

# Requirements Engineering: League of Legends eSports

Bellemans Jonah

Caerts Stijn  
Verhoest Elise

Dimova Yana  
Vos Pieter

Lambrichts Midas

23 december 2016

# Inhoudsopgave

<b>1</b>	<b>Motivatie sportkeuze</b>	<b>2</b>
<b>2</b>	<b>Beschrijving Competitie</b>	<b>2</b>
2.1	Plaatsingsmatches . . . . .	2
2.2	Wedstrijdverloop . . . . .	2
2.3	Competitieverloop . . . . .	2
2.3.1	Eerste competitieronde: Group Stage . . . . .	2
2.3.2	Tweede competitieronde: Quarterfinals . . . . .	3
2.3.3	Derde competitieronde: Semifinals . . . . .	3
2.3.4	Finale . . . . .	3
<b>3</b>	<b>Systeemcontext</b>	<b>4</b>
<b>4</b>	<b>Doelen</b>	<b>8</b>
4.1	Beschrijving Goals . . . . .	8
4.2	And/Or-decomposition . . . . .	13
4.3	Dependencies . . . . .	15
4.4	Rationale . . . . .	17
<b>5</b>	<b>Scenarios</b>	<b>17</b>
5.1	“Planning opstellen” . . . . .	17
5.2	“Planning updaten” . . . . .	18
5.3	“Planning wijzigen” . . . . .	18
5.4	“Ploeg aanmaken” . . . . .	19
<b>6</b>	<b>Datamodel</b>	<b>19</b>
6.1	UML diagram . . . . .	19
6.2	OCL Constraints . . . . .	20
<b>7</b>	<b>State machine</b>	<b>21</b>

# 1 Motivatie sportkeuze

We opteerden om e-sports, meer bepaald de “League of Legends World Championships (LoL Worlds)”, te kiezen als sporttak voor dit project. Wij kozen hiervoor omdat deze sporttak origineel is, en een zeer duidelijke afbakening van de spel- en wedstrijdreglementen voorziet. Aangezien de meerderheid van de groepsleden vanuit een computer science-gerelateerde achtergrond komt, is de e-sport goed door de leden gekend, en is iedereen gemotiveerd om hieraan te werken.

## 2 Beschrijving Competitie

### 2.1 Plaatsingsmatches

Teams plaatsen zich voor de competitie door middel van plaatsingswedstrijden die vroeger in het seizoen plaatsvinden. Voor elke regio (West-Europa, Noord-Amerika, Korea, . . .) worden een aantal “Seeds” vrijgehouden, die worden ingevuld door de beste ploegen van die regionale competities. De absolute topploegen van de wereld, zoals winnaars van eerdere jaren, kunnen via een “wildcard” rechtstreeks uitgenodigd worden voor deelname door het organisatorisch comité.

### 2.2 Wedstrijdverloop

Elke wedstrijd van het spel League of Legends (“LoL”) wordt gespeeld door twee teams van elk vijf spelers. De teamleden besturen elk een individueel personage in de game, die men “Champions” noemt. Na het selecteren van de champions voor elke speler, worden de teams in de spelwereld (“Summoner’s Rift”) gezet. Het doel van het spel is om de “Nexus” van het andere team te vernietigen, terwijl men de eigen Nexus verdedigt. Het spel blijft lopen totdat er de Nexus van één van beide teams vernietigd is. Het is dus onmogelijk om een match te beëindigen zonder dat er een éénduidige winnaar bepaald is.

### 2.3 Competitieverloop

#### 2.3.1 Eerste competitieronde: Group Stage

De wereldkampioenschappen beginnen met 16 verschillende teams, die opgedeeld worden in groepen. Binnen de groep spelen de teams telkens één tegen één. De “matchups” worden bepaald met een round-robin systeem, waarbij elk team tweemaal niet-opéenvolgend tegen ieder ander team speelt (eenmaal aan elke zijde van Summoner’s Rift).

