RE: A report from the trenches

drs. Eric D. Schabell
System Specialist

04 November 2008

About me

- Born in the USA (Oregon), traveled after school, 1987-1992.
- In NL since 1992.
- IBM Netherlands, 1998.
- VU, Informatica 2002.
- SW engineering Escador/Aramiska, 2000-2001.
- RUN, WP IRIS, 2001 – 2007.
- SNS Bank IT, Sep 2007 - present.

Agenda

- **A perfect world**
- Reality intrudes
- Real life example: STP Aanvragen
  - Introduction
  - Life in the trenches
  - Example: web service
  - Example: jBPM process
- Building on Legacy: STP Contractrekeningen (Deposito's)
  - Use Cases and legacy problems
- Towards the future: STP Betalen
  - Use Cases, getting better?

Up to now…

The world is a pretty place:

- Business rules
- Use cases
- Requirements
- Workshops
- etc…

How we like(d) to work

Overview of the work flow (iterations):

- Biz workshops (IA)
- Functional Designs (FO), use cases.

- Use cases, more detailed (requirements).
- Technical Designs (TO)
- Unit tests + code

Agenda

- A perfect world
- **Reality intrudes**
- Real life example: STP Aanvragen
  - Introduction
  - Life in the trenches
  - Example: web service
  - Example: jBPM process
- Building on Legacy: STP Contractrekeningen (Deposito's)
  - Use Cases and legacy problems
- Towards the future: STP Betalen
  - Use Cases, getting better?

Reality is a wake up call

Sometimes there are other factors:

- Business needs
- Market changes
- Lack of personnel (quality / quantity)
- Technical challenges
- Planning problems

Agenda

- A perfect world
- Reality intrudes
- **Real life example: STP Aanvragen**
  - Introduction
  - Life in the trenches
  - Example: web service
  - Example: jBPM process
- Building on Legacy: STP Contractrekeningen (Deposito's)
  - Use Cases and legacy problems
- Towards the future: STP Betalen
  - Use Cases, getting better?

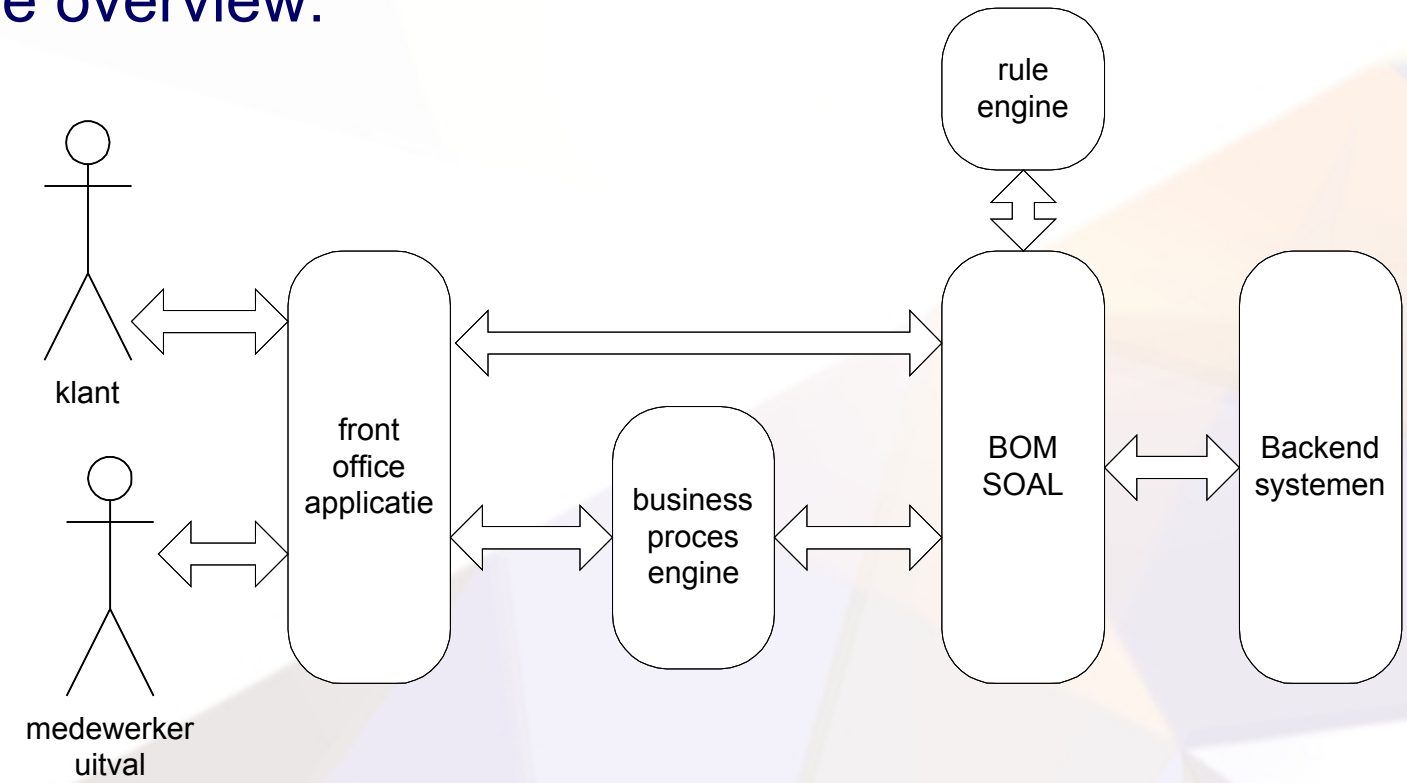
All bets are off!

