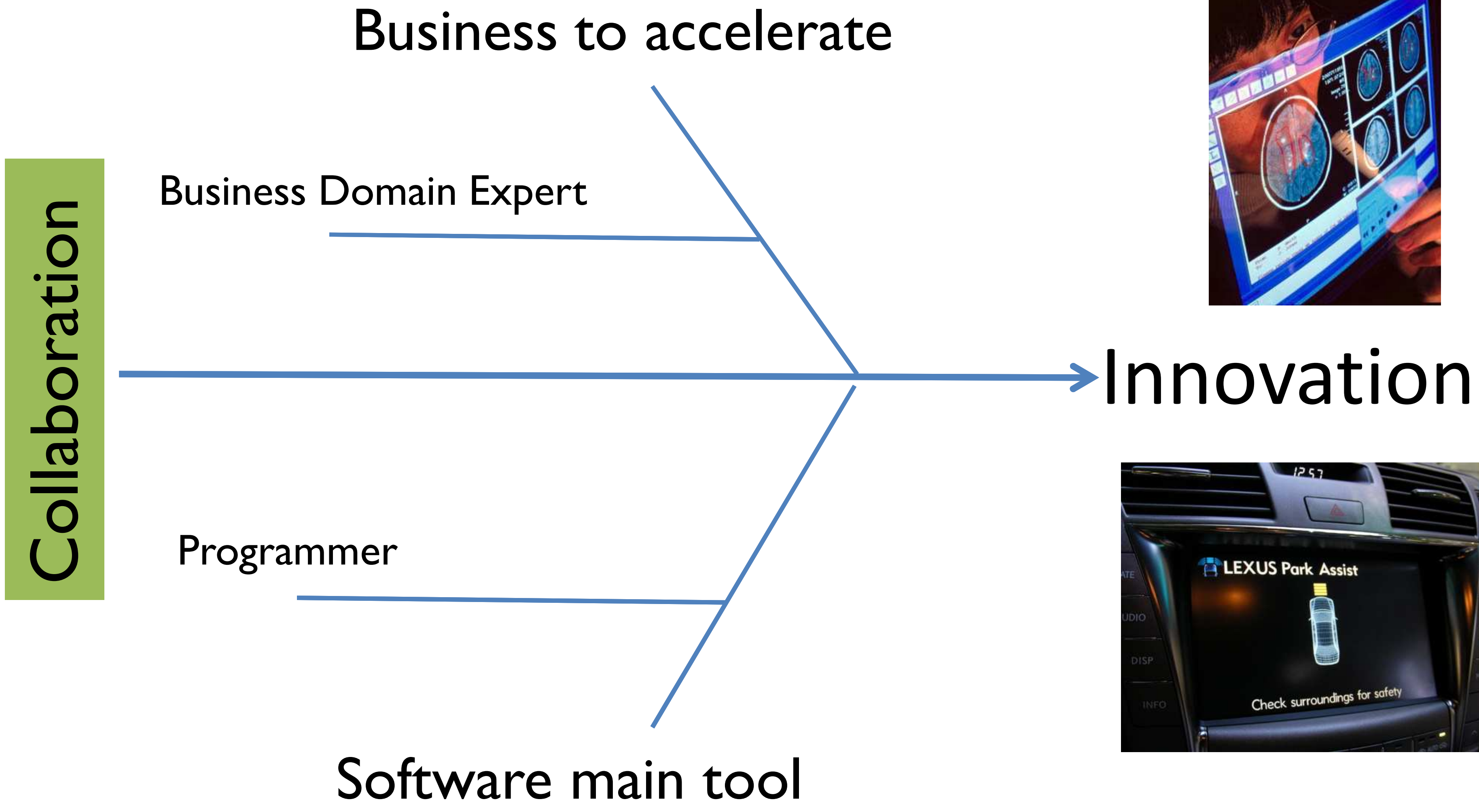
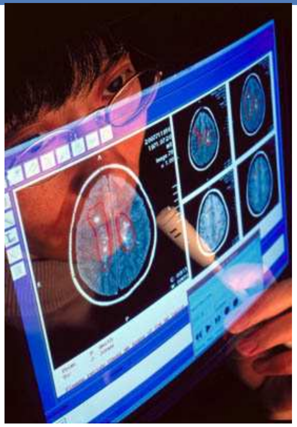
INTENTIONAL™
SOFTWARE

Domain Expert DSLs

Magnus Christerson
Intentional Software Corporation
Henk Kolk
CTO Financial Services, Capgemini

The Challenge





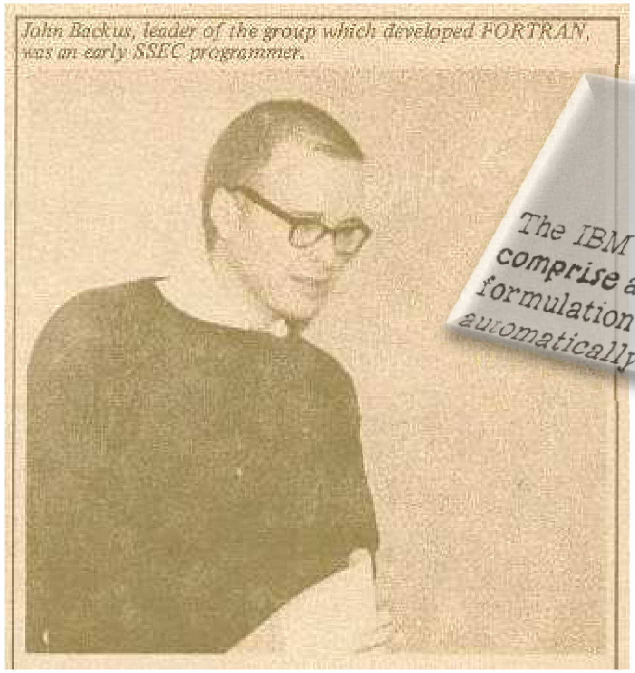
The Key Players

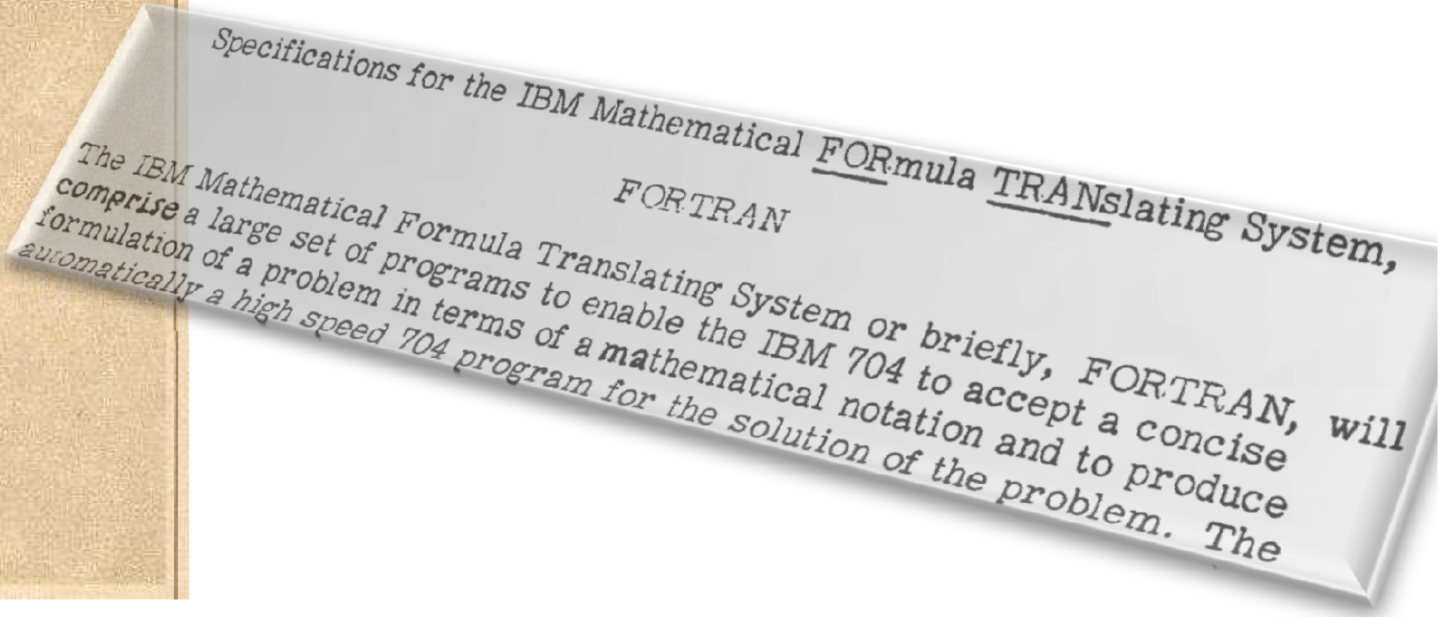
Domain Expert **Programmer**

A brief history of software

- A struggle to distinguish and treat separately problem and program

1954:

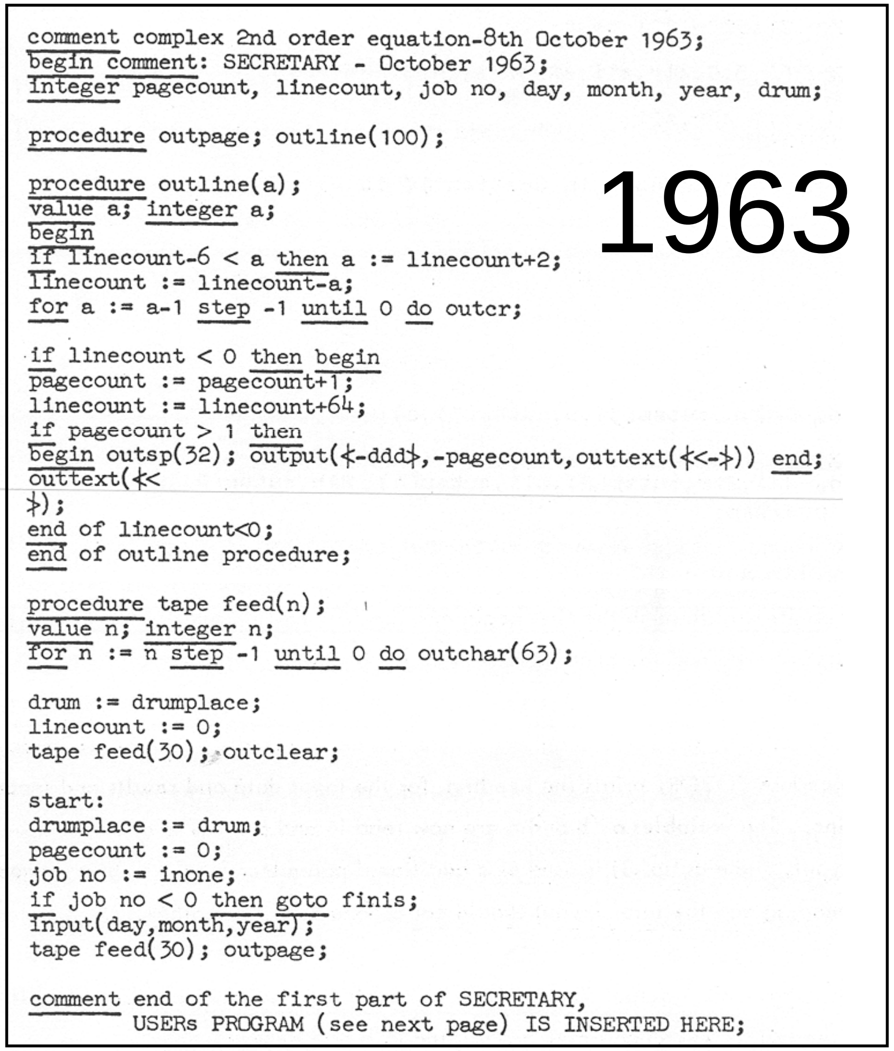

John Backus, leader of the group which developed FORTRAN, was an early SSEC programmer.