**Tiebreaker** Als twee teams dezelfde score behalen, meer bepaald, hetzelfde win-percentages behalen, zal de zogenaamde “head-to-head” score (de score die door de teams behaald werd tijdens hun onderlinge wedstrijden) gebruikt worden om een winnaar te bepalen. Als ook hieruit geen winnaar besloten kan worden, zal een enkele tiebreak match gespeeld worden.

### **2.3.2 Tweede competitieronde: Quarterfinals**

Tijdens de kwartfinales ( "Quarterfinals" ) blijven slechts acht seeds over, die ingenomen worden door de twee hoogst scorende teams uit iedere groep van de Group Stage. De teams die #1 werden binnen hun groep, spelen tegen de teams die #2 werden in de andere groep. Welke groepen dan juist gecombineerd worden, wordt willekeurig bepaald. Eens er een matchup bepaald is, ligt daarmee ook de "samenhangende" matchup vast. Als de willekeurige trekking dus bepaalt dat Team 1 uit groep A speelt tegen Team 2 uit groep D, dan zal ook Team 2 uit groep A spelen tegen Team 1 uit groep D. Nadat alle matchups vastliggen, spelen de gekozen teams telkens een "best-of-five" match, waaruit de winnaar doorstroomt naar de volgende ronde.

### **2.3.3 Derde competitieronde: Semifinals**

In de halve finale ( "Semifinals" ) spelen de winnende teams uit de kwartfinales opnieuw een "best-of-five" match tegen elkaar. De twee overblijvende winnende teams gaan vervolgens naar de finale.

### **2.3.4 Finale**

De finale wordt gespeeld door de twee overgebleven teams. Ook hier wordt de uiteindelijke winnaar bepaald door een "best-of-five" match. Het team dat op het einde van deze vijf games het meeste gewonnen heeft, is de uiteindelijke winnaar van het toernooi.

### 3 Systeemcontext

- Subject Context Facet
  - Requirement Sources
    - \* Stakeholders
      - Wedstrijdleader
      - Competitieleider
      - Spelers
      - Scheidsrechters
      - Shoutcasters
    - \* Documentatie
      - World Championship rule set
      - Leaguepedia
      - League of Legends Terms of Use
    - \* Bestaande systemen
      - Vlaamse Basketbal Liga
  - Context Objects
    - \* Spelers
    - \* Scheidsrechters
    - \* Games
    - \* Seizoenen
    - \* Stadium
    - \* Materiaal (pc's, toetsenborden, muizen)
    - \* Internetconnectie
    - \* Internetsnelheid
    - \* Publiek
    - \* League accounts
  - Properties and relationships
    - \* Efficiente planning van de wedstrijden
    - \* Correctheid van de planning
    - \* Respect tegenover het reglement
    - \* Beschikbaarheid van de spelers
    - \* Beschikbaarheid van de zalen
    - \* Beschikbaarheid van de scheidsrechters
    - \* Beschikbaarheid van de shoutcasters
    - \* Werking van het materiaal
    - \* Werking van het internet
    - \* Veiligheidsmaatregelen
    - \* Internetsnelheid

- \* De games worden gespeeld in het juiste seizoen
  - \* De volgorde van de seizoenen wordt gerespecteerd
- Usage Context Facet
  - Requirement Sources
    - \* Stakeholders
      - Spelers
      - Publiek
      - Scheidsrechters
      - Competitieleider
      - Wedstrijdleider
      - Shoutcasters
    - \* Systemen
      - Andere systemen die resultaten van een competitie bijhouden (voor een andere sport bijv.)
      - Andere live uitzending
  - Context Objects
    - \* Publiek
    - \* Spelers
    - \* Competitieleider
    - \* Wedstrijdleider
    - \* Scheidsrechters
    - \* Shoutcasters
  - Properties and relationships
    - \* Gebruik van de user interface
    - \* Efficiënte planning van de wedstrijden
    - \* Toewijzen van scheidsrechters voor games
    - \* Publiek, shoutcasters en scheidsrechters kijken wedstrijd ter plaatse
    - \* Publiek kijkt wedstrijd via streaming service
    - \* Publiek dat teams en spelers kan bekijken
    - \* Updaten van de resultaten
    - \* Update competitie via website
- IT Context Facet
  - Requirement Sources
    - \* Stakeholders
      - Wedstrijdleider
      - Competitieleider
      - Architect

