



WERKGROEP GAP		308
20 april 2010	Not.	8

**Aanwezig:** Johan, Tim, Bart  
**Verontschuldigd:** Peter, Broes

## Agenda

- Implementatie
  - Loose Coupling
  - Exceptions
- Management
  - Op weg naar beta-2

## Loose Coupling

Zie ticket #433: <https://develop.chiro.be/trac/cg2/ticket/433>

*Entity's mogen door onze services niet meer als datacontract gebruikt worden. In de plaats komen onze -Info en -Detail-klasses. Door Automapper te gebruiken, blijft de inspanning hiervoor beperkt.*

Zie ticket #450: <https://develop.chiro.be/trac/cg2/ticket/450>

*Enums zoals GeslachtsType, LidType,... worden als datacontract gebruikt, maar zitten samen met de entity's in het project Chiro.Gap.Orm. Omdat we de entity's weg willen als datacontract (zie #443), is het beter de enums die de service wel wil gebruiken in een ander project onder te brengen. (Zo moeten de consumers van de service Chiro.Gap.Orm niet meer referencen.)*

Ik heb de zaken die zowel voor de UI als voor de business nodig zijn, bijeengezet in Chiro.Gap.Domain:  
<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Domain>

Omdat exceptions enkel gethrowd worden in Chiro.Gap.Workers, en enkel gecatcht worden in Chiro.Gap.Services (die Workers aanroept), heb ik de exceptions gemigreerd naar Chiro.Gap.Workers.Exceptions. Zie  
<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Workers/Exceptions>

Op die manier kan het project Chiro.Gap.Fouten weg.

*We kiezen toch voor een enum als foutcode; dat is meer typesafe. Uit het vorige verslag had ik verkeerd begrepen dat enums geen goed idee zouden zijn voor exceptions. Het was anders: enums gebruiken als generic parameter voor de foutcode was niet goed. Met een enum voor foutcodes is op zich niets mis. (#465)*

## Exceptions (ctd)

De 'basisexception' voor ons project is nu GapException:  
<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Workers/Exceptions/GapException.cs>

Alle nodige constructoren zijn nu voorzien, ook die voor deserializatie:

```

#region serializatie

protected GapException(SerializationInfo info, StreamingContext context)
: base(info, context)
{
    if (info != null)
    {
        _foutNummer = info.GetInt32("_foutNummer");
        _items = (IEnumerable<string>)info.GetValue(
            "_items",
            typeof(IEnumerable<string>));
        // FIXME: underscores in de 'names' (_foutNummer, _items, ...)
        // weglaten. (zie #466)
    }
}

public override void GetObjectData(SerializationInfo info, StreamingContext context)
{
    base.GetObjectData(info, context);
    if (info != null)
    {
        info.AddValue("_foutNummer", _foutNummer);
        info.AddValue("_items", _items);
    }
}

#endregion

```

Zie ook de unittest voor (de)serializatie van de GapException:

<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/TestProjecten/Chiro.Gap.Workers.Test/GapExceptionTest.cs#L74>

Een GapException heeft een foutnummer. Deze foutnummers zijn gedefinieerd in

<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Domain/Foutnummers.cs>

In een aantal gevallen kan een GapException rechtstreeks gebruikt worden, bijv:

```

else if (l.GroepsWerkJaar.ID != _groepsWjDao.RecentsteOphalen(l.GroepsWerkJaar.Groep.ID).ID)
{
    throw new GapException(
        Foutnummers.GroepsWerkJaarNietBeschikbaar,
        Properties.Resources.GroepsWerkJaarVoorbij);
}

```

Andere specifieke exceptions erven van GapException, zoals bijv:

- GeenGavException (voor alle gevallen van 'geen GAV'; geeft zo goed als geen info)  
<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Workers/Exceptions/GeenGavException.cs>
- BestaatAlException<T> (voor als men 1 object probeert te maken, terwijl dat al bestaat)  
<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Workers/Exceptions/BestaatAlException.cs>
- BlokkerendeObjectenException<T> (voor als meerdere objecten in de weg zitten)  
<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.Workers/Exceptions/BlokkerendeObjectenException.cs>

Als de servicelaag een exception catcht, wordt die (in het ideale geval) als FaultException, volgens een bepaald faultcontract, over de service gestuurd. De FaultContracts corresponderend met de Exceptions vind je in

<https://develop.chiro.be/trac/cg2/browser/trunk/Solution/Chiro.Gap.ServiceContracts/FaultContracts>.

Feedback?

Foutnummer en 'Items' zijn niet relevant voor BestaatAlException, GeenGavException,... Vandaar dat we die weghalen uit GapException, en we een specifieke exception maken die wel foutnummer en items heeft. Naam nog te bedenken. (FoutNummerException?). Zie #467.

## Op naar beta-2

- Deadline: 31 mei. **We verschuiven dat naar 7 juni.**
- In het oog springende features:
  - Heel wat bugfixing op de bestaande app
  - Buitenlandse adressen (#238)
    - **Zie EA.**
    - **Datacontract:**
      - **ID's verwijderen, maar enkel strings uitwisselen (namen van gemeentes, straten,...)**
      - **Toevoegen Land**
    - **Business bepaalt hoe adressen gesaved moeten worden, en checks die moeten gebeuren**
    - **UI: standaard: land=België. Als dat verandert, vervallen de Ajax-trucs.**
  - 'eigen' dingen van de groep
    - Eigen functies (#428)
    - Vrije velden op persoon- en lidniveau (#279, #280)
    - Persoonlijke instellingen (paginagrootte, standaardwoonplaats, standaardgeslacht) (#377, #323)
  - Import initiele gegevens uit kipadmin en aansluitingsbestanden (eenmalig) (#100)
- Voorstel taakverdeling (ik zeg maar iets; kan uiteraard anders)
  - Broes: PersoonLidInfo (#434)
  - Bart: Eigen functies (#428) en user documentatie (nog geen ticket)
  - Tommy: Persoonlijke instellingen (#377, #323), Excel (#464)
  - Peter: Nog eens terug bekijken user interface. Zijn de gemaakte tekeningen van vroeger nog OK? Moeten ze aangevuld worden? Kan een prototype gemaakt worden. (#468)
  - Johan: werkt aan de overige taken
- **Als je aan een taak aan het werken bent, accepteer ze dan ook in de trac, om dubbel werk en conflicten te vermijden.**

Opmerking: Referentiele integriteit en uniciteit moeten door de database gecontroleerd worden, omdat zij dat het beste kan. Dat zal wel wat refactoring vragen. Bij catchen van SQL exception altijd error code checken. Pas dan 'problematische' leden/entity's/... ophalen. (#469)

Gezinsgebonden telefoonnr's: 2 use cases

- 1. broer of zus maken: gezinsgebonden nummers overnemen
- 2. gezinsgebonden nr toevoegen/wijzigen: gezinsgenoten nemen nummer over

Deze feature is verschoven tot na 1.0.

Excel genereren. Er bestaan library's om ooxml-documenten mee te manipuleren (#464).

## Varia

- Feedback van 2 testers, en 1 wannabe-tester! ☺
- Lid, Kind, Leiding; hoe krijgt de user dit te zien. **User ziet Leden, Leiding en 'Leden en Leiding' ipv Kind, Leiding, Lid. Misschien icoontjes om het duidelijk te maken. (Er bestaan misschien icoontjes voor in de vorige beleidsnota.) (#470)**
- Licentie resharper OK
- Opzoeken in msdn download: Welke editie van Visual Studio kunnen we krijgen?
- Team foundation server is nu makkelijker te krijgen via msdn. Na te kijken. (Msdn professional?)