Can you be flexible in this environment?

- Biz workshops (4x)
- FO is 92 page PowerPoint presentation
- Use cases done by developers
- Coding (minimal unit tests, no mock web services framework)
- TO done after or iterative during coding.

Architecture

Simple overview:



What is going on: technical?

A very fast time to market:

- Challenging technical environment
  - jBPM
  - Web services
  - Back-ends: KIS and SAS mainframe
  - Uitval application

What is going on: business?

A very fast time to market:

- Four new savings products
  - Via internet channel
  - Request processing max 2 days (STP degree)
  - Straight Through Processing (STP Aanvragen)
  - Customer insisting on unrealistic deadlines

What is going on: process?

A very fast time to market:

- All banking issues and requirements remain
  - Privacy
  - Laws
  - Security
  - Validation
  - Data access / manipulation

Life in the trenches

In the trenches:

- Receive a task to implement some functionality
- Check FO (remember == ppt)
- Extract use cases for task (discuss with IA)
- Create TO
- Write unit tests
- Code implementation
- Test it!

Example: web service

Postcode validation:

- Receive a task, postcode validation
- Check FO (remember == ppt), slide 24
- Extract use cases for task (discuss with IA), see example
- Create TO, see SIC
- Write unit tests, see SIC
- Code implementation
- Test it!

Functional Design (FO): slide 24

SNS XXXXXXXXXX
Verplichte tegenrekening
Internetbankieren of TIN code of softlogin
SNS XXXXXXXXXXX
Verplichte tegenrekening
Internetbankieren of TIN code of softlogin
SNS XXXXXXXXXX
Verplichte tegenrekening
Internetbankieren of TIN code of softlogin
SNS XXXXXXXXXXXX
Verplichte tegenrekening
Internetbankieren of TIN code of softlogin