- Webdesigner
  - Software developer
  - CEO, CTO...
  - Technology consultants
- \* Systemen
  - Andere systemen die resultaten van een competitie bijhouden (voor een andere sport bijvoorbeeld)
  - Andere live uitzendingen
  - Setup van andere toernooien die op een LAN netwerk gespeeld worden
- Context Objects
  - \* Website
  - \* Internet
  - \* User interface
  - \* User database
  - \* Scheidsrechters database
  - \* Zalen database
  - \* Resultaten database
  - \* Stream mogelijkheden
- Properties and relationships
  - \* Updaten van resultaten
  - \* Correct bijhouden van gegevens in databases
  - \* Live uitzending van de stream
  - \* Beschikbaarheid van website
  - \* Gebruiksvriendelijkheid van user interface
- Development Context Facet
  - Requirement Sources
    - \* Stakeholders
      - Requirements engineering
      - Architecten
      - Competiteider
      - Webdesigner
      - Software developpers
      - IT-department
      - Technology Consultants
  - Context Objects, Properties and relationships
    - \* Development van web interface
    - \* Website
    - \* Databases

- \* Integrated Development Environment
  - \* Gangbare standaarden van streaming services
  - \* Streaming websites (uitzenden van de matches)
  - \* Gebruiksovereenkomsten streaming software
  - \* Webdesigners zorgen ervoor dat stream altijd beschikbaar is tijdens uitzendingen
- Out of scope
  - Ticketverkoop
  - Catering
  - Speciale voorzieningen voor spelers



## 4 Doelen

### 4.1 Beschrijving Goals

**Goal G1:** Het systeem zorgt ervoor dat de competitieleiders, de scheidsrechters, de spelers (en teamleiders), de ploegen efficiënt en betrouwbaar kunnen beheren

**Goal G1,1:** Het systeem moet een gebruiksvriendelijke invoer van gegevens voorzien

**Goal G1,1,1:** Het systeem moet ervoor zorgen dat de ploegen en spelers (actieve en reserve spelers) zich op een gebruiksvriendelijke manier kunnen registreren

**Goal G1,1,2:** Het systeem moet ervoor zorgen dat de scheidsrechters zich op een gebruiksvriendelijke manier kunnen registreren (applyn)

**Goal G1,2:** Het systeem moet de betrouwbaarheid van de gegevens bewaren om confidential leaks te voorkomen

**Goal G1,3:** Het systeem moet flexibel zijn zodat het een gemakkelijke update en wijziging van de teamsamenstelling mogelijk maakt

**Goal G1,3,1:** Het systeem moet ervoor zorgen dat de gegevens over de spelers gemakkelijk geupdated en gewijzigd kunnen worden

**Goal G1,3,2:** Het systeem moet ervoor zorgen dat de gegevens over de teamleiders gemakkelijk geupdated en gewijzigd kunnen worden

**Goal G1,3,3:** Het systeem moet ervoor zorgen dat de gegevens over de reservespelers gemakkelijk geupdated en gewijzigd kunnen worden

**Goal G1,4:** Het systeem moet ten alle tijde een gestructureerd overzicht kunnen geven van de ploegen en spelers

**Goal G1,5:** Het systeem moet de toegankelijkheid van de overzichten kunnen aanpassen aan de stakeholders

**Goal G1,5,1:** Het systeem moet te allen tijde een overzicht kunnen geven aan de competitieleider, in dit overzicht moet alle informatie omtrent de ploegen en de spelers verschaft worden

**Goal G1,5,2:** Het systeem moet te allen tijde een overzicht kunnen geven aan de scheidsrechter, in dit overzicht moet alle informatie omtrent de ploegen en de spelers verschaft worden