Specifications for the IBM Mathematical FORmula TRANSlating System,

FORTRAN

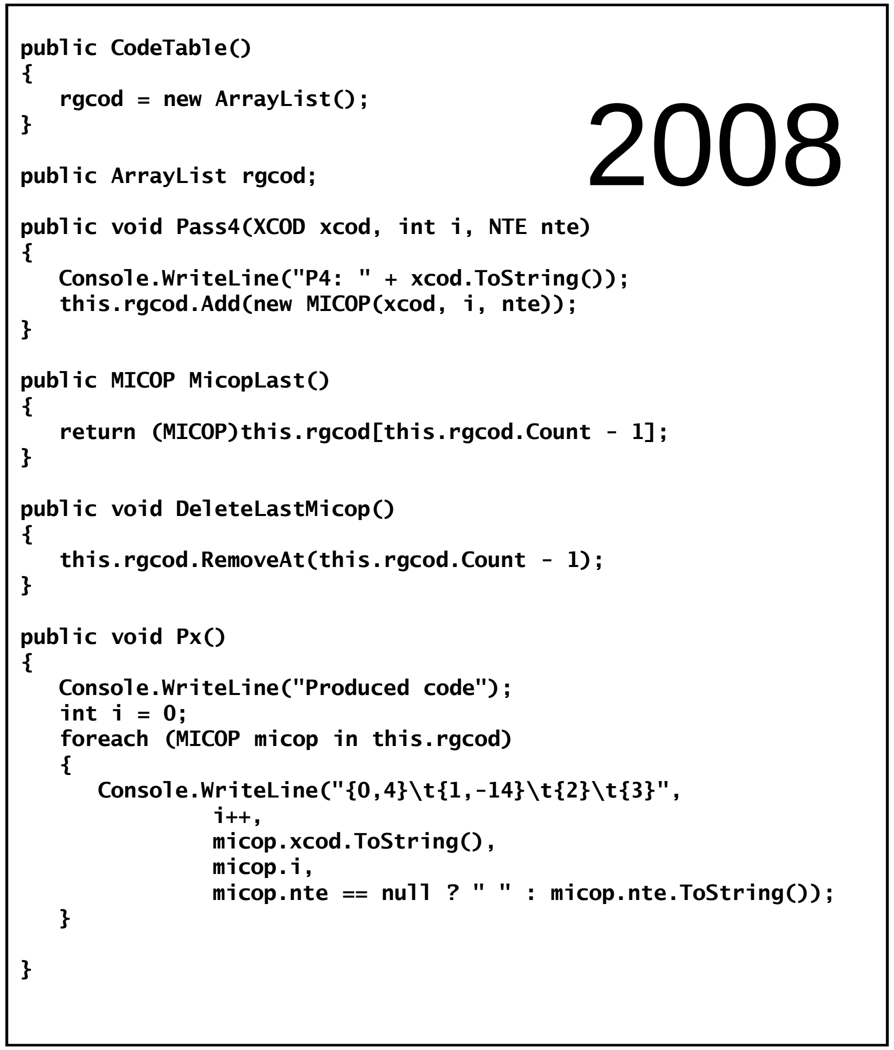
The IBM Mathematical Formula Translating System or briefly, FORTRAN, will comprise a large set of programs to enable the IBM 704 to accept a concise formulation of a problem in terms of a mathematical notation and to produce automatically a high speed 704 program for the solution of the problem. The

Software progress?



1963

```
comment complex 2nd order equation-8th October 1963;
begin comment: SECRETARY - October 1963;
integer pagecount, linecount, job no, day, month, year, drum;

procedure outpage; outline(100);

procedure outline(a);
value a; integer a;
begin
if linecount-6 < a then a := linecount+2;
linecount := linecount-a;
for a := a-1 step -1 until 0 do outcr;

if linecount < 0 then begin
pagecount := pagecount+1;
linecount := linecount+64;
if pagecount > 1 then
begin outsp(32); output(‹-ddd›,-pagecount,outtext(‹<-›)) end;
outtext(‹<
›);
end of linecount<0;
end of outline procedure;

procedure tape feed(n);
value n; integer n;
for n := n step -1 until 0 do outchar(63);

drum := drumplace;
linecount := 0;
tape feed(30); outclear;

start:
drumplace := drum;
pagecount := 0;
job no := inone;
if job no < 0 then goto finis;
input(day,month,year);
tape feed(30); outpage;

comment end of the first part of SECRETARY,
        USERs PROGRAM (see next page) IS INSERTED HERE;
```



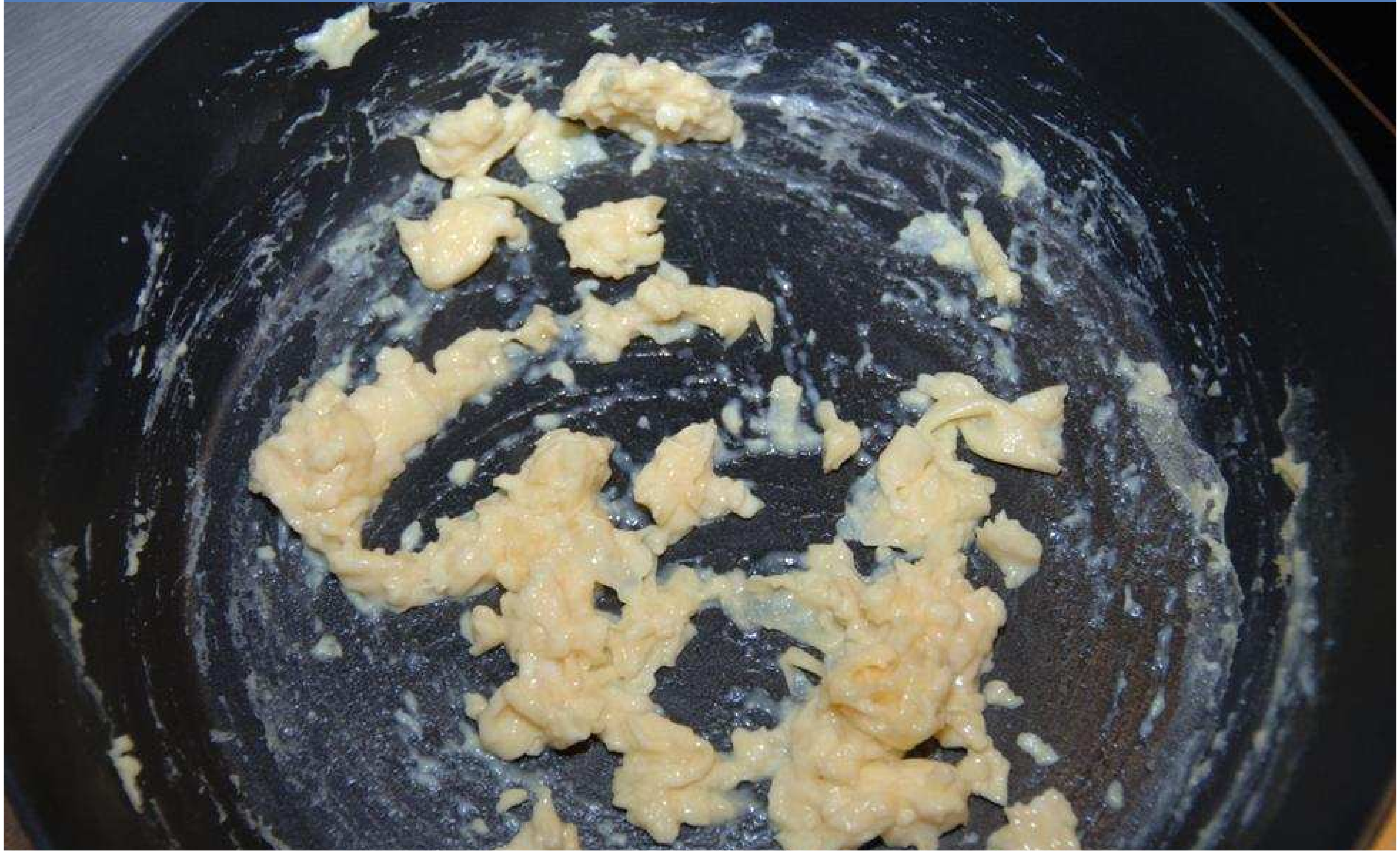
2008

```
public CodeTable()
{
   rgcod = new ArrayList();
}

public ArrayList rgcod;

public void Pass4(XCOD xcod, int i, NTE nte)
{
   Console.WriteLine("P4: " + xcod.ToString());
   this.rgcod.Add(new MICOP(xcod, i, nte));
}

public MICOP MicopLast()
{
   return (MICOP)this.rgcod[this.rgcod.Count - 1];
}

public void DeleteLastMicop()
{
   this.rgcod.RemoveAt(this.rgcod.Count - 1);
}

public void Px()
{
   Console.WriteLine("Produced code");
   int i = 0;
   foreach (MICOP micop in this.rgcod)
   {
      Console.WriteLine("{0,4}\t{1,-14}\t{2}\t{3}",
               i++,
               micop.xcod.ToString(),
               micop.i,
               micop.nte == null ? " " : micop.nte.ToString());
   }

}
```

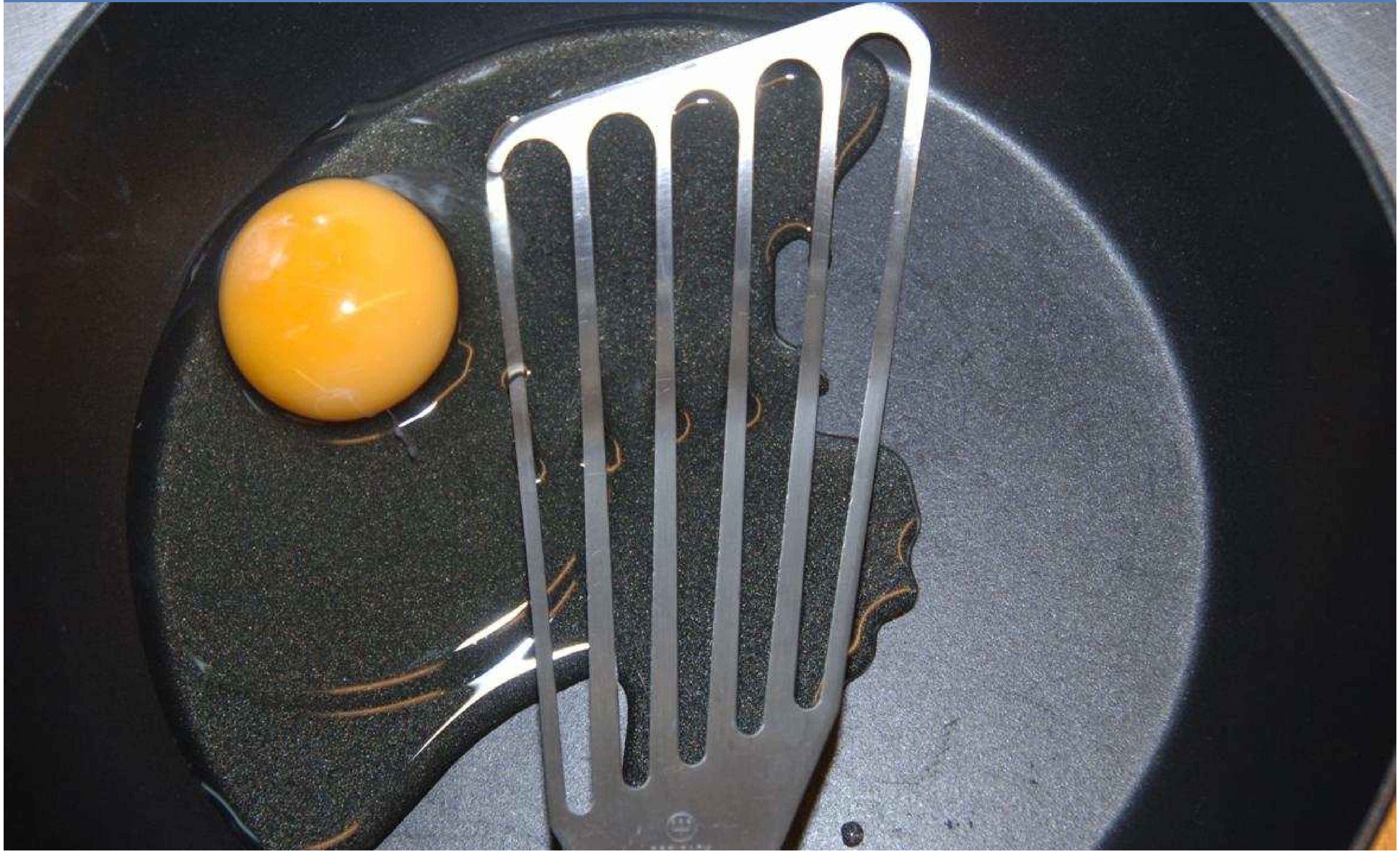
A brief history continued

- 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
  - Complexity becomes (problem x program), not (problem + program)
  - We get stuck with improving the resulting complex mess

Complexity of scrambled eggs



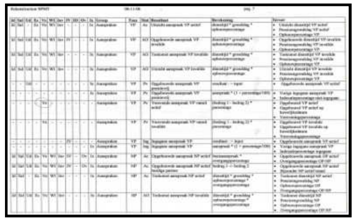
Input + Process



After refactoring ;-)

