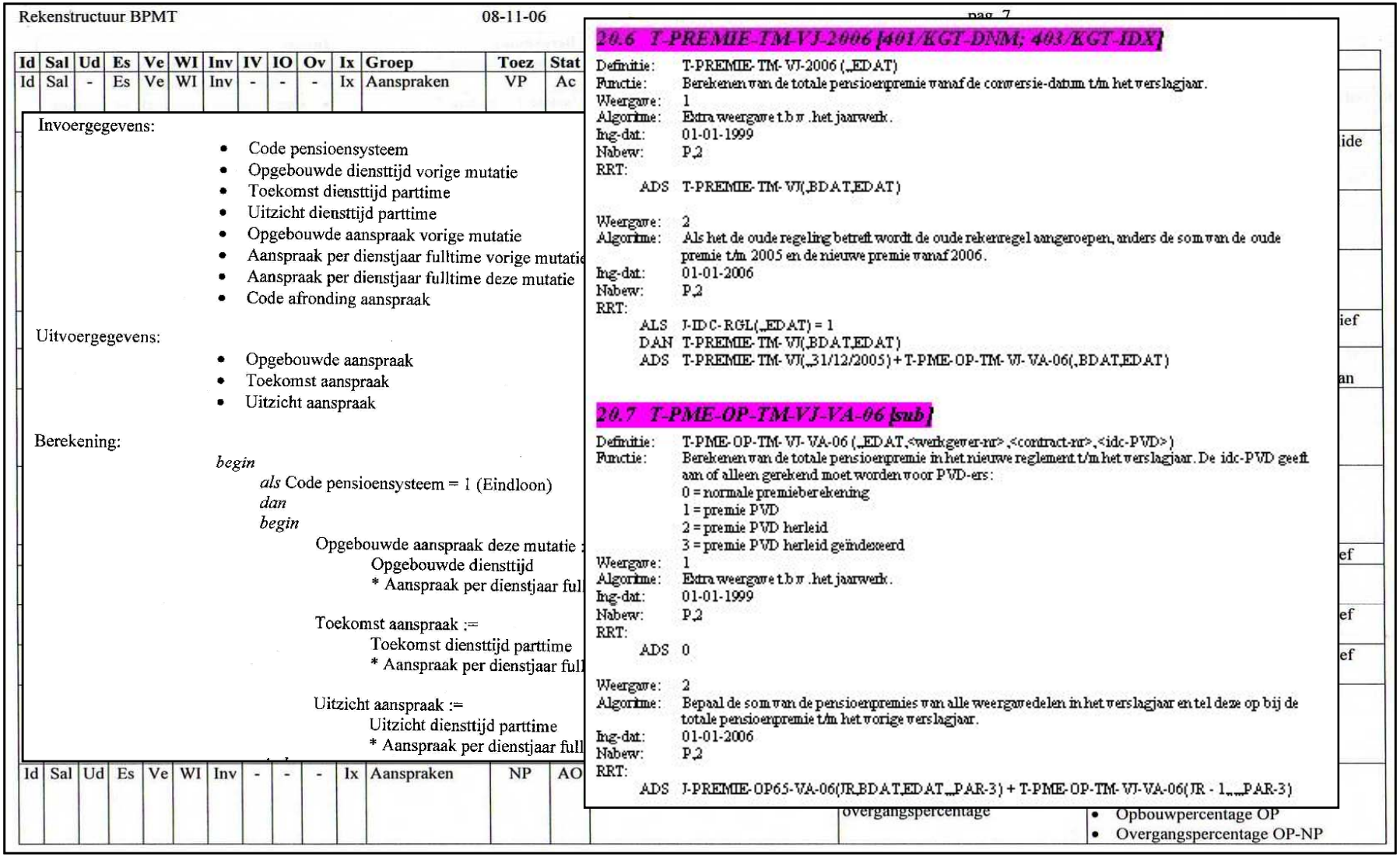
New way: connected domains



No apparent handover
Traceability comes for free

Validation & Handover

Old way: Excel & Word



Rekenstructuur BPMT 08-11-06

Id	Sal	Ud	Es	Ve	WI	Inv	IV	IO	Ov	Ix	Groep	Toez	Stat
Id	Sal	-	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	VP	Ac

Invoergegevens:

- Code pensioensysteem
- Opgebouwde diensttijd vorige mutatie
- Toekomst diensttijd parttime
- Uitzicht diensttijd parttime
- Opgebouwde aanspraak vorige mutatie
- Aanspraak per dienstjaar fulltime vorige mutatie
- Aanspraak per dienstjaar fulltime deze mutatie
- Code afronding aanspraak

Uitvoergegevens:

- Opgebouwde aanspraak
- Toekomst aanspraak
- Uitzicht aanspraak

Berekening:

begin
als Code pensioensysteem = 1 (Eindloon)
dan
begin
Opgebouwde aanspraak deze mutatie :
Opgebouwde diensttijd
* Aanspraak per dienstjaar ful

Toekomst aanspraak :=
Toekomst diensttijd parttime
* Aanspraak per dienstjaar full

Uitzicht aanspraak :=
Uitzicht diensttijd parttime
* Aanspraak per dienstjaar full

Id	Sal	Ud	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	NP	AO

20.6 T-PREMIE-TM-VJ-2006 [401/KGT-DNM; 403/KGT-IDX]

Definitie: T-PREMIE-TM-VJ-2006 („EDAT)
Functie: Berekenen van de totale pensioenpremie vanaf de conversie-datum t/m het verslagjaar.
Weergave: 1
Algoritme: Extra weergave t.b.v. het jaarwerk.
Ing-dat: 01-01-1999
Nabew: P,2
RRT:
ADS T-PREMIE-TM-VJ(,BDAT,EDAT)

Weergave: 2
Algoritme: Als het de oude regeling betreft wordt de oude rekenregel aangeroepen, anders de som van de oude premie t/m 2005 en de nieuwe premie vanaf 2006.
Ing-dat: 01-01-2006
Nabew: P,2
RRT:
ALS I-IDC-RGL(„EDAT) = 1
DAN T-PREMIE-TM-VJ(,BDAT,EDAT)
ADS T-PREMIE-TM-VJ(„31/12/2005) + T-PME-OP-TM-VJ-VA-06(,BDAT,EDAT)

20.7 T-PME-OP-TM-VJ-VA-06 [sub]

Definitie: T-PME-OP-TM-VJ-VA-06 („EDAT,<werkgever-nr>,<contract-nr>,<idc-PVD>)
Functie: Berekenen van de totale pensioenpremie in het nieuwe reglement t/m het verslagjaar. De idc-PVD geeft aan of alleen gerekend moet worden voor PVD-ers:
0 = normale premieberekening
1 = premie PVD
2 = premie PVD herleid
3 = premie PVD herleid geïndexeerd
Weergave: 1
Algoritme: Extra weergave t.b.v. het jaarwerk.
Ing-dat: 01-01-1999
Nabew: P,2
RRT:
ADS 0

Weergave: 2
Algoritme: Bepaal de som van de pensioenpremies van alle weergavedelen in het verslagjaar en tel deze op bij de totale pensioenpremie t/m het vorige verslagjaar.
Ing-dat: 01-01-2006
Nabew: P,2
RRT:
ADS I-PREMIE-OP65-VA-06(JR,BDAT,EDAT„PAR-3) + T-PME-OP-TM-VJ-VA-06(JR - 1„„PAR-3)

overgangspercentage

- Opbouwpercentage OP
- Overgangspercentage OP-NP

New way: Pension Workbench

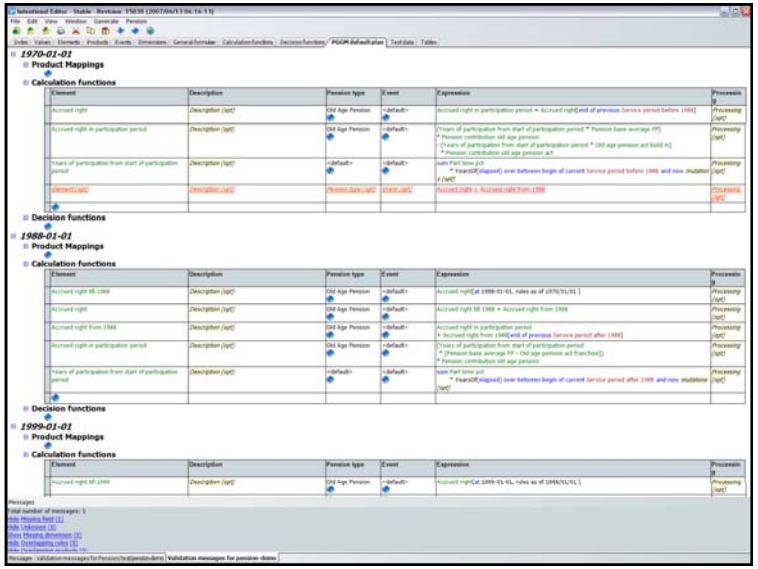
- Matching existing notations
  - Pension experts record pension world in their notations