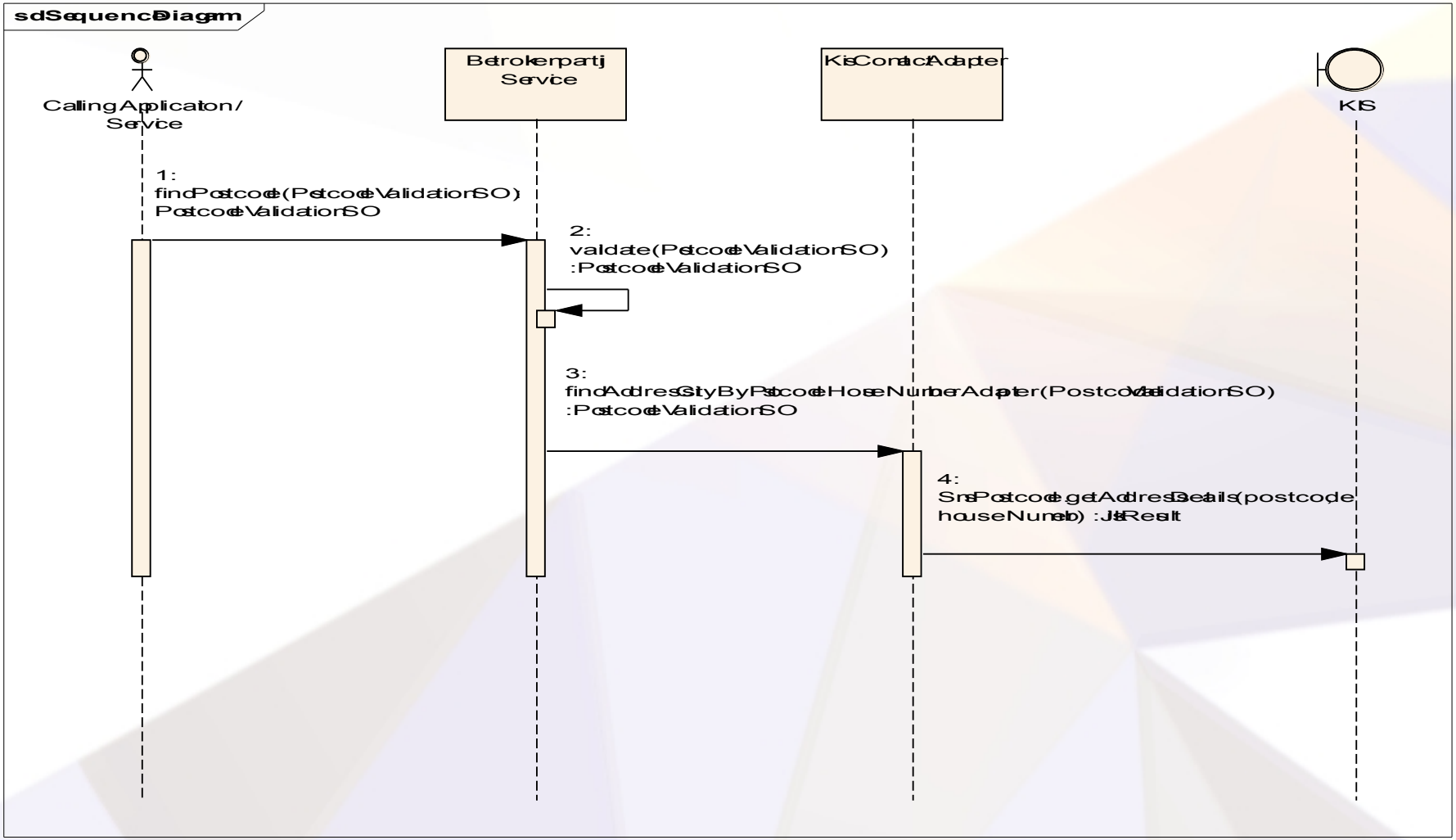
..en de rest….

- **postcodevalidatie: als postcode niet in onze tabel voorkomt dan uitval.**

«actor,entity»

Service interface contract

Sequence diagram:



Java anyone?

```
/** {@inheritDoc} */
  public PostcodeValidationSO findPostcode(PostcodeValidationSO postcodeSO) throws BetrokkenPartijServiceException {
     PostcodeValidationSO returnPostcodeValidationSO = new PostcodeValidationSO();
     logStart(postcodeSO);

     try {
        validate(postcodeSO);
     } catch (ServiceValidationException e) {
         throw createBetrokkenPartijServiceException(e, Originator.SERVICES);
     }

     try {
         // call to KIS for postcode lookup.
         returnPostcodeValidationSO = KISContactAdapter.getInstance().findAddressCityByPostcodeHouseNumber(postcodeSO);

         if (returnPostcodeValidationSO == null || StringUtils.isEmpty(returnPostcodeValidationSO.address)
             || StringUtils.isEmpty(returnPostcodeValidationSO.city)) {

                 // found invalid postcode, setup for returning an empty SO.
                 returnPostcodeValidationSO.address = "";
                 returnPostcodeValidationSO.city = "";
        }
     } catch (ServiceException e) {
        throw createBetrokkenPartijServiceException(e, Originator.KIS);
     } finally {
        // Do some logging at the end of the call.
        logEnd(returnPostcodeValidationSO);
     }

     return returnPostcodeValidationSO;
  }
```