**Goal G1,5,3:** Het systeem moet te allen tijde een overzicht kunnen geven aan de spelers en de teamleiders van elke ploeg, in dit overzicht moet beperkte informatie omtrent de ploegen en de spelers verschaft worden; dit om competitie tussen de ploegen niet te storen

**Goal G1,5,4:** Het systeem moet te allen tijde een overzicht kunnen geven aan de media, in dit overzicht moet beperkte informatie omtrent de ploegen en de spelers verschaft worden; dit om confidential leaks te voorkomen

**Goal G2:** De competitieleider moet via het systeem een efficiënte, gebruiksvriendelijke en faire wedstrijdplanning mogelijk maken

**Goal G2,1:** Het systeem moet het mogelijk maken dat de competitieleider de wedstrijden kan plannen

**Goal G2,2:** Het systeem moet het mogelijk maken dat de competitieleider de wedstrijden kan wijzigen

**Goal G2, 1-2, 1:** Het systeem moet op een overzichtelijke en betrouwbare manier de beschikbaarheden van de plaatsen, personeel, tijdstippen, scheidsrechters en shoutcasters weergeven

**Goal G2, 1-2, 1, 1:** Het systeem moet het mogelijk maken om de beschikbare plaatsen weer te geven

**Goal G2, 1-2, 1, 1, 1:** Een database moet de beschikbare plaatsen te allen tijde kunnen weergeven

**Goal G2, 1-2, 1, 1, 2:** Een database met de beschikbare plaatsen moet up to date zijn

**Goal G2, 1-2, 1, 1, 2, 1:** Het systeem moet het mogelijk maken om nieuwe plaatsen op een gebruiksvriendelijke manier in de database te plaatsen

**Goal G2, 1-2, 1, 1, 2, 2:** Het systeem moet het mogelijk maken om plaatsen op een gebruiksvriendelijke manier uit de database te verwijderen

**Goal G2, 1-2, 1, 2:** Het systeem moet het mogelijk maken om de beschikbare tijdstippen weer te geven

**Goal G2, 1-2, 1, 2, 1:** Het systeem moet de beschikbaarheden van de ploegen controleren, rekening houdend met het feit dat er geen overlap mag zijn met andere competities (dat ploeg A niet kan spelen in competitie B en competitie C op hetzelfde moment)

**Goal G2, 1-2, 1, 2, 2:** Het systeem moet de haalbaarheid van de beschikbaarheden van de ploegen controleren.

**Goal G2, 1-2, 1, 2, 2, 1:** Het systeem moet controleren dat een ploeg niet 2 maal per dag gescheduled staat

**Goal G2, 1-2, 1, 2, 2, 2:** Het systeem moet controleren dat een ploeg niet speelt tegen ploeg B en ploeg C op hetzelfde moment

**Goal G2, 1-2, 1, 3:** Het systeem moet het mogelijk maken om de beschikbare scheidsrechters weer te geven

**Goal G2, 1-2, 1, 3, 1:** Het systeem moet informatie halen uit een database die up to date is

**Goal G2, 1-2, 1, 3, 1, 1:** Het systeem moet een gemakkelijke invoer van kandidaten uit de database door de competitieleider mogelijk maken

**Goal G2, 1-2, 1, 3, 1, 1, 1:** De competitieleider moet de kandidatuurstellingen op een efficiënte manier kunnen controleren en goedkeuren

**Goal G2, 1-2, 1, 3, 1, 1, 1, 1:** De kandidaat scheidsrechters moeten zich op de website via een gebruiksvriendelijk formulier kunnen inschrijven

**Goal G2, 1-2, 1, 3, 1, 2:** Het systeem moet een gemakkelijke eliminatie van kandidaten uit de database door de competitieleider mogelijk maken

**Goal G2, 1-2, 1, 3, 2:** Het systeem moet de haalbaarheid van de beschikbaarheid van de scheidsrechters controleren

**Goal G2, 1-2, 1, 3, 2, 1:** Het systeem moet controleren dat de verplaatsingsmogelijkheden reel zijn (bv. scheidsrechter A kan niet match fluiten in Zuid-Korea en de volgende dag een fluiten in de VS)