Old spreadsheet

Rekenstructuur BPMT 08-11-06 pag. 7

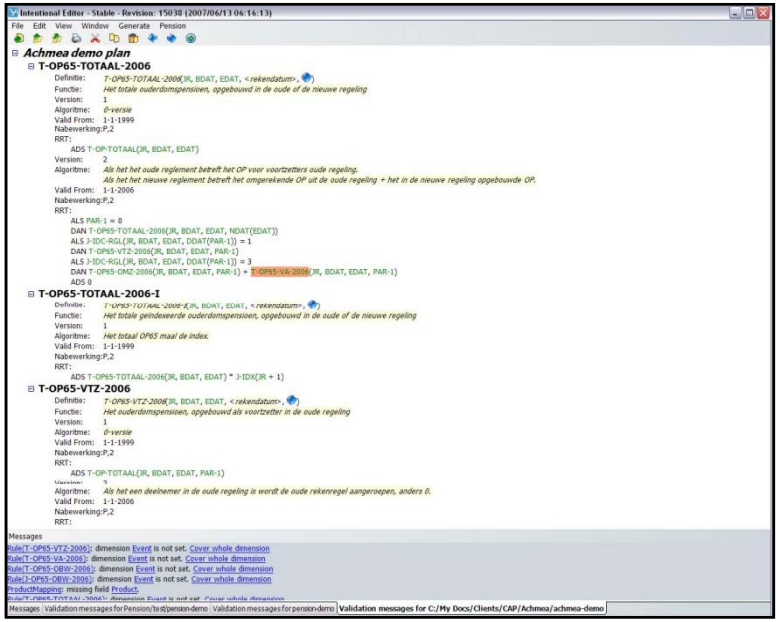
Id	Sal	Ud	Es	Ve	WI	Inv	IV	IO	Ov	Ix	Groep	Toez	Stat	Resultaat	Berekening	Invoer
Id	Sal	-	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	VP	Ac	Uitzicht aanspraak VP actief	diensttijd * grondslag * opbouwpercentage	• Uitzicht diensttijd VP actief • Pensioengrondslag VP actief • Opbouwpercentage VP
Id	Sal	Ud	Es	Ve	WI	Inv	IV	-	-	Ix	Aanspraken	VP	AO	Opgebouwde aanspraak VP invalide	diensttijd * grondslag * opbouwpercentage	• Opgebouwde diensttijd VP invalide • Pensioengrondslag VP invalide • Opbouwpercentage VP
Id	Sal	Ud	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	VP	AO	Toekomst aanspraak VP invalide	diensttijd * grondslag * opbouwpercentage	• Toekomst diensttijd VP invalide • Pensioengrondslag VP invalide • Opbouwpercentage VP
Id	Sal	Ud	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	VP	AO	Uitzicht aanspraak VP invalide	diensttijd * grondslag * opbouwpercentage	• Uitzicht diensttijd VP invalide • Pensioengrondslag VP invalide • Opbouwpercentage VP
-	-	Ud	-	-	-	-	-	-	-	Ix	Aanspraken	VP	Pv	Opgebouwde aanspraak VP premievrij	resultaat : = input	• Opgebouwde aanspraak VP actief
-	-	-	-	-	-	-	-	-	-	Ix	Aanspraken	VP	Pv	Opgebouwde aanspraak VP premievrij	aanspraak * (1 + percentage/100)	• Vorige ingegane aanspraak VP • Indexatiepercentage niet-ingegaan
-	-	-	-	Ve	-	-	-	-	-	-	Aanspraken	VP	Pv	Verevende aanspraak VP vanuit actief	(bedrag 1 - bedrag 2) * percentage	• Opgebouwd VP actief • Opgebouwd VP actief op huwelijksdatum • Vereveningspercentage
-	-	-	-	Ve	-	-	-	-	-	-	Aanspraken	VP	Pv	Verevende aanspraak VP vanuit invalide	(bedrag 1 - bedrag 2) * percentage	• Opgebouwd VP invalide • Opgebouwd VP invalide op huwelijksdatum • Vereveningspercentage
-	-	-	-	-	-	-	IV	-	-	-	Aanspraken	VP	Ing	Ingegane aanspraak VP	resultaat : = input	• Opgebouwde aanspraak VP actief
-	-	-	-	-	-	-	-	-	-	Ix	Aanspraken	VP	Ing	Ingegane aanspraak VP	aanspraak * (1 + percentage/100)	• Vorige ingegane aanspraak VP • Indexatiepercentage ingegaan
Id	Sal	Ud	Es	Ve	WI	Inv	IV	-	Ov	Ix	Aanspraken	NP	Ac	Opgebouwde aanspraak NP actief	basisaanspraak * overgangspercentage	• Opgebouwde aanspraak OP actief • Overgangspercentage OP-NP
Id	Sal	Ud	Es	Ve	WI	Inv	IV	-	Ov	Ix	Aanspraken	NP	Ac	Opgebouwde aanspraak NP actief	bedrag 1 -/- bedrag 2	• Opgebouwde aanspraak NP actief • Bijzonder NP actief totaal
Id	Sal	-	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	NP	Ac	Toekomst aanspraak NP actief	diensttijd * grondslag * opbouwpercentage * overgangspercentage	• Toekomst diensttijd NP actief • Pensioengrondslag NP • Opbouwpercentage OP • Overgangspercentage OP-NP
Id	Sal	Ud	Es	Ve	WI	Inv	-	-	-	Ix	Aanspraken	NP	AO	Toekomst aanspraak NP invalide	diensttijd * grondslag * opbouwpercentage * overgangspercentage	• Toekomst diensttijd NP • Pensioengrondslag NP • Opbouwpercentage OP • Overgangspercentage OP-NP