Software Development Today

Domain Knowledge



Edits

Domain Expert

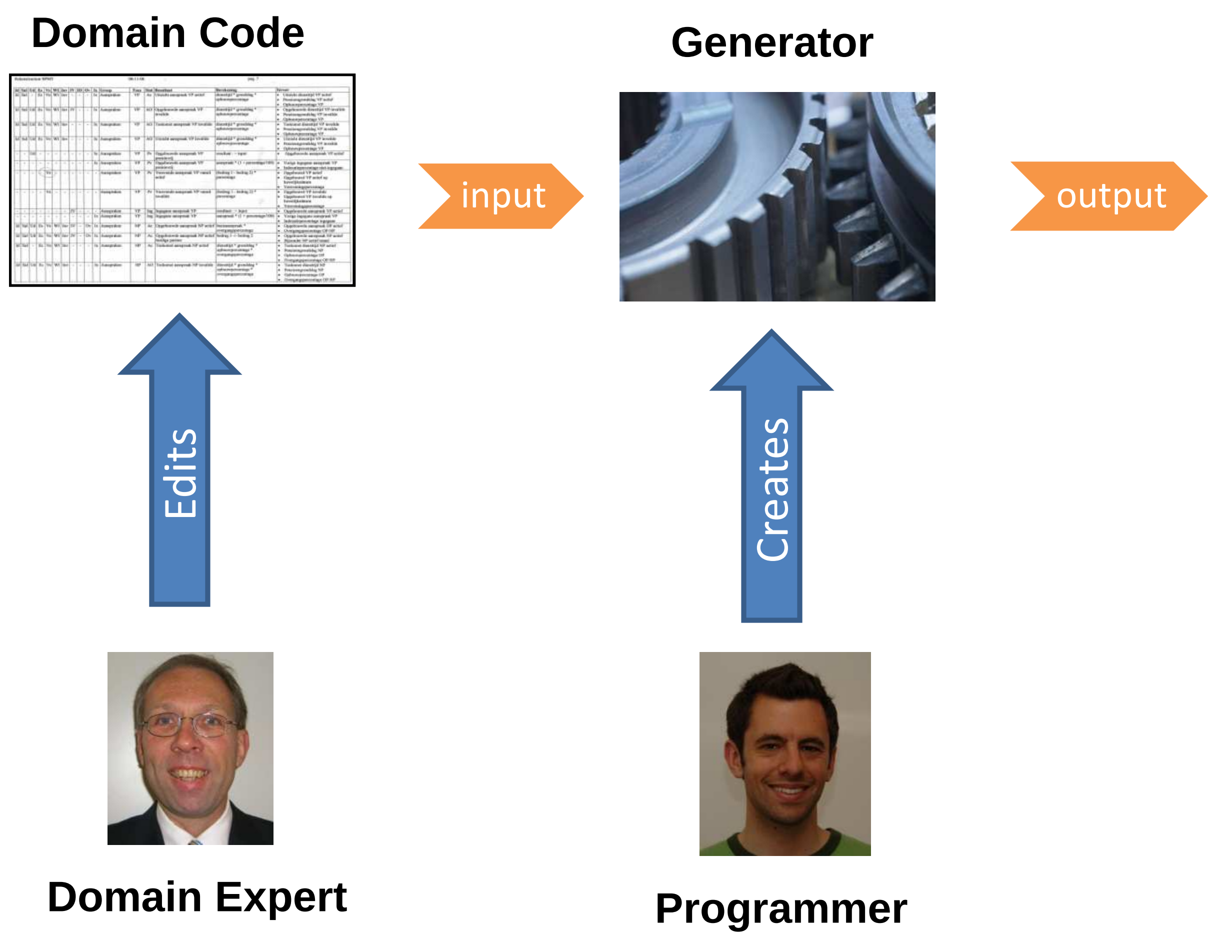
Explains

Programmer

Edits

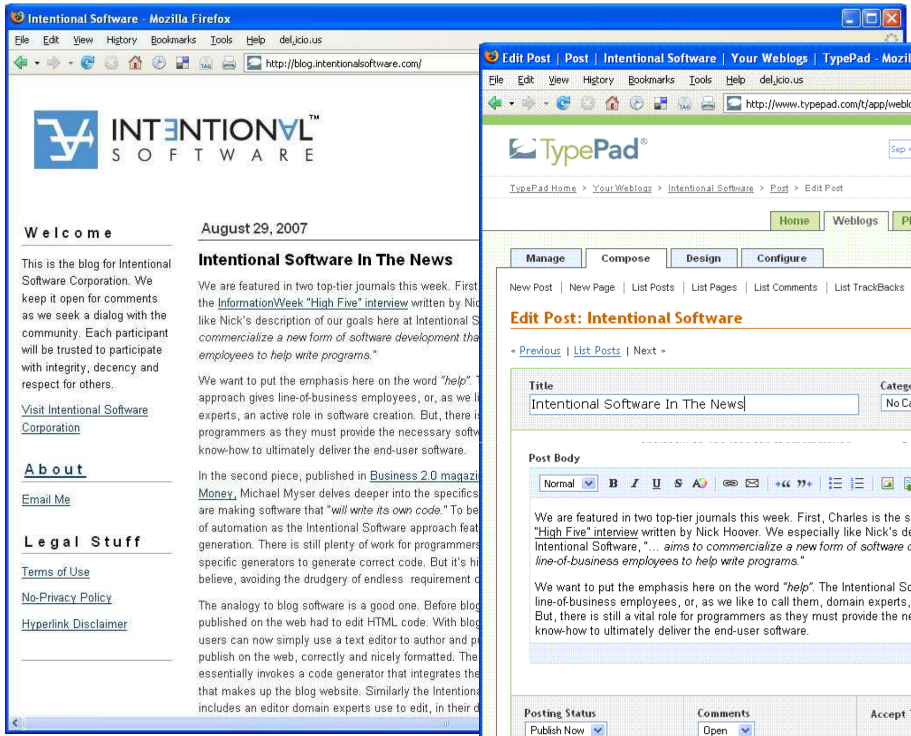
```
public CodeTable()
{
   rgcod = new ArrayList();
}

public ArrayList rgcod;

public void Pass4(XCOD xcod, int i,
NTE nte)
{
   Console.WriteLine("P4: " +
xcod.ToString());
   this.rgcod.Add(new MICOP(xcod,
i, nte));
}

public MICOP MicopLast()
{
   return
(MICOP)this.rgcod[this.rgcod.Count
- 1];
}

public void DeleteLastMicop()
{

this.rgcod.RemoveAt(this.rgcod.Coun
t - 1);
}

public void Px()
{
   Console.WriteLine("Produced
code");
   int i = 0;
   foreach (MICOP micop in
this.rgcod)
   {

Console.WriteLine("{0,4}\t{1,-
14}\t{2}\t{3}",
                    i++,

                    micop.xcod.ToSt
ring(),
                    micop.i,
                    micop.nte ==
null ? " " : micop.nte.ToString());
   }

}
```

Intentional: Input + Process



```
public CodeTable()
{
   rgcod = new ArrayList();
}

public ArrayList rgcod;

public void Pass4(XCOD xcod, int i,
NTE nte)
{
   Console.WriteLine("P4: " +
xcod.ToString());
   this.rgcod.Add(new MICOP(xcod,
i, nte));
}

public MICOP MicopLast()
{
   return
(MICOP)this.rgcod[this.rgcod.Count
- 1];
}

public void DeleteLastMicop()
{

this.rgcod.RemoveAt(this.rgcod.Coun
t - 1);
}

public void Px()
{
   Console.WriteLine("Produced
code");
   int i = 0;
   foreach (MICOP micop in
this.rgcod)
   {

Console.WriteLine("{0,4}\t{1,-
14}\t{2}\t{3}",
                    i++,

                    micop.xcod.ToSt
ring(),
                    micop.i,
                    micop.nte ==
null ? " " : micop.nte.ToString());
   }

}
```

Analogy: Blog Software


Intentional Software - Mozilla Firefox
File Edit View History Bookmarks Tools Help del.icio.us
http://blog.intentionalsoftware.com/
INTENTIONAL SOFTWARE

Welcome

This is the blog for Intentional Software Corporation. We keep it open for comments as we seek a dialog with the community. Each participant will be trusted to participate with integrity, decency and respect for others.

Visit Intentional Software Corporation

About

Email Me

Legal Stuff

Terms of Use

No-Privacy Policy

Hyperlink Disclaimer

August 29, 2007

Intentional Software In The News

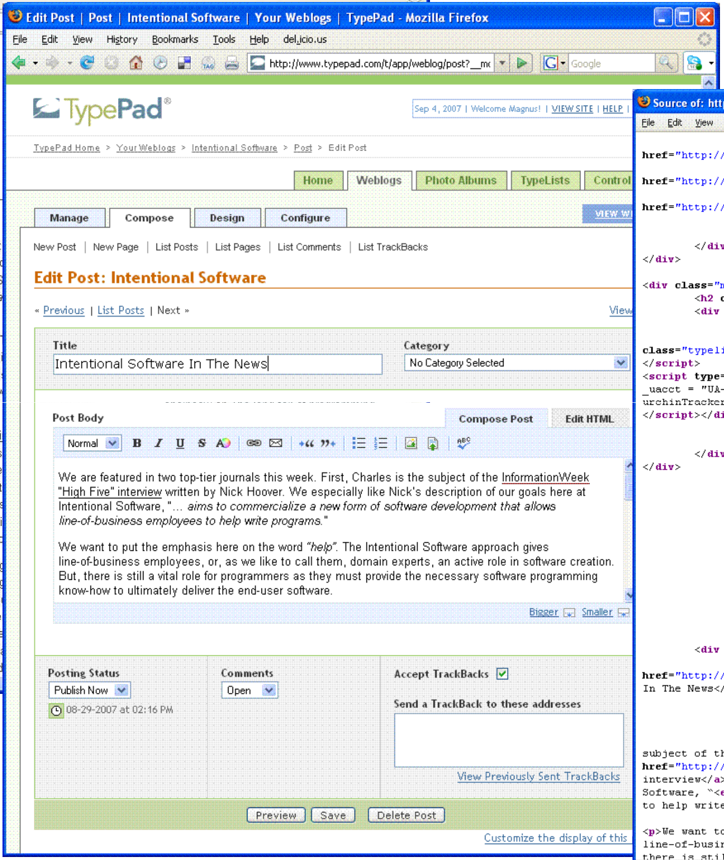
We are featured in two top-tier journals this week. First
the InformationWeek "High Five" interview written by Nic
like Nick's description of our goals here at Intentional S
commercialize a new form of software development tha
employees to help write programs."

We want to put the emphasis here on the word *"help"*. T
approach gives line-of-business employees, or, as we li
experts, an active role in software creation. But, there i
programmers as they must provide the necessary softw
know-how to ultimately deliver the end-user software.

In the second piece, published in Business 2.0 magazi
Money, Michael Myser delves deeper into the specifics
are making software that "*will write its own code.*" To be
of automation as the Intentional Software approach feat
generation. There is still plenty of work for programmers
specific generators to generate correct code. But it's hi
believe, avoiding the drudgery of endless requirement c

The analogy to blog software is a good one. Before blog
published on the web had to edit HTML code. With blog
users can now simply use a text editor to author and pu
publish on the web, correctly and nicely formatted. The
essentially invokes a code generator that integrates the
that makes up the blog website. Similarly the Intentiona
includes an editor domain experts use to edit, in their d

As viewed


Edit Post | Post | Intentional Software | Your Weblogs | TypePad - Mozilla Firefox
File Edit View History Bookmarks Tools Help del.icio.us
http://www.typepad.com/t/app/weblog/post?__mc
Google
TypePad
Sep 4, 2007 | Welcome Magnus! | VIEW SITE | HELP |
TypePad Home > Your Weblogs > Intentional Software > Post > Edit Post
Home
Weblogs
Photo Albums
TypeLists
Control
Manage
Compose
Design
Configure
VIEW WE
New Post | New Page | List Posts | List Pages | List Comments | List TrackBacks

Edit Post: Intentional Software

« Previous | List Posts | Next »

Title: Intentional Software In The News

Category: No Category Selected

Post Body

Compose Post
Edit HTML
Normal

We are featured in two top-tier journals this week. First, Charles is the subject of the InformationWeek "High Five" interview written by Nick Hoover. We especially like Nick's description of our goals here at Intentional Software, "... *aims to commercialize a new form of software development that allows line-of-business employees to help write programs.*"

We want to put the emphasis here on the word *"help"*. The Intentional Software approach gives line-of-business employees, or, as we like to call them, domain experts, an active role in software creation. But, there is still a vital role for programmers as they must provide the necessary software programming know-how to ultimately deliver the end-user software.