**Goal G2, 1-2, 1, 3, 2, 2:** Het systeem moet controleren dat een scheidsrechter niet aanwezig moet zijn op 2 matches tegelijkertijd

**Goal G2, 1-2, 1, 4:** Het systeem moet het mogelijk maken om de beschikbare shoutcasters weer te geven

**Goal G2, 1-2, 1, 4, 1:** Het systeem moet informatie halen uit een database die up to date is

**Goal G2, 1-2, 1, 4, 1, 1:** Het systeem moet een gemakkelijke invoer van kandidaten uit de database door de competitieleider mogelijk maken

**Goal G2, 1-2, 1, 4, 1, 1, 1:** De competitieleider moet de kandidatuurstellingen op een efficiënte manier kunnen controleren en goedkeuren

**Goal G2, 1-2, 1, 4, 1, 1, 1, 1:** De kandidaat shoutcasters moeten zich op de website via een gebruiksvriendelijk formulier kunnen inschrijven

**Goal G2, 1-2, 1, 4, 1, 2:** Het systeem moet een gemakkelijke eliminatie van kandidaten uit de database door de competitieleider mogelijk maken

**Goal G2, 1-2, 1, 4, 2:** Het systeem moet de haalbaarheid van de beschikbaarheid van de shoutcasters controleren

**Goal G2, 1-2, 1, 4, 2, 1:** Het systeem moet controleren dat de verplaatsingsmogelijkheden reel zijn (bv. shoutcaster A kan niet match fluiten in Zuid-Korea en de volgende dag een becommentariëren in de VS)

**Goal G2, 1-2, 1, 4, 2, 2:** Het systeem moet controleren dat een shoutcaster niet aanwezig moet zijn op 2 matches tegelijkertijd

**Goal G3:** Het systeem moet de wedstrijduitslagen op een efficiënte manier bijhouden en overzichtelijke tussenstanden weergeven; deze info is toegankelijk voor de competitieleider, scheidsrechters, media, spelers, teamleiders en shoutcasters

**Goal G3, 1:** Het systeem moet de wedstrijduitslagen op een efficiënte manier registreren

**Goal G3, 1, 1:** De wedstrijdleader moet de wedstrijduitslagen op een gebruiksvriendelijke manier kunnen invoeren

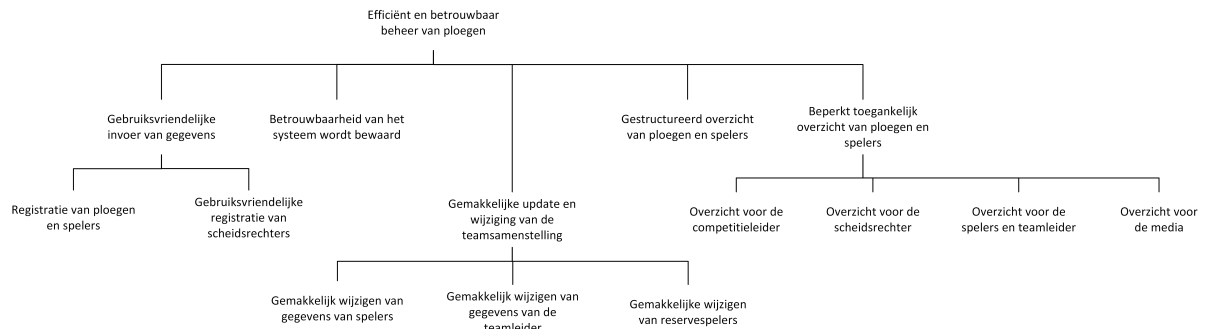
**Goal G3, 1, 2:** De wedstrijdleader en de competitieleider moeten de uitslagen bevestigen

**Goal G3, 1, 2-3, 1:** De wedstrijdleader en de competitieleider moeten via het systeem efficiënt kunnen communiceren met elkaar