Pension Workbench

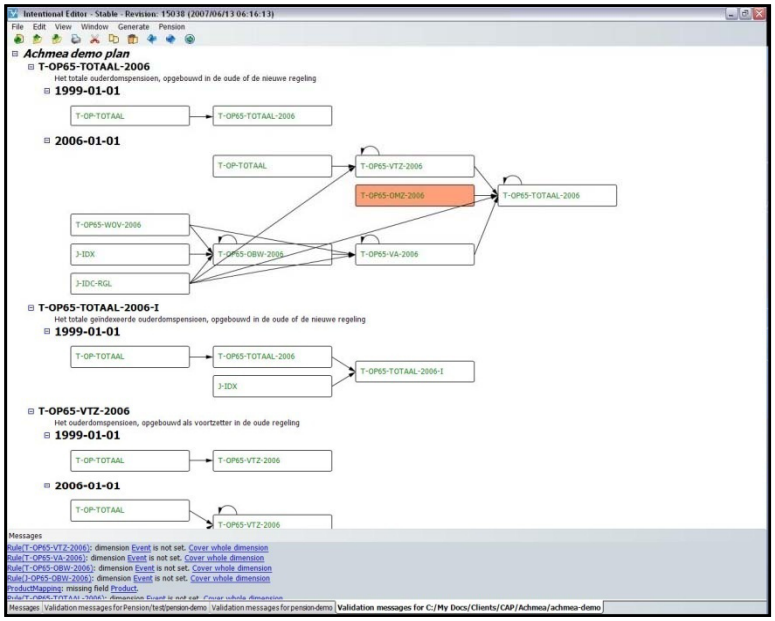


Multiple Views with Graphics

Pension Plan versions

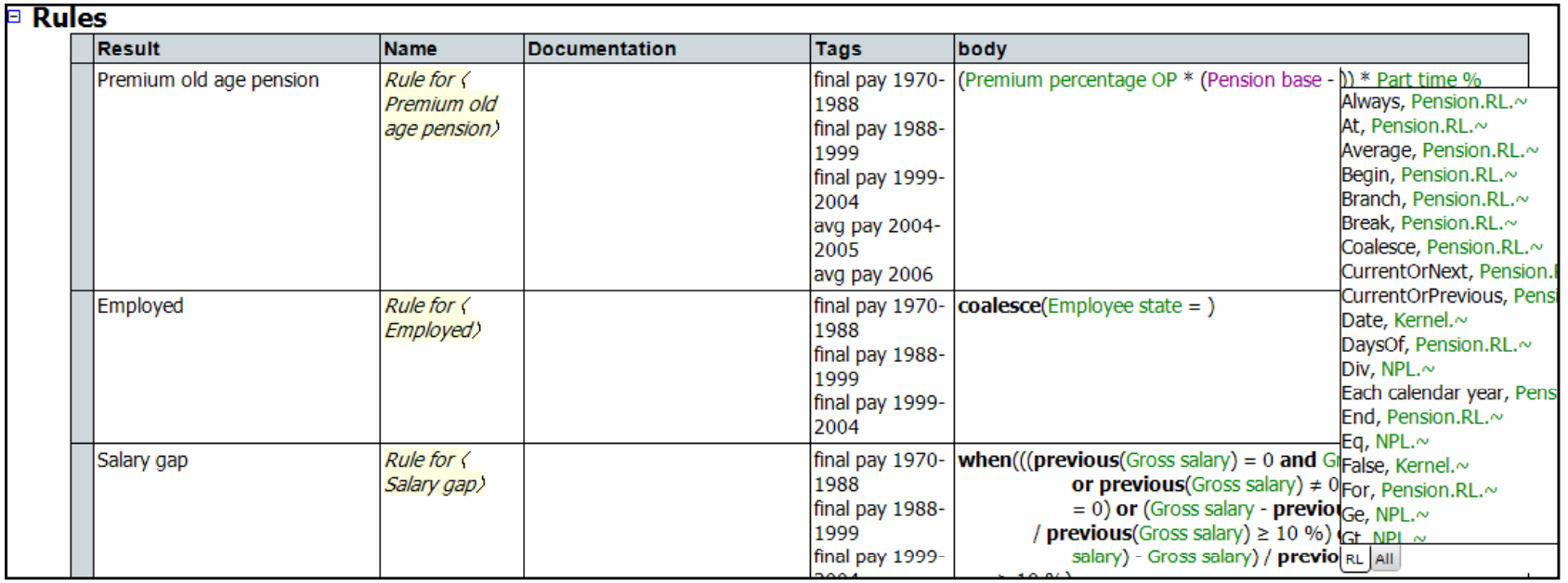


Rule dependencies



Compose Business Domain

- Domain Schema
- Projectional Editors



Rules

Result	Name	Documentation	Tags	body
Premium old age pension	*Rule for (Premium old age pension)*		final pay 1970-1988 final pay 1988-1999 final pay 1999-2004 avg pay 2004-2005 avg pay 2006	(Premium percentage OP * (Pension base -
Employed	*Rule for (Employed)*		final pay 1970-1988 final pay 1988-1999 final pay 1999-2004	**coalesce**(Employee state =)
Salary gap	*Rule for (Salary gap)*		final pay 1970-1988 final pay 1988-1999 final pay 1999-	**when**(((**previous**(Gross salary) = 0 **and** G **or previous**(Gross salary) ≠ 0 = 0) **or** (Gross salary - **previo** / **previous**(Gross salary) ≥ 10 %) salary) - Gross salary) / **previo**

)) * Part time %

Always, Pension.RL.~
At, Pension.RL.~
Average, Pension.RL.~
Begin, Pension.RL.~
Branch, Pension.RL.~
Break, Pension.RL.~
Coalesce, Pension.RL.~