Posting Status: Publish Now

08-29-2007 at 02:16 PM

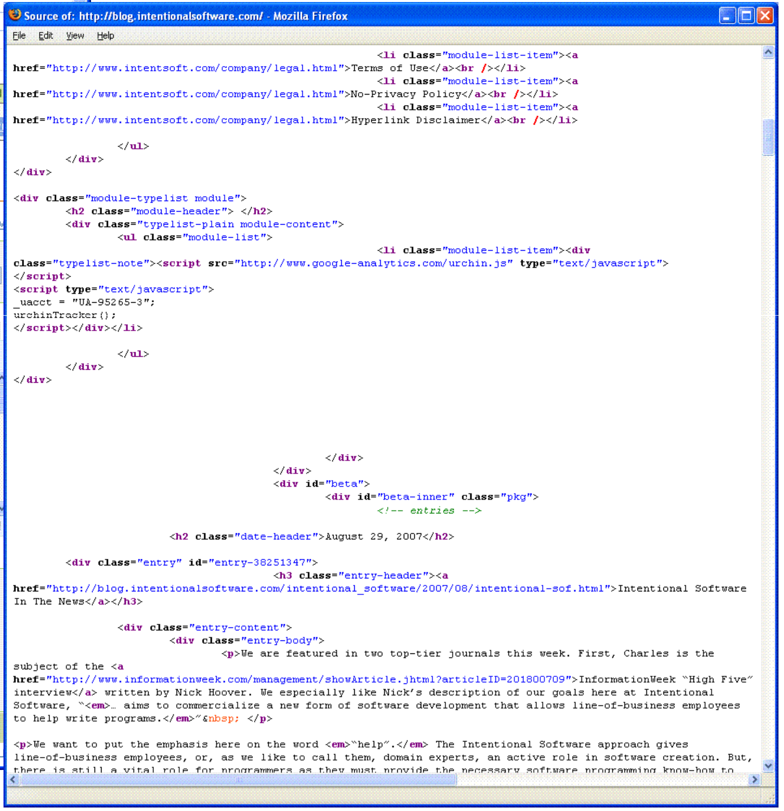
Comments: Open

Accept TrackBacks ✓

Send a TrackBack to these addresses

View Previously Sent TrackBacks
Preview
Save
Delete Post
Customize the display of this

As edited
(input)


Source of: http://blog.intentionalsoftware.com/ - Mozilla Firefox
File Edit View Help

```
                                        <li class="module-list-item"><a
href="http://www.intentsoft.com/company/legal.html">Terms of Use</a><br /></li>
                                        <li class="module-list-item"><a
href="http://www.intentsoft.com/company/legal.html">No-Privacy Policy</a><br /></li>
                                        <li class="module-list-item"><a
href="http://www.intentsoft.com/company/legal.html">Hyperlink Disclaimer</a><br /></li>

                <ul>
        </div>
</div>

<div class="module-typelist module">
        <h2 class="module-header"> </h2>
        <div class="typelist-plain module-content">
                <ul class="module-list">
                                        <li class="module-list-item"><div
class="typelist-note"><script src="http://www.google-analytics.com/urchin.js" type="text/javascript">
</script>
<script type="text/javascript">
_uacct = "UA-95265-3";
urchinTracker();
</script></div></li>

                </ul>
        </div>
</div>




                            </div>
                        </div>
                        <div id="beta">
                            <div id="beta-inner" class="pkg">
                                <!-- entries -->

                <h2 class="date-header">August 29, 2007</h2>

        <div class="entry" id="entry-38251347">
                        <h3 class="entry-header"><a
href="http://blog.intentionalsoftware.com/intentional_software/2007/08/intentional-sof.html">Intentional Software
In The News</a></h3>

                <div class="entry-content">
                        <div class="entry-body">
                                <p>We are featured in two top-tier journals this week. First, Charles is the
subject of the <a
href="http://www.informationweek.com/management/showArticle.jhtml?articleID=201800709">InformationWeek “High Five”
interview</a> written by Nick Hoover. We especially like Nick’s description of our goals here at Intentional
Software, “<em>… aims to commercialize a new form of software development that allows line-of-business employees
to help write programs.</em>”&nbsp; </p>

<p>We want to put the emphasis here on the word <em>“help”.</em> The Intentional Software approach gives
line-of-business employees, or, as we like to call them, domain experts, an active role in software creation. But,
there is still a vital role for programmers as they must provide the necessary software programming know-how to
```

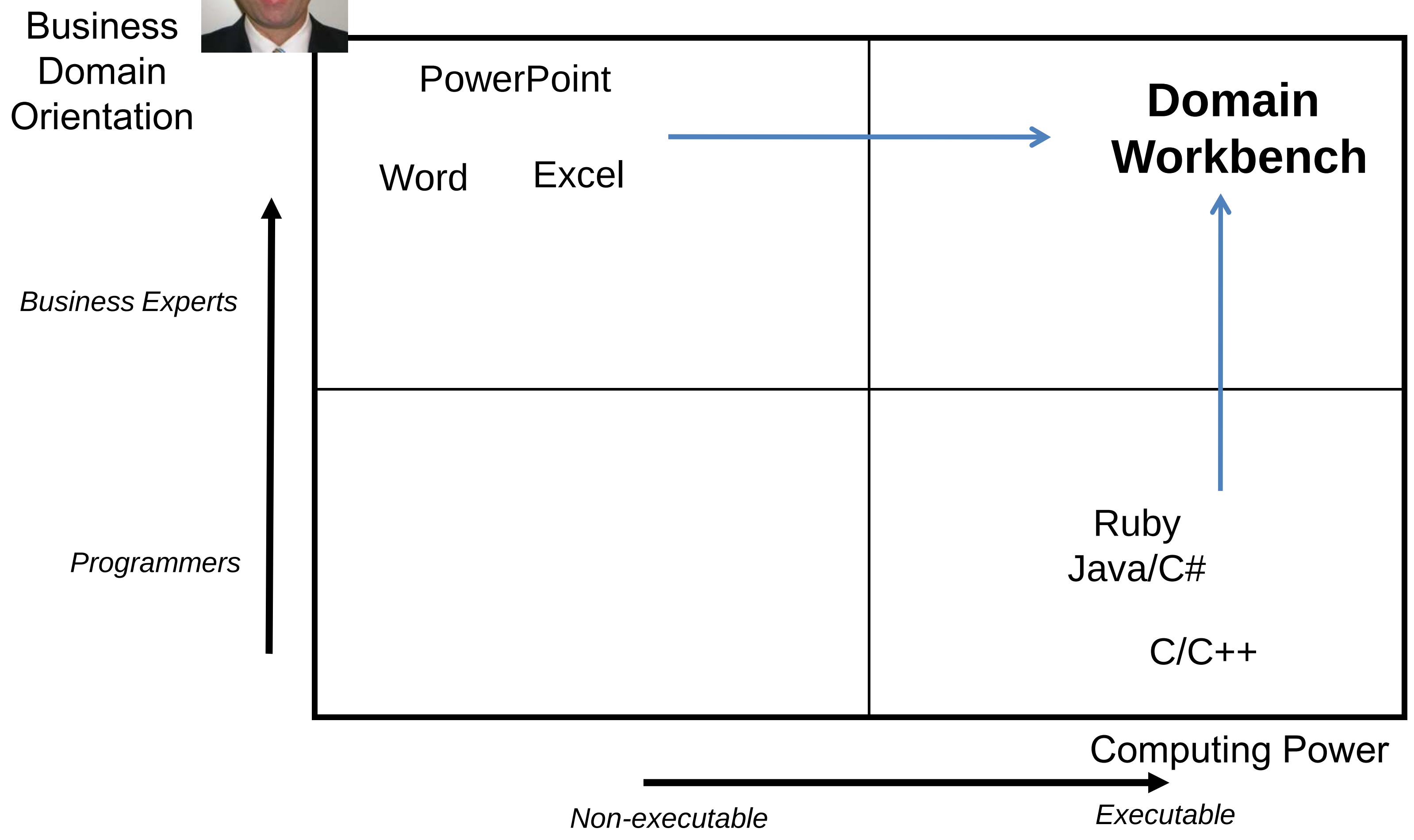
As generated
(output)

More ”Input + Process” Analogies

- DNA
  - Growing an organ, e.g. Optic nerve
  - Brevity of DNA makes evolution possible

- Kolmogorov complexity

Separating and Weaving Domains



Key Benefits

- Domain Expert participation feasible – domain knowledge isolated from technology
- Separation of concerns – complexity is reduced
- Programmers create a more valuable artifact: Generator –weaves domain input with Software Engineering knowledge

Domain Orientation Trends:

- Domain Specific Languages (DSL)
- Code Generation/Generative Programming (GP)
- Domain Specific Modeling (DSM)
- Domain Driven Design (DDD)
- Model Driven Development (MDD)
- Meta Programming
- …

What prevents DSL mainstream use?

- Integrate Domain Experts fully
  - Matching existing notations
  - Mixing graphical/textual notations
- Multi-domain
  - Compose independent domains
  - References between domains
- Domain evolution, domains must be able to evolve without limitations (structure and notation)
- Groupware for domain experts

Programming Languages as Base?

- Programming languages as the model leaves major issues:
  - Text-only not satisfactory
  - Parsing requirement constrains language design
  - Multi-domain is unaddressed
  - Domain evolution is unaddressed
  - Current groupware (CM) not feasible for domain experts

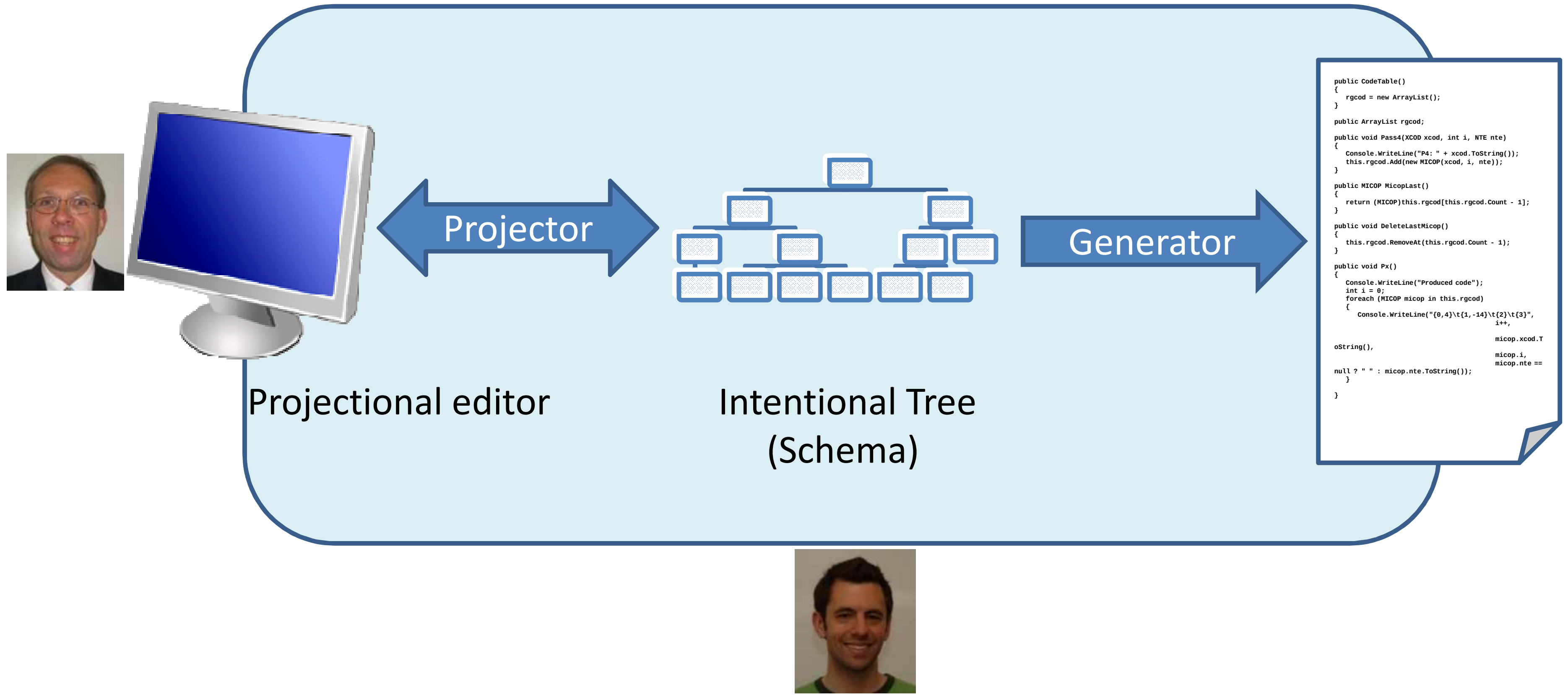
Intentional Domain Workbench

- Bring domain orientation to a new level by changing software creation to truly integrate Domain Experts

Def Domain Workbench (Martin Fowler)

1. Users can freely define new domains, including languages, that are fully integrated with each other.
2. The primary source of information is a persistent abstract representation.
3. Domain designers define domains in three main parts: schemas, editors, and generators.
4. Domain users manipulate a domain through a projectional editor.
5. A domain workbench can work with incomplete and contradictory information.