Unit tests

```
/**
 * Test for an invalid postcode.
 * Note: there is no backend mock for services, thus is integration test.
 */
public void testFindPostcodeInvalid() {

    // fill our invalid postcode service request object.
    postcodeSO.postcode    = invalidPostcode;
    postcodeSO.houseNumber = invalidHouseNumber;

    try {
        PostcodeValidationSO so = bp.findPostcode(postcodeSO);

        // should be empty SO attributes.
        assertEquals(so.address, "");
        assertEquals(so.city, "");
    } catch (ServiceException e) {
        e.printStackTrace();
    }
}
```

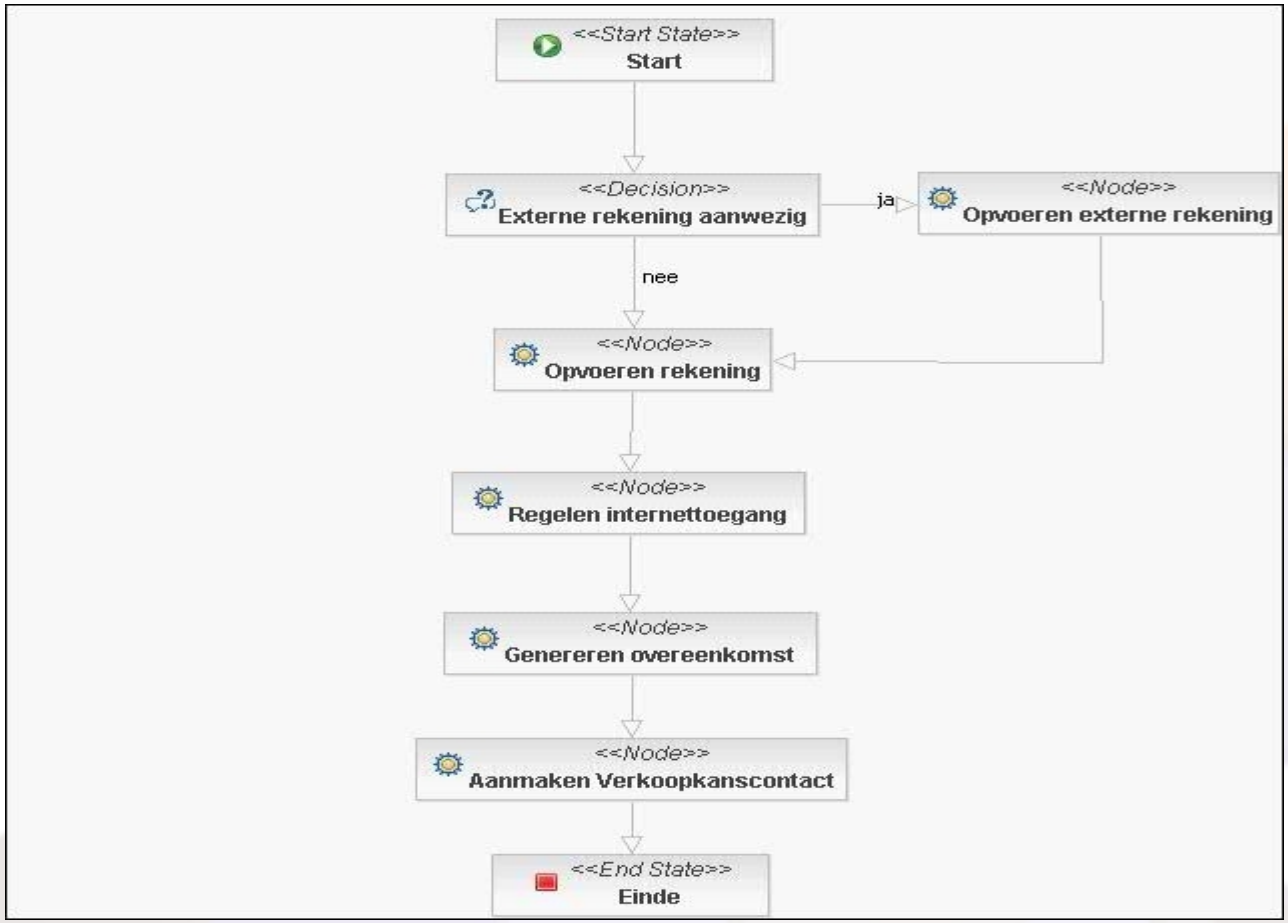
Example: jBPM process

Aanmaak product:

- Receive a task
- Check FO (remember == ppt), outside scope
- Extract use cases for task (discuss with IA), outside scope
- **Create TO, receive from lead developer, iterative updates**
- Write unit tests, ha ha, that's a good one!
- **Code implementation**
- **Test it, soapui**

Aanmaak Product (jBPM)

Process flow:



Agenda

- A perfect world
- Reality intrudes
- Real life example: STP Aanvragen
  - Introduction
  - Life in the trenches
  - Example: web service
  - Example: jBPM process
- **Building on Legacy: STP Contractrekeningen (Deposito's)**
  - Use Cases and legacy problems
- Towards the future: STP Betalen
  - Use Cases, getting better?

How we like to work

Overview of the work flow (iterations):

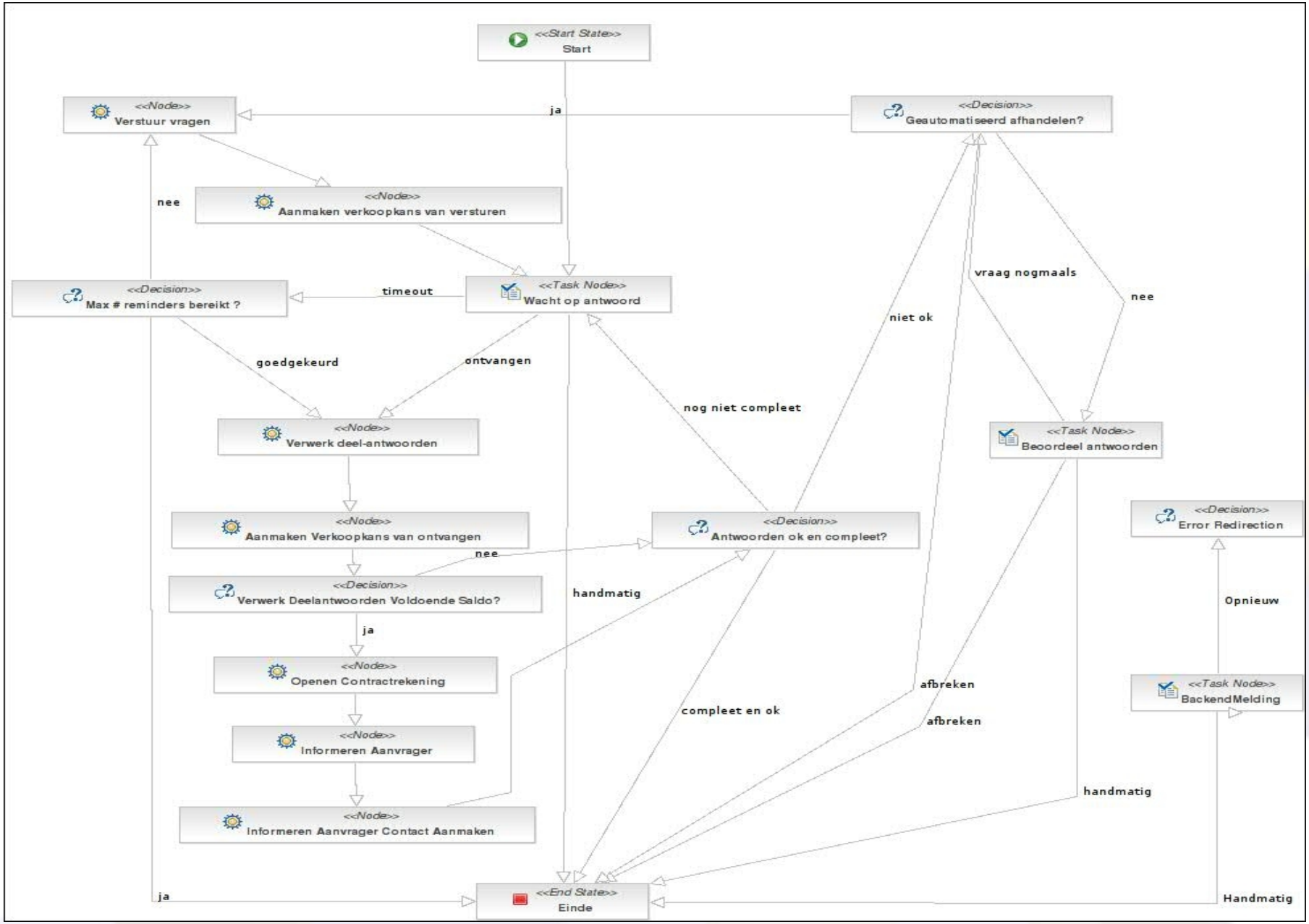
- Biz workshops (IA)
- Functional Designs (FO), set UC's and UCM.