CurrentOrNext, Pension.
CurrentOrPrevious, Pens
Date, Kernel.~
DaysOf, Pension.RL.~
Div, NPL.~
Each calendar year, Pens
End, Pension.RL.~
Eq, NPL.~
False, Kernel.~
For, Pension.RL.~
Ge, NPL.~
Gt, NPL.~
RL All

Integrate Rule Test Domain

- Unit Tests for pension rules
- Real time evaluation

Salary gap	*Rule for (Salary gap)*		final pay 1970-1988 final pay 1988-1999 final pay 1999-2004	**when**(((**previous**(Gross salary) = 0 **and** Gross salary ≠ 0 **or previous**(Gross salary) ≠ 0 **and** Gross salary = 0) **or** (Gross salary - **previous**(Gross salary)) / **previous**(Gross salary) ≥ 10 %) **or** (**previous**(Gross salary) - Gross salary) / **previous**(Gross salary) ≥ 10 %)

Name	Documentation	Tags	Valid time	Fixture	Expected value	Actual value
			1990-1-31	Piet Van Dijk	**true**	**true**
			1990-1-31	Jan De Jong	**true**	Nil
			1991-1-31	Piet Van Dijk	Nil	Nil
			1992-1-31	Piet Van Dijk	Nil	Nil

Integrate System Test Domain

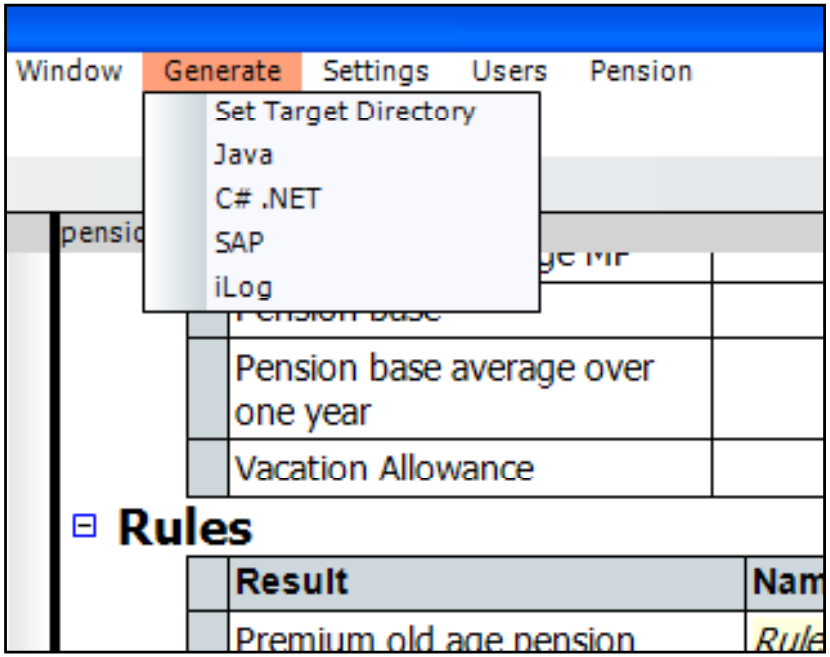
- Test cascading rules and their interrelations
- Real time evaluation