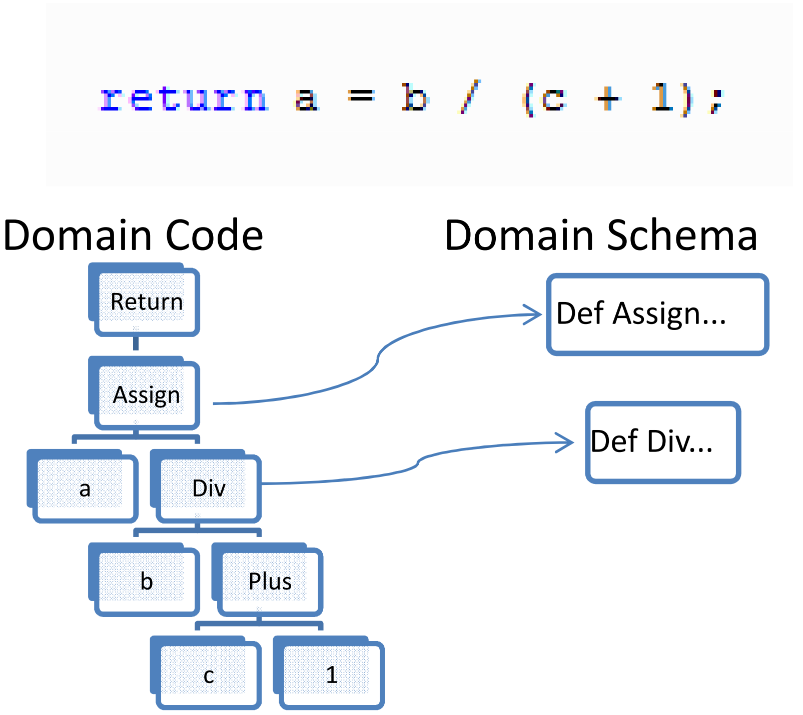
Inside the Domain Workbench



```
public CodeTable()
{
   rgcod = new ArrayList();
}

public ArrayList rgcod;

public void Pass4(XCOD xcod, int i, NTE nte)
{
   Console.WriteLine("P4: " + xcod.ToString());
   this.rgcod.Add(new MICOP(xcod, i, nte));
}

public MICOP MicopLast()
{
   return (MICOP)this.rgcod[this.rgcod.Count - 1];
}

public void DeleteLastMicop()
{
   this.rgcod.RemoveAt(this.rgcod.Count - 1);
}

public void Px()
{
   Console.WriteLine("Produced code");
   int i = 0;
   foreach (MICOP micop in this.rgcod)
   {
      Console.WriteLine("{0,4}\t{1,-14}\t{2}\t{3}",
                                          i++,

                                          micop.xcod.T
oString(),
                                          micop.i,
                                          micop.nte ==
null ? " " : micop.nte.ToString());
   }

}
```

Intentional Tree

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



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

Projectional editing

- Separates underlying representation from notation (syntax)
- Works in two directions: output and editing
- Special selections that take tree structure into account
- Large number of notations for:
  - matching existing notations
  - multi-domain
  - ambiguity resolving
  - domain evolution
- Can also edit Programs, Schema, Generators

Some Notational Examples

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

or

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

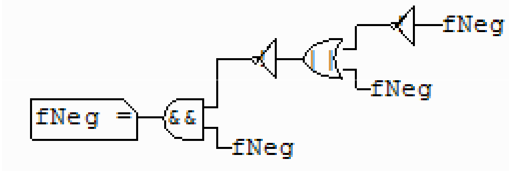
```
pi = fNeg ? pi - 4 / j : pi + 4 / j;
```

or

$$\texttt{pi} = \begin{cases} \texttt{pi} - \dfrac{4}{\texttt{j}} & \texttt{if fNeg} \\ \texttt{pi} + \dfrac{4}{\texttt{j}} & \texttt{otherwise} \end{cases}$$

```
fNeg = fNeg && !(fNeg || !fNeg);
```

or



$$\texttt{time\_track\_enter} = \texttt{t+}\int \frac{\Delta D}{airline\_flight : aircraft : speed\_air}$$

```
method vector_velocity(aircraft, earth)
    {
```

$$v_{north} = aircraft : speed * \cos(aircraft : fpa) * \cos(aircraft : ha)$$
$$v_{east} = aircraft : speed * \cos(aircraft : fpa) * \sin(aircraft : ha)$$
$$v_{vertcl} = aircraft : speed * \sin(aircraft : fpa)$$

```
        cla = cos(aircraft:latitude)
        sla = sin(aircraft:latitude)
        clo = cos(aircraft:longitude)
        slo = sin(aircraft:longitude)
```

$$\begin{pmatrix} v_{x\_pos} \\ v_{y\_pos} \\ v_{z\_pos} \end{pmatrix} = \begin{bmatrix} cla & 0 & -sla \\ -sla*clo & -slo & -cla*clo \\ -sla*slo & clo & -cla*slo \end{bmatrix} \begin{pmatrix} v_{north} \\ v_{east} \\ -v_{vertcl} \end{pmatrix}$$

$$= \begin{bmatrix} cla & 0 & sla \\ -sla*clo & -slo & cla*clo \\ -sla*slo & clo & cla*slo \end{bmatrix} \begin{pmatrix} v_{north} \\ v_{east} \\ v_{vertcl} \end{pmatrix}$$

```
    }    end method
```

Integrate Domain Experts

- Matching existing notations
- Mixing notation graphics / text

- 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

Multi domain

- Compose independent domains
- References between domains

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

Domain evolution

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

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, open, update, commit, merge

Intentional Domain Workbench Status

- Technology is fully capable of handling the Domain Workbench requirements.
- Nearing operational use in selected domains
- Working with selected customers only, for example with Capgemini.

Capgemini

CONSULTING.TECHNOLOGY.OUTSOURCING

Henk Kolk

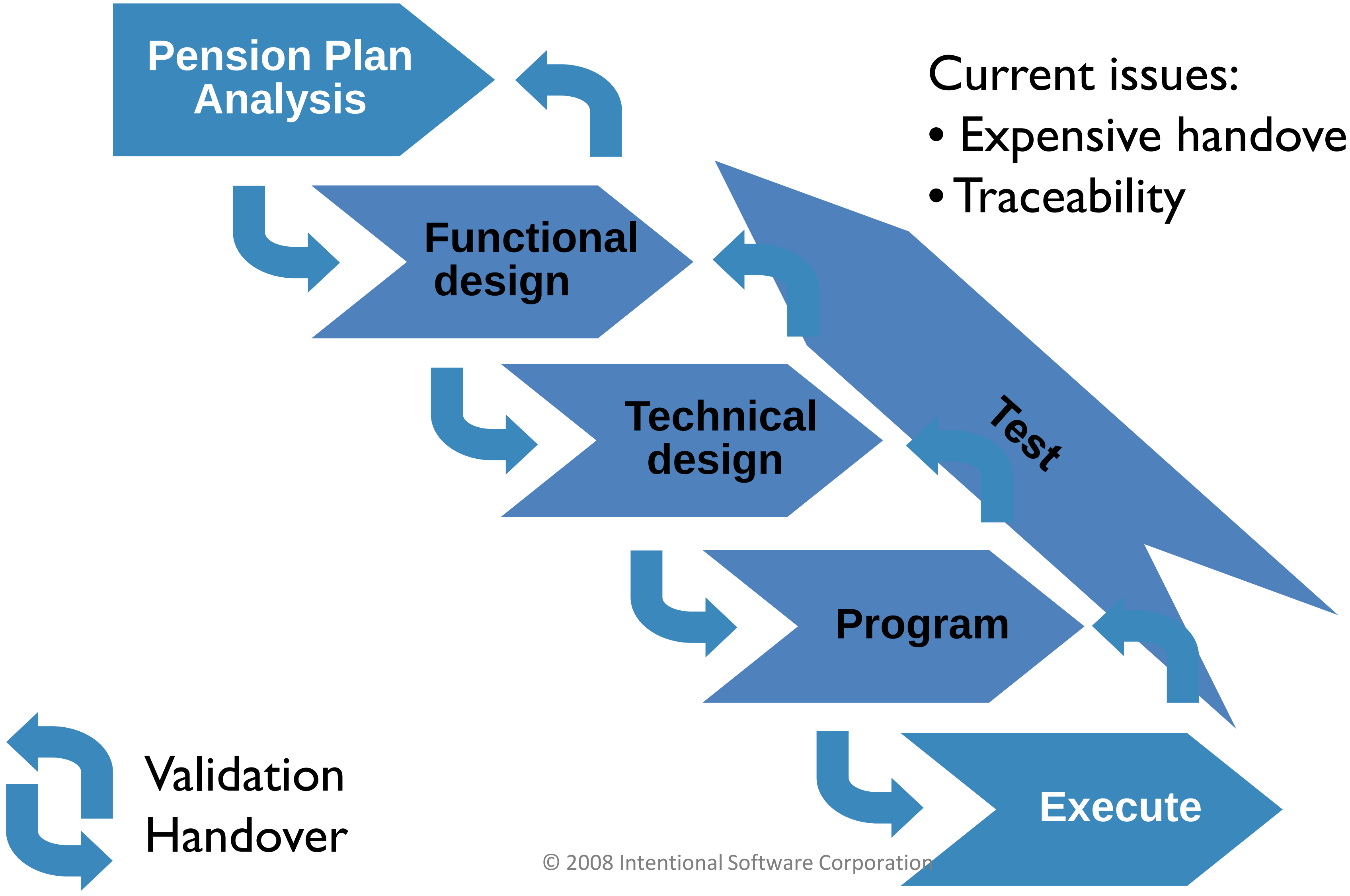
CTO Financial Services Capgemini

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

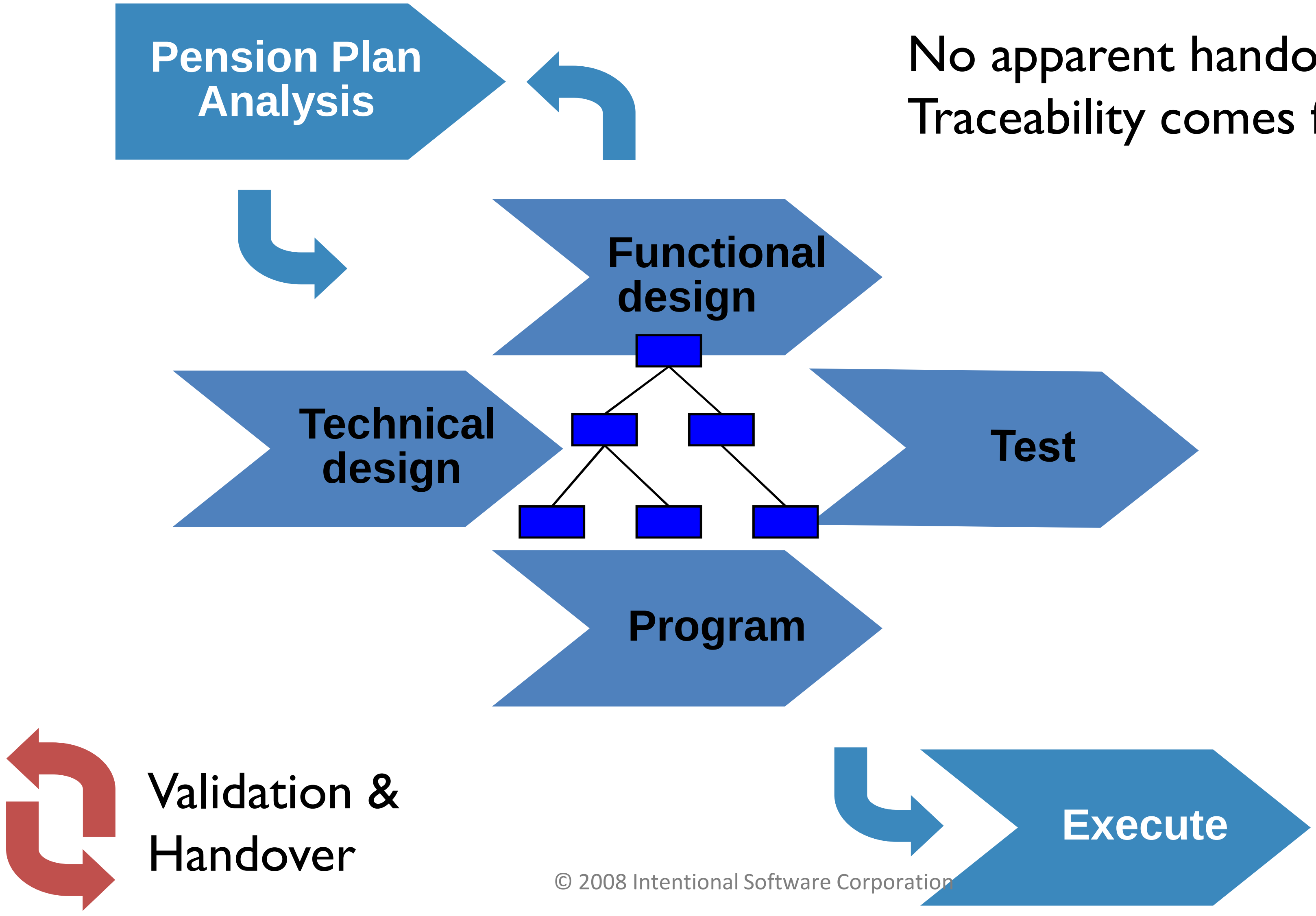
Old way: disconnected domains



Current issues:

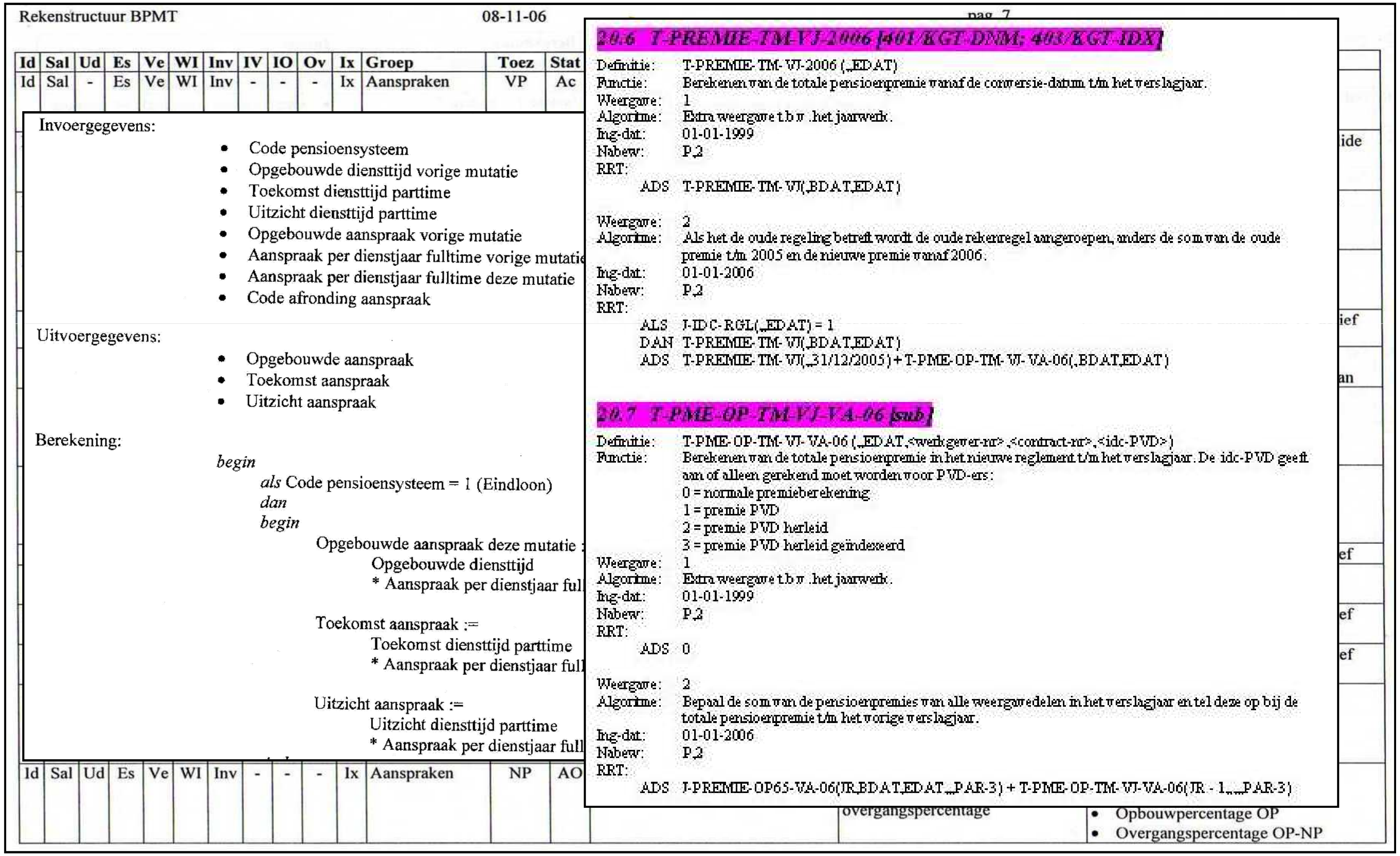
- Expensive handovers
- Traceability

New way: connected domains



No apparent handover
Traceability comes for free

Old way: Excel & Word



Rekenstructuur BPMT 08-11-06 pag. 7

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 full

        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

ide

ief

an

ef

ef

ef

overgangspercentage

- Opbouwpercentage OP
- Overgangspercentage OP-NP

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  J-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)
```

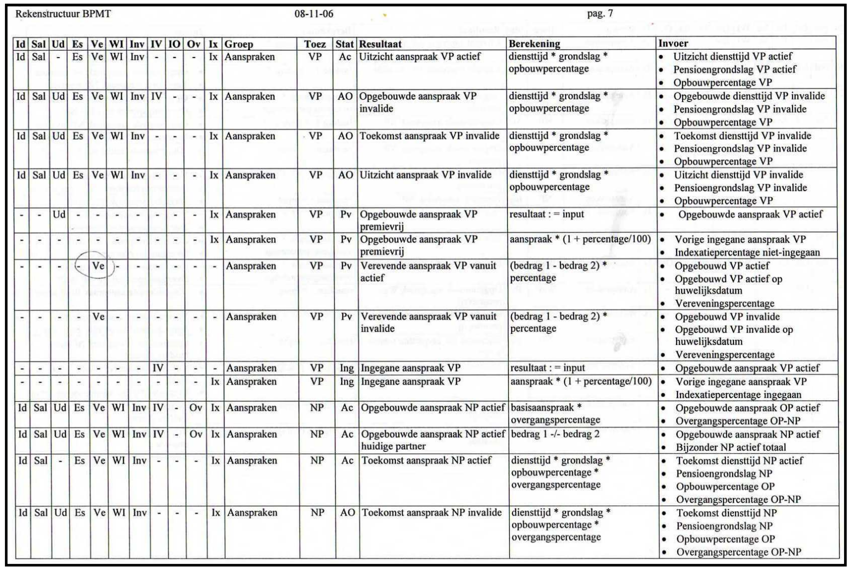
20.7 T-PME-OP-TM-VJ-VA-06 [sub]

```
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  J-PREMIE-OP65-VA-06(JR,BDAT,EDAT,,,PAR-3) + T-PME-OP-TM-VJ-VA-06(JR - 1,,,,PAR-3)
```

New way: Pension Workbench

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

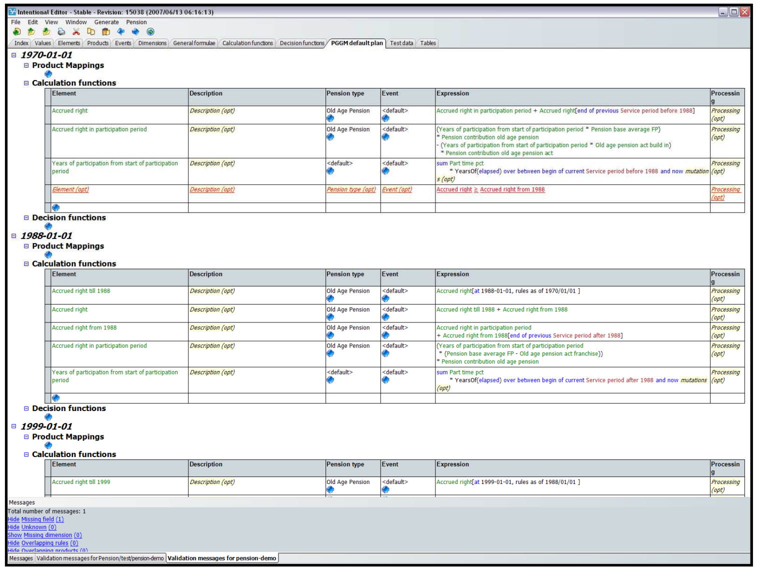
Old spreadsheet



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 huidige partner	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


Intentional Editor - Stable - Revision: 15038 (2007/06/13 06:16:13)
File Edit View Window Generate Pension
Index Values Elements Products Events Dimensions General formulae Calculation functions Decision functions PGGM default plan Test data Tables

1970-01-01

Product Mappings

Calculation functions

Element	Description	Pension type	Event	Expression	Processing
Accrued right	Description (opt)	Old Age Pension	<default>	Accrued right in participation period + Accrued right[end of previous Service period before 1988]	Processing (opt)
Accrued right in participation period	Description (opt)	Old Age Pension	<default>	(Years of participation from start of participation period * Pension base average FP) * Pension contribution old age pension - (Years of participation from start of participation period * Old age pension act build in) * Pension contribution old age pension act	Processing (opt)
Years of participation from start of participation period	Description (opt)	<default>	<default>	sum Part time pct * YearsOf(elapsed) over between begin of current Service period before 1988 and now mutations (opt)	Processing (opt)
Element (opt)	Description (opt)	Pension type (opt)	Event (opt)	Accrued right ≥ Accrued right from 1988	Processing (opt)

Decision functions

1988-01-01

Product Mappings

Calculation functions

Element	Description	Pension type	Event	Expression	Processing
Accrued right till 1988	Description (opt)	Old Age Pension	<default>	Accrued right[at 1988-01-01, rules as of 1970/01/01]	Processing (opt)
Accrued right	Description (opt)	Old Age Pension	<default>	Accrued right till 1988 + Accrued right from 1988	Processing (opt)
Accrued right from 1988	Description (opt)	Old Age Pension	<default>	Accrued right in participation period + Accrued right from 1988[end of previous Service period after 1988]	Processing (opt)
Accrued right in participation period	Description (opt)	Old Age Pension	<default>	(Years of participation from start of participation period * (Pension base average FP - Old age pension act franchise)) * Pension contribution old age pension	Processing (opt)
Years of participation from start of participation period	Description (opt)	<default>	<default>	sum Part time pct * YearsOf(elapsed) over between begin of current Service period after 1988 and now mutations (opt)	Processing (opt)

Decision functions

1999-01-01

Product Mappings

Calculation functions

Element	Description	Pension type	Event	Expression	Processing
Accrued right till 1999	Description (opt)	Old Age Pension	<default>	Accrued right[at 1999-01-01, rules as of 1988/01/01]	Processing (opt)

Messages

Total number of messages: 1

Hide Missing field (1)

Hide Unknown (0)

Show Missing dimension (0)

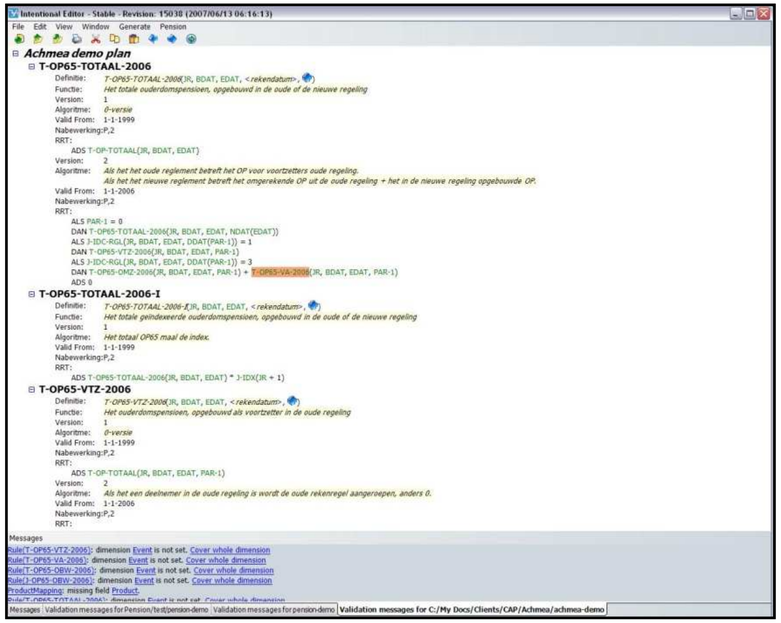
Hide Overlapping rules (0)

Hide Overlapping products (0)

Messages
Validation messages for Pension/test/pension-demo
Validation messages for pension-demo

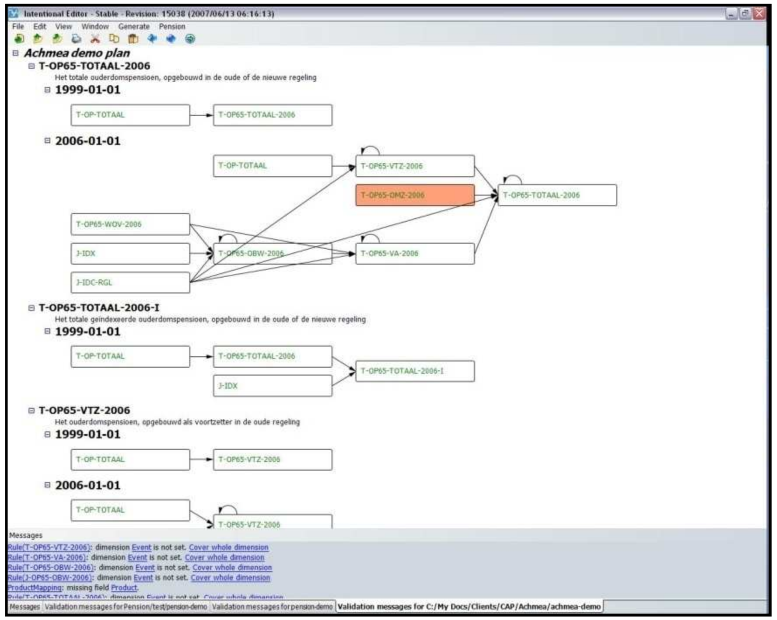
Multiple Views with Graphics

Pension Plan versions