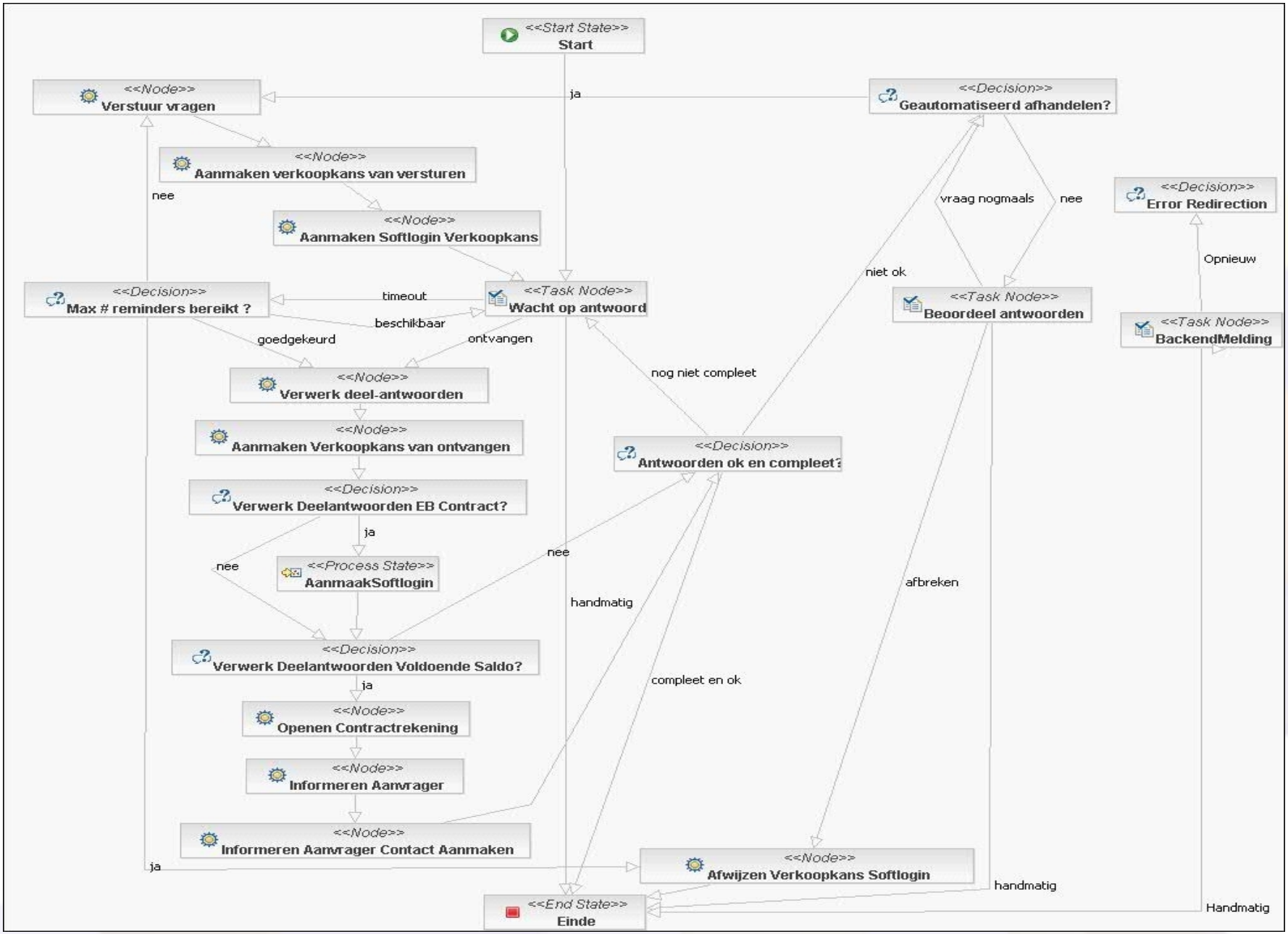
- SW Arch Document (SAD)
- Use Case Realizations, more detailed (requirements).
- Technical Design (SIC, components)
- Unit tests + code