Test cases

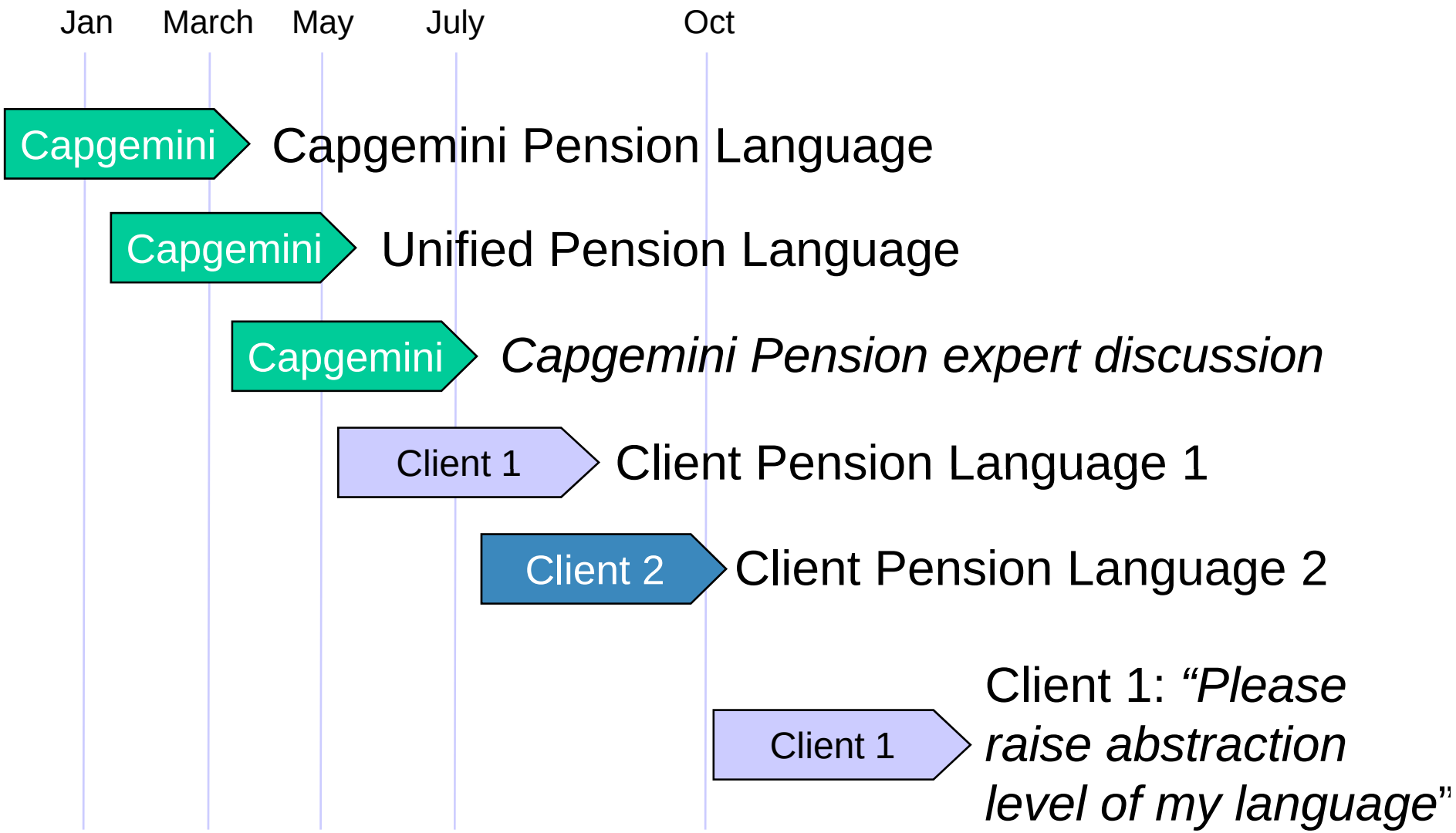
Name	Documentation	Tags	Valid time	Transaction time	Fixture	Product	Element	Expected value	Actual value
Accrued right at retireme			2006-12-31	2007-9-24	Jan De Jong	Old Age Pension	Accrued right	761.0402	761.0402
Accrued Right last final pay			2004-1-1	2007-9-24	Jan De Jong	Old Age Pension	Accrued right	705.0589	705.0589
premiun last year			2006-1-1	2007-9-24	Jan De Jong	Old Age Pension	Premium old age pension	329.0625	329.0625
Accrued right at retireme 2)			2006-12-31	2007-9-24	Piet Van Dijk	Old Age Pension	Accrued right	740.94	724.7658