```
Intentional Editor - Stable - Revision: 15038 (2007/06/13 06:16:13)
File  Edit  View  Window  Generate  Pension
⊟ Achmea demo plan
  ⊟ T-OP65-TOTAAL-2006
      Definitie:     T-OP65-TOTAAL-2006(JR, BDAT, EDAT, <rekendatum>, )
      Functie:       Het totale ouderdomspensioen, opgebouwd in de oude of de nieuwe regeling
      Version:       1
      Algoritme:     0-versie
      Valid From:    1-1-1999
      Nabewerking:P,2
      RRT:
         ADS T-OP-TOTAAL(JR, BDAT, EDAT)
      Version:       2
      Algoritme:     Als het het oude reglement betreft het OP voor voortzetters oude regeling.
                     Als het het nieuwe reglement betreft het omgerekende OP uit de oude regeling + het in de nieuwe regeling opgebouwde OP.
      Valid From:    1-1-2006
      Nabewerking:P,2
      RRT:
         ALS PAR-1 = 0
         DAN T-OP65-TOTAAL-2006(JR, BDAT, EDAT, NDAT(EDAT))
         ALS J-IDC-RGL(JR, BDAT, EDAT, DDAT(PAR-1)) = 1
         DAN T-OP65-VTZ-2006(JR, BDAT, EDAT, PAR-1)
         ALS J-IDC-RGL(JR, BDAT, EDAT, DDAT(PAR-1)) = 3
         DAN T-OP65-OMZ-2006(JR, BDAT, EDAT, PAR-1) + T-OP65-VA-2006(JR, BDAT, EDAT, PAR-1)
         ADS 0
  ⊟ T-OP65-TOTAAL-2006-I
      Definitie:     T-OP65-TOTAAL-2006-I(JR, BDAT, EDAT, <rekendatum>, )
      Functie:       Het totale geindexeerde ouderdomspensioen, opgebouwd in de oude of de nieuwe regeling
      Version:       1
      Algoritme:     Het totaal OP65 maal de index.
      Valid From:    1-1-1999
      Nabewerking:P,2
      RRT:
         ADS T-OP65-TOTAAL-2006(JR, BDAT, EDAT) * J-IDX(JR + 1)
  ⊟ T-OP65-VTZ-2006
      Definitie:     T-OP65-VTZ-2006(JR, BDAT, EDAT, <rekendatum>, )
      Functie:       Het ouderdomspensioen, opgebouwd als voortzetter in de oude regeling
      Version:       1
      Algoritme:     0-versie
      Valid From:    1-1-1999
      Nabewerking:P,2
      RRT:
         ADS T-OP-TOTAAL(JR, BDAT, EDAT, PAR-1)
      Version:       2
      Algoritme:     Als het een deelnemer in de oude regeling is wordt de oude rekenregel aangeroepen, anders 0.
      Valid From:    1-1-2006
      Nabewerking:P,2
      RRT:
Messages
Rule(T-OP65-VTZ-2006): dimension Event is not set. Cover whole dimension
Rule(T-OP65-VA-2006): dimension Event is not set. Cover whole dimension
Rule(T-OP65-OBW-2006): dimension Event is not set. Cover whole dimension
Rule(J-OP65-OBW-2006): dimension Event is not set. Cover whole dimension
ProductMapping: missing field Product.
Rule(T-OP65-TOTAAL-2006): dimension Event is not set. Cover whole dimension
Messages | Validation messages for Pension/test/pension-demo | Validation messages for pension-demo | Validation messages for C:/My Docs/Clients/CAP/Achmea/achmea-demo
```

Rule dependencies



Intentional Editor - Stable - Revision: 15038 (2007/06/13 06:16:13)

⊟ Achmea demo plan

⊟ T-OP65-TOTAAL-2006

Het totale ouderdomspensioen, opgebouwd in de oude of de nieuwe regeling

⊟ 1999-01-01

⊟ 2006-01-01

⊟ T-OP65-TOTAAL-2006-I

Het totale geindexeerde ouderdomspensioen, opgebouwd in de oude of de nieuwe regeling

⊟ 1999-01-01

⊟ T-OP65-VTZ-2006

Het ouderdomspensioen, opgebouwd als voortzetter in de oude regeling

⊟ 1999-01-01

⊟ 2006-01-01

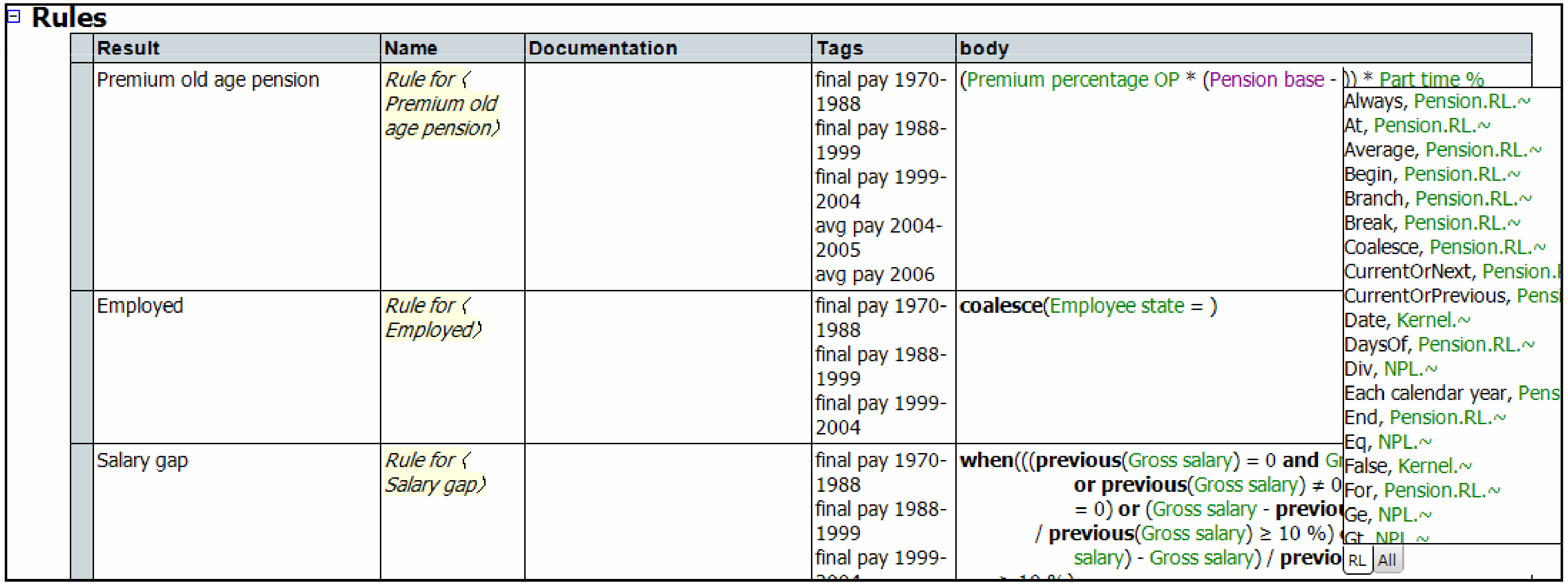
Messages

Rule(T-OP65-VTZ-2006): dimension Event is not set. Cover whole dimension
Rule(T-OP65-VA-2006): dimension Event is not set. Cover whole dimension
Rule(T-OP65-OBW-2006): dimension Event is not set. Cover whole dimension
Rule(J-OP65-OBW-2006): dimension Event is not set. Cover whole dimension
ProductMapping: missing field Product.
Rule(T-OP65-TOTAAL-2006): dimension Event is not set. Cover whole dimension

Messages | Validation messages for Pension/test/pension-demo | Validation messages for pension-demo | Validation messages for C:/My Docs/Clients/CAP/Achmea/achmea-demo

Compose Business Domain

- Domain Schema
- Projectional Editors

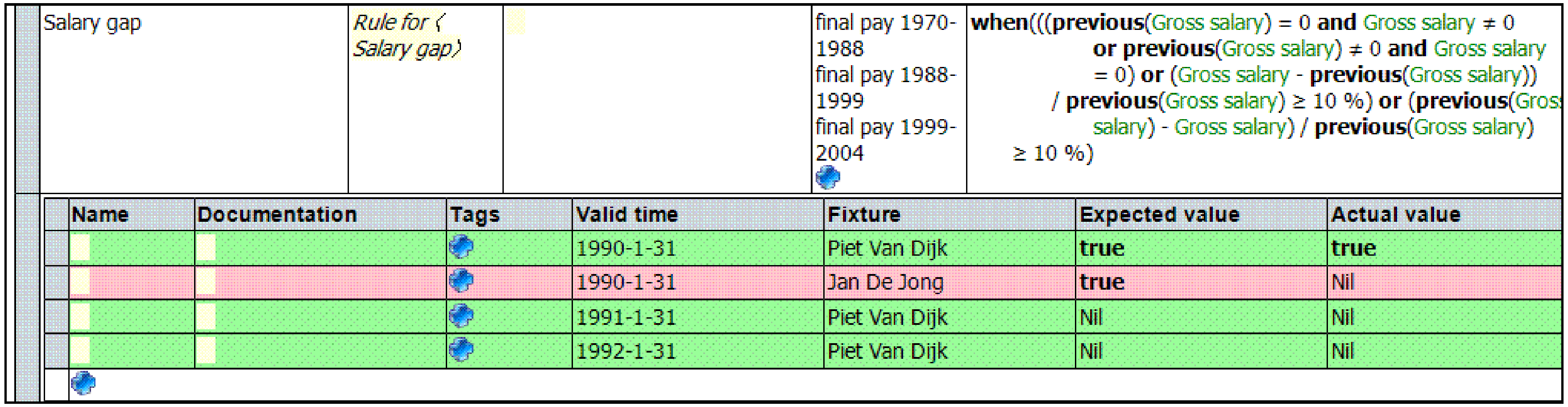


Rules

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 -)) * Part time %
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** Gr **or previous**(Gross salary) ≠ 0 = 0) **or** (Gross salary - **previo** / **previous**(Gross salary) ≥ 10 %) salary) - Gross salary) / **previo**

Integrate Rule Test Domain

- Unit Tests for pension rules
- Real time evaluation



Salary gap

Rule for ⟨Salary gap⟩

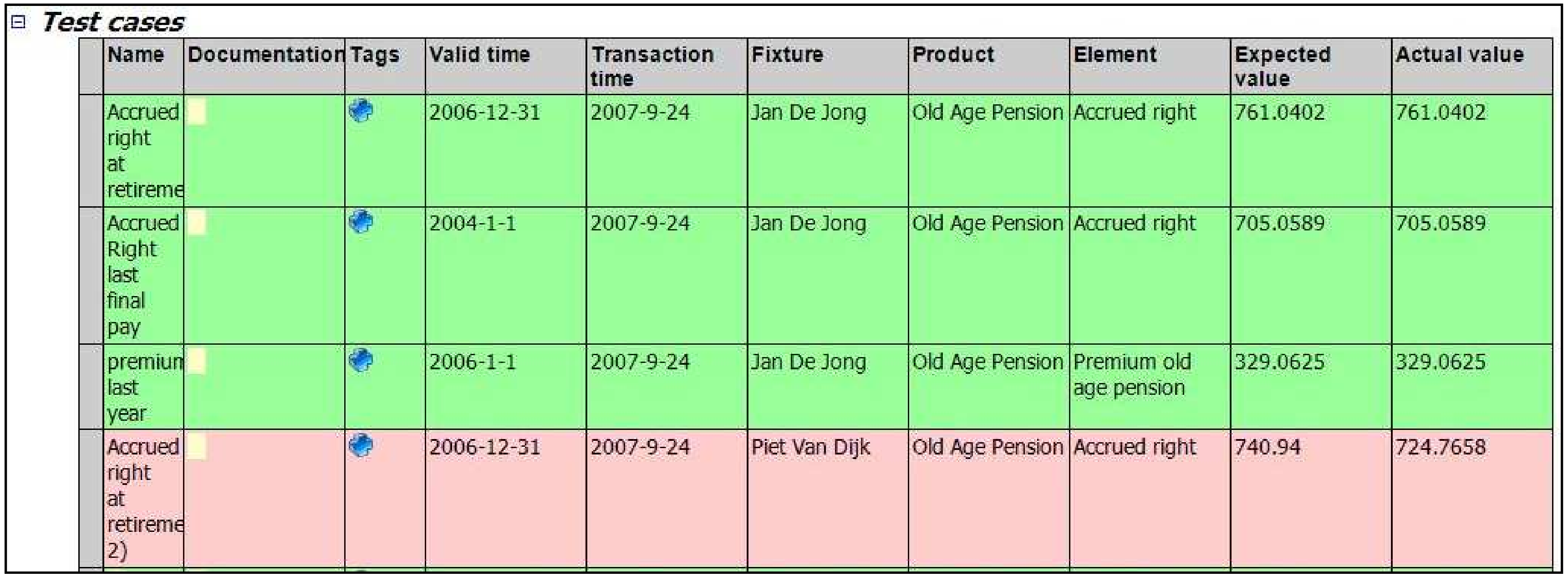
final pay 1970-1988
final pay 1988-1999
final pay 1999-2004

```
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(Gros
      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