STP CR UC's

Some facts:

- Actually have UC's!
- Very technical
- Never (still to this day) have seen the non-concept versions!
- Legacy process decisions haunt implementation:
  - Pollers/schedulers growth (pre-processing,AI,Docbewaking, Saldo)
  - Business logic in process layer, expanded with more…
  - Reference project (jBPM) never applied due political/time pressures.


<<Start State>>
Start
<<Node>>
Verstuur vragen
ja
<<Decision>>
Geautomatiseerd afhandelen?
nee
<<Node>>
Aanmaken verkoopkans van versturen
<<Decision>>
Max # reminders bereikt ?
timeout
<<Task Node>>
Wacht op antwoord
vraag nogmaals
nee
niet ok
goedgekeurd
ontvangen
nog niet compleet
<<Node>>
Verwerk deel-antwoorden
<<Task Node>>
Beoordeel antwoorden
<<Node>>
Aanmaken Verkoopkans van ontvangen
<<Decision>>
Antwoorden ok en compleet?
<<Decision>>
Error Redirection
nee
<<Decision>>
Verwerk Deelantwoorden Voldoende Saldo?
handmatig
Opnieuw
ja
<<Node>>
Openen Contractrekening
afbreken
<<Task Node>>
BackendMelding
compleet en ok
<<Node>>
Informeren Aanvrager
afbreken
handmatig
<<Node>>
Informeren Aanvrager Contact Aanmaken
ja
<<End State>>
Einde
Handmatig


<<Start State>>
Start
<<Node>>
Verstuur vragen
ja
<<Decision>>
Geautomatiseerd afhandelen?
<<Node>>
Aanmaken verkoopkans van versturen
nee
<<Node>>
Aanmaken Softlogin Verkoopkans
vraag nogmaals
nee
<<Decision>>
Error Redirection
Opnieuw
niet ok
<<Decision>>
Max # reminders bereikt ?
timeout
<<Task Node>>
Wacht op antwoord
beschikbaar
goedgekeurd
ontvangen
<<Task Node>>
Beoordeel antwoorden
<<Task Node>>
BackendMelding
<<Node>>
Verwerk deel-antwoorden
nog niet compleet
<<Node>>
Aanmaken Verkoopkans van ontvangen
<<Decision>>
Antwoorden ok en compleet?
<<Decision>>
Verwerk Deelantwoorden EB Contract?
ja
nee
<<Process State>>
AanmaakSoftlogin
nee
afbreken
handmatig
<<Decision>>
Verwerk Deelantwoorden Voldoende Saldo?
ja
compleet en ok
<<Node>>
Openen Contractrekening
<<Node>>
Informeren Aanvrager
<<Node>>
Informeren Aanvrager Contact Aanmaken
ja
<<Node>>
Afwijzen Verkoopkans Softlogin
handmatig
<<End State>>
Einde
Handmatig

Agenda

- A perfect world
- Reality intrudes
- Real life example: STP Aanvragen
  - Introduction
  - Life in the trenches
  - Example: web service
  - Example: jBPM process
- Building on Legacy: STP Contractrekeningen (Deposito's)
  - Use Cases and legacy problems
- **Towards the future: STP Betalen**
  - Use Cases, getting better?

STP Betalen UCM/UC's

- 1x UCM, 10x UC's
- 65x findings in team review (one team member!)
  - Object model in UCM!
  - 'boolean' appears 2x in UC's!
  - Instructs where data elements needs to go and exactly the contents to be stored!
  - Details system components as actors!
  - 2x technically impossible to implement process steps!

Questions?

http://www.schabell.org

eric@schabell.org