Build Code Generators

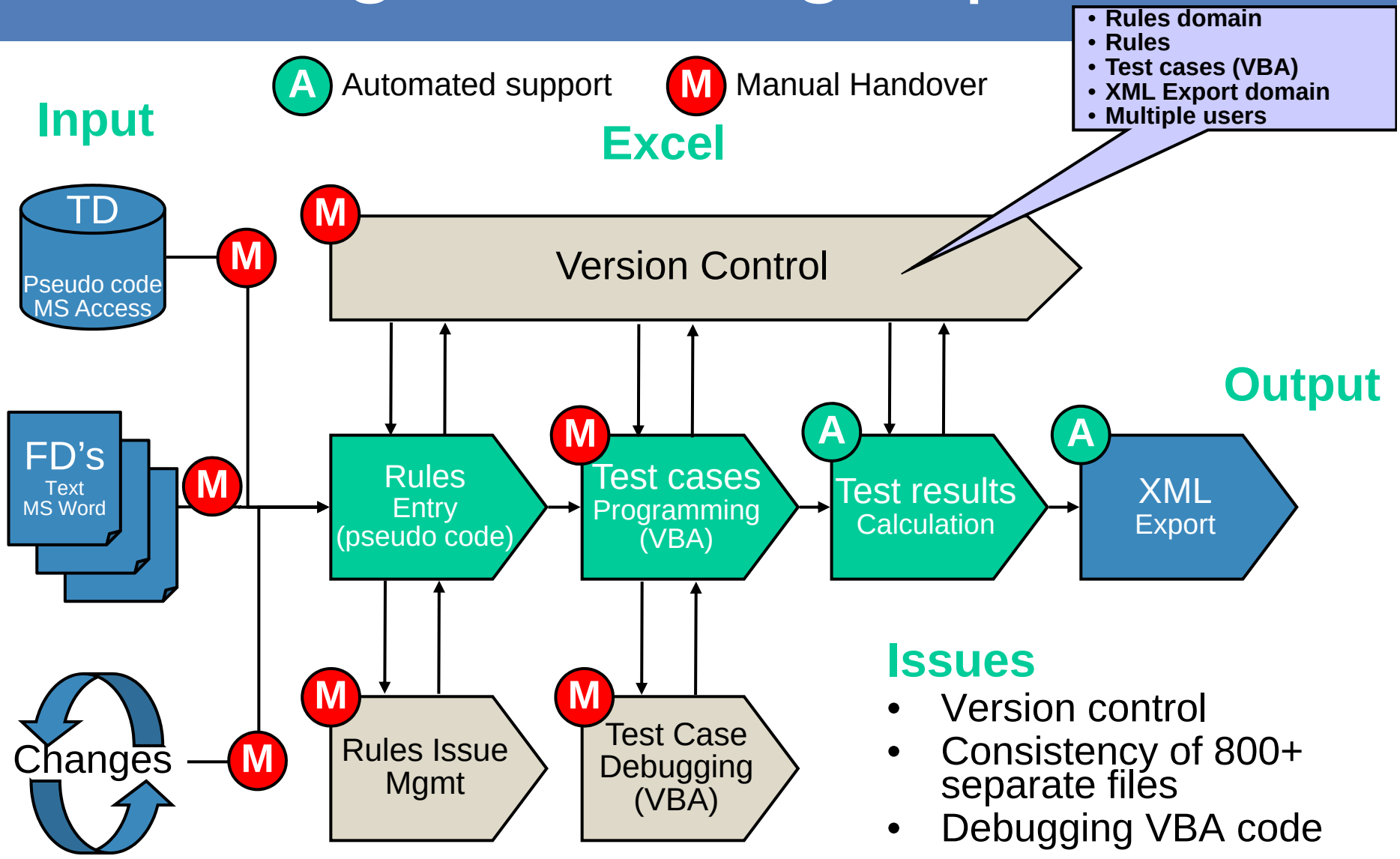
- Multiple implementation target languages