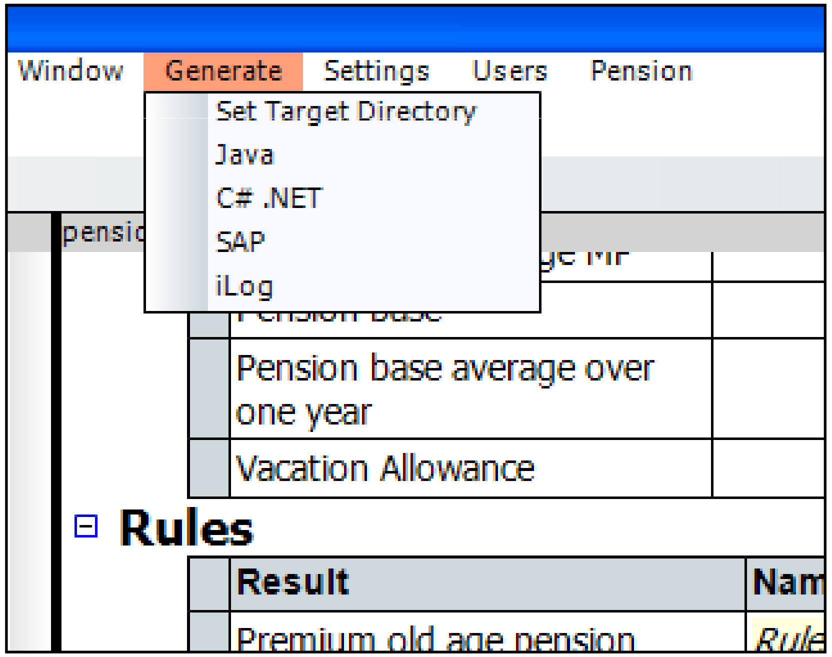


⊟ ***Test cases***

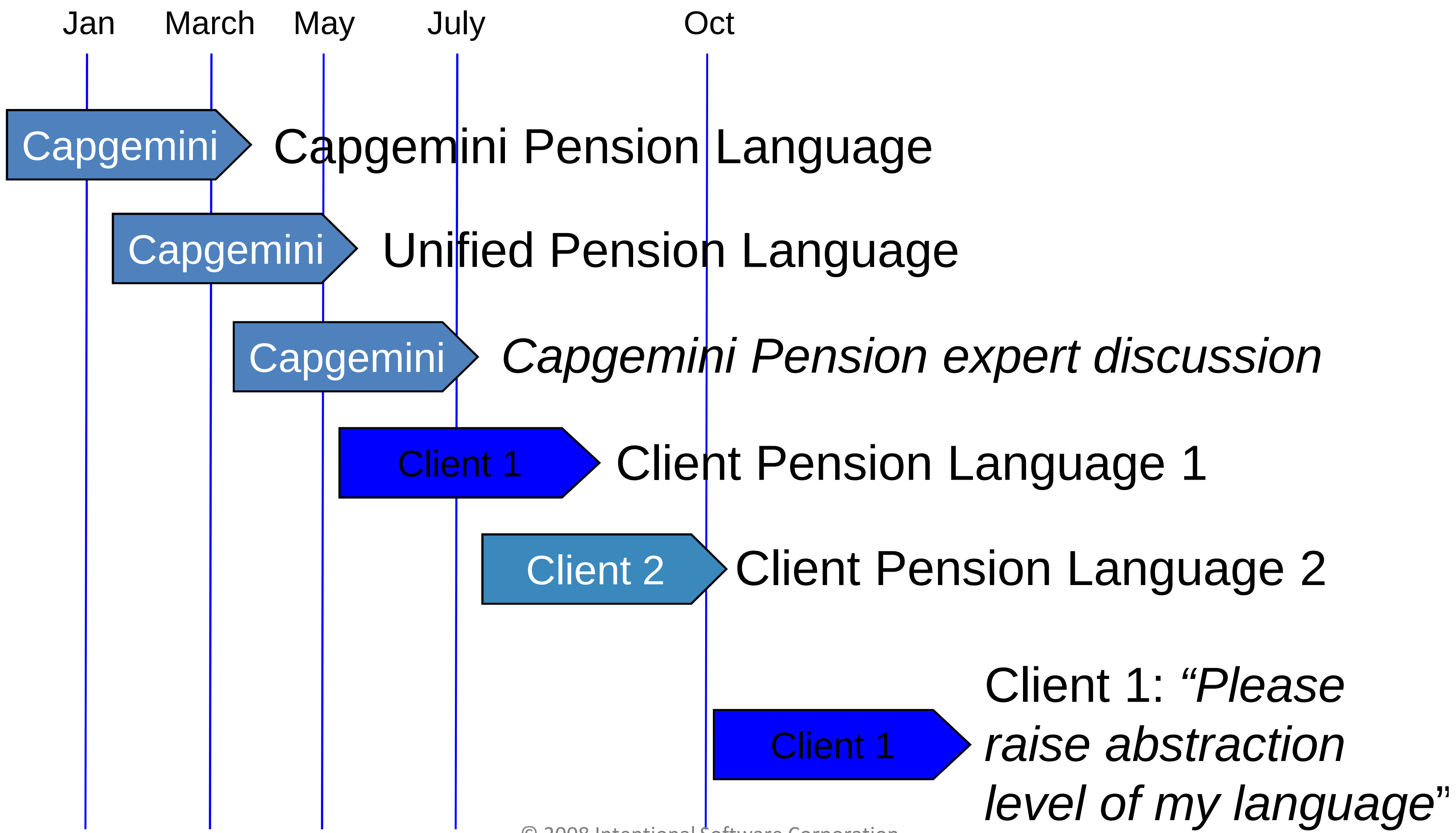
Name	Documentation	Tags	Valid time	Transaction time	Fixture	Product	Element	Expected value	Actual value
Accrued right at retireme			2006-12-31	2007-9-24	Jan De Jong	Old Age Pension	Accrued right	761.0402	761.0402
Accrued Right last final pay			2004-1-1	2007-9-24	Jan De Jong	Old Age Pension	Accrued right	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

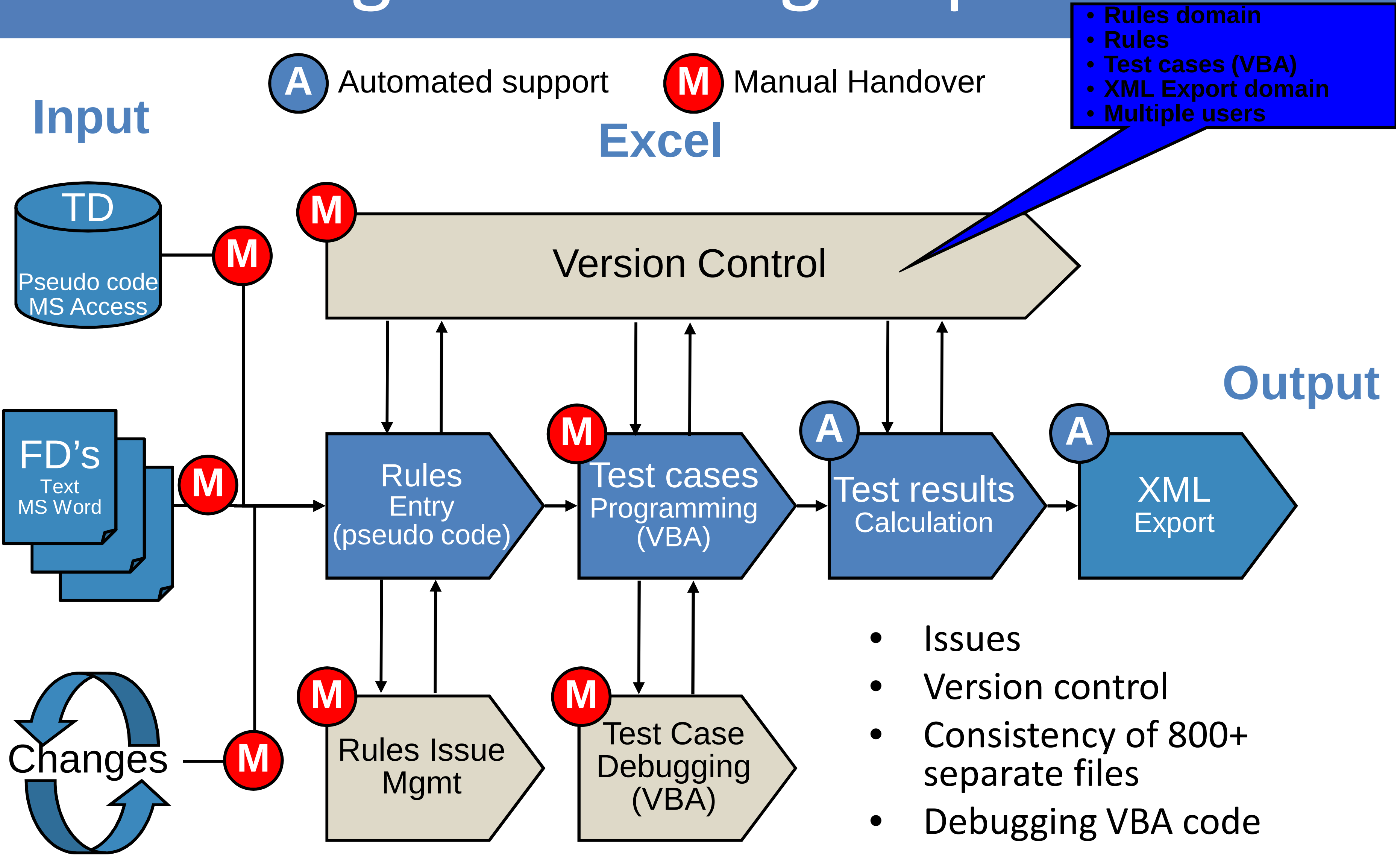
- Multiple implementation target languages



Domain Language Evolution

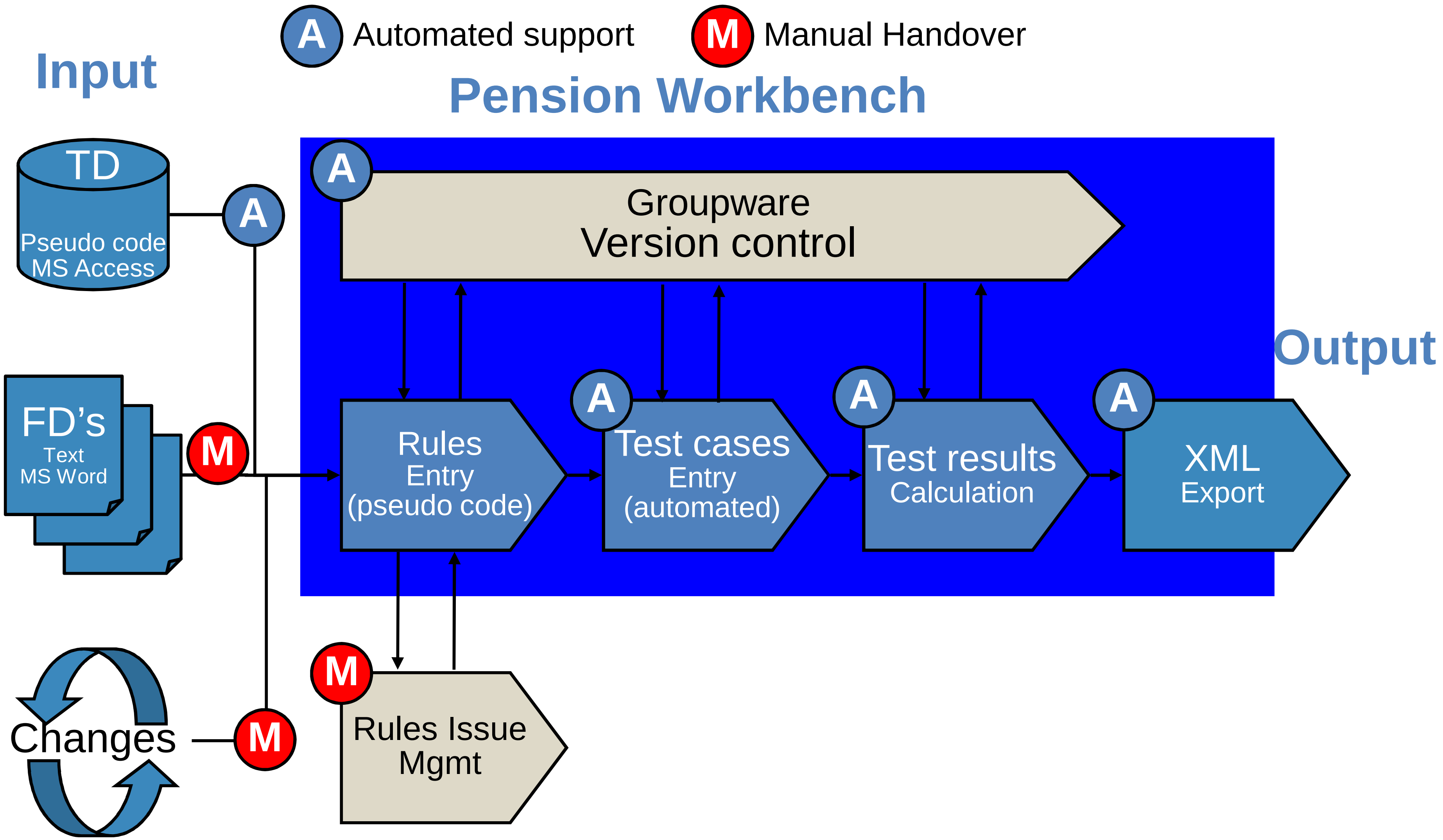

Jan
March
May
July
Oct
Capgemini
Capgemini Pension Language
Capgemini
Unified Pension Language
Capgemini
Capgemini Pension expert discussion
Client 1
Client Pension Language 1
Client 2
Client Pension Language 2
Client 1
Client 1: *“Please raise abstraction level of my language*”

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