Domain Language Evolution



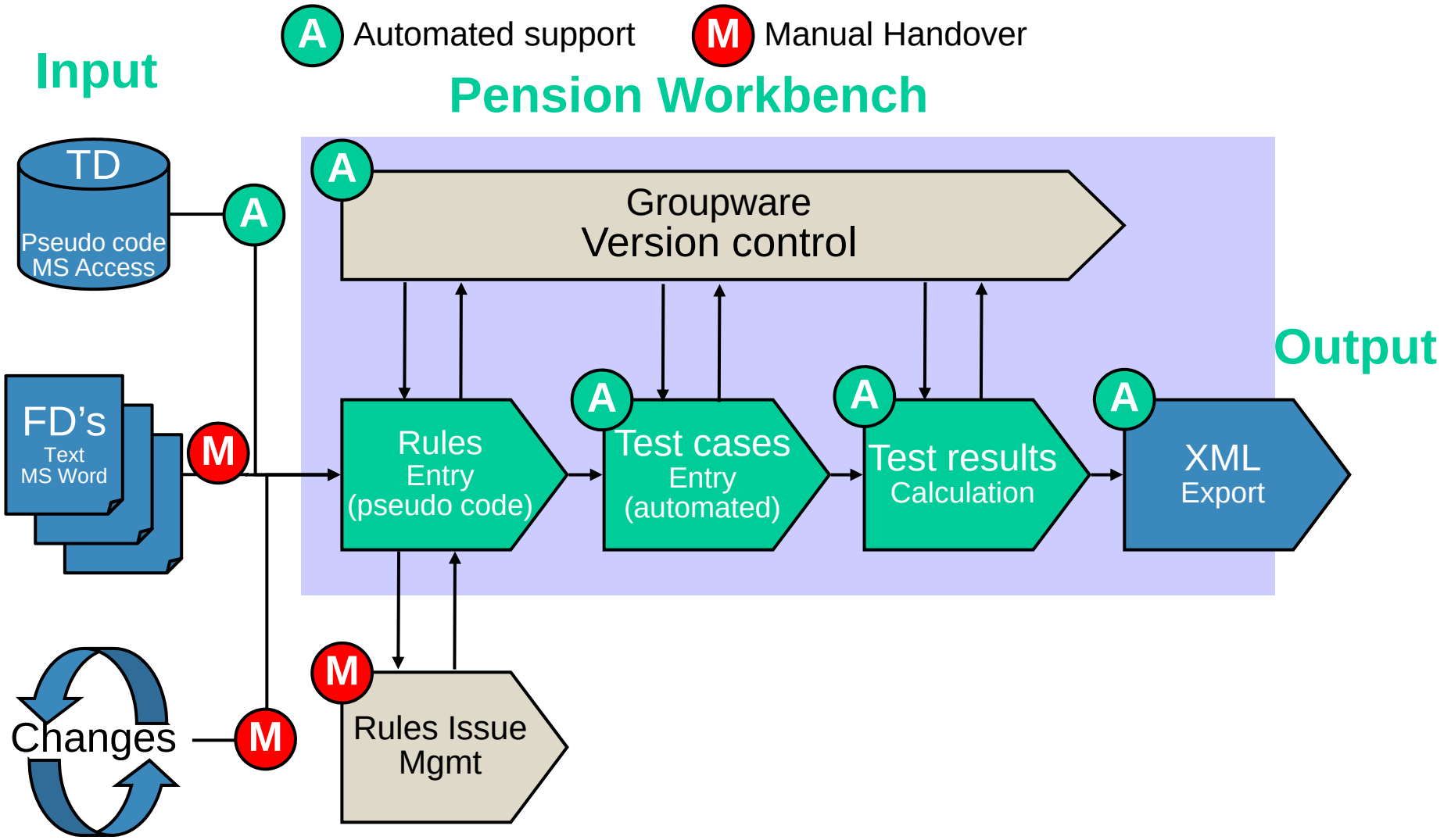
Testing – Lack of "groupware"



Issues

- Version control
- Consistency of 800+ separate files
- Debugging VBA code

New Groupware



Integrating Pension Experts

Herman Gerbscheid, Pension Architect:

- "This is the stuff I had to do mentally and keep consistent in my head all the time. It's great to finally have tools for it."

Suzanne Pront, Pension Expert:

- "Normally I know what I want, but don't know how to tell engineers. Now I can do this myself. This is a revolution!"

Sybren den Hartog, Java Architect:

- "Now we can generate business rules and domain structure, which we could not do in UML based MDA."

Summary

- Intentional Software is helping us to accelerate Pension Product innovation for our clients
- We were able to demonstrate a radical change in time to market and quality
- We used Pensions as a pilot, but we see many opportunities in other domains