**Goal G3, 1, 3:** Het systeem moet de tussenstand van de competitie efficiënt kunnen berekenen

**Goal G4:** informatie omtrent de wedstrijdplanning en samenstelling van ploegen moet beschikbaar gesteld worden aan de media, de competitieleider, de scheidsrechters, de spelers (en teamleiders) en de ploegen

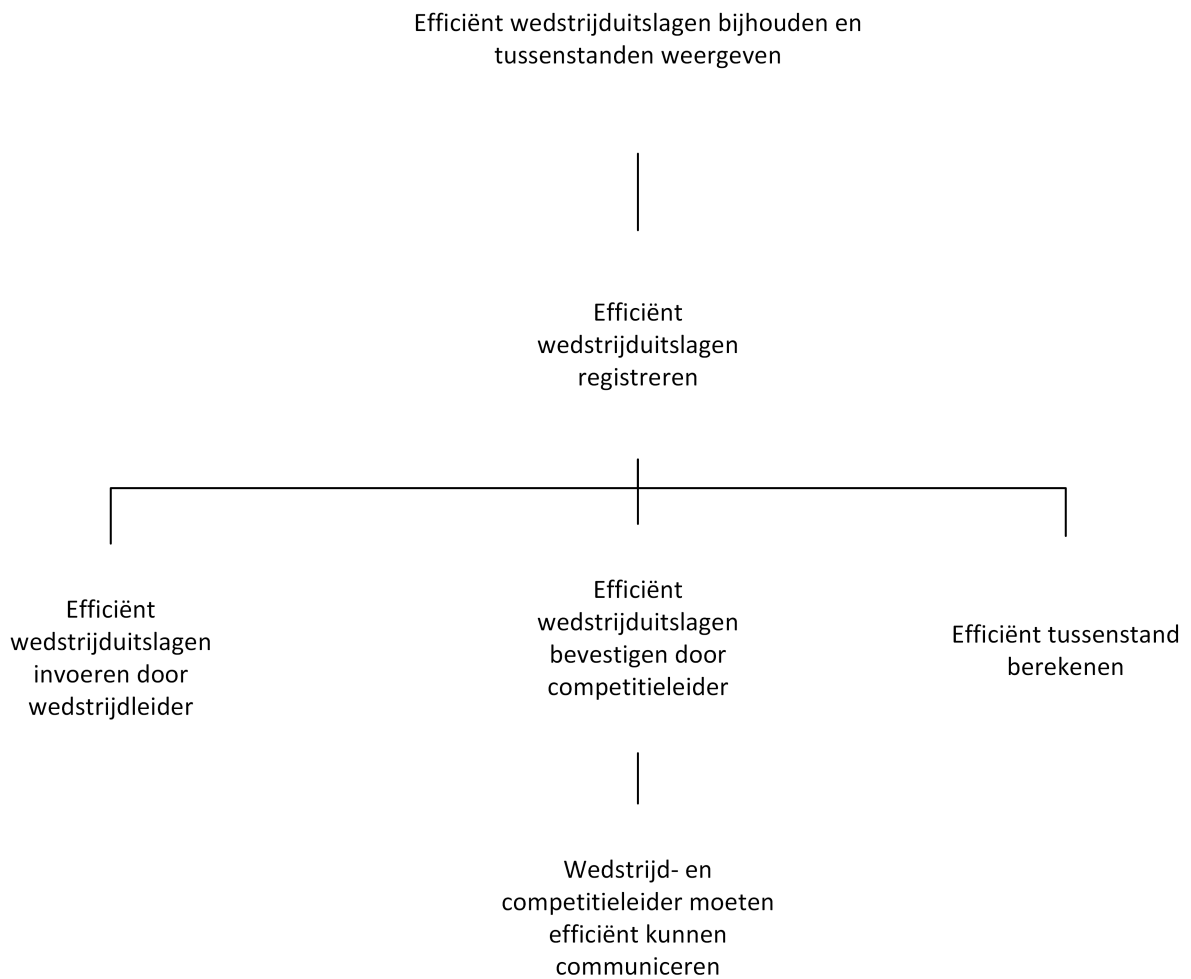
## 4.2 And/Or-decomposition



Figuur 1: And/Or decompositie van G1

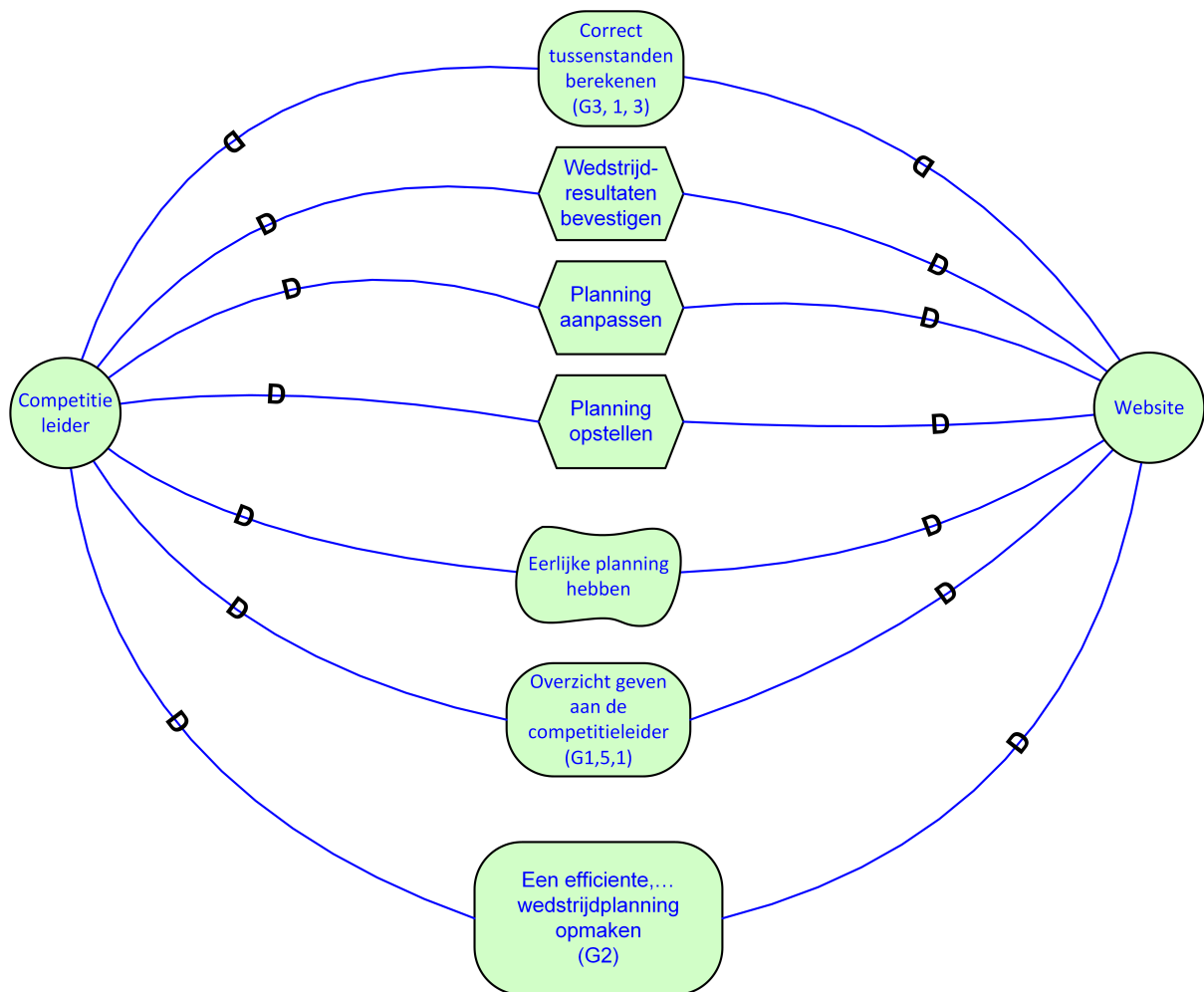


Figuur 2: And/Or decompositie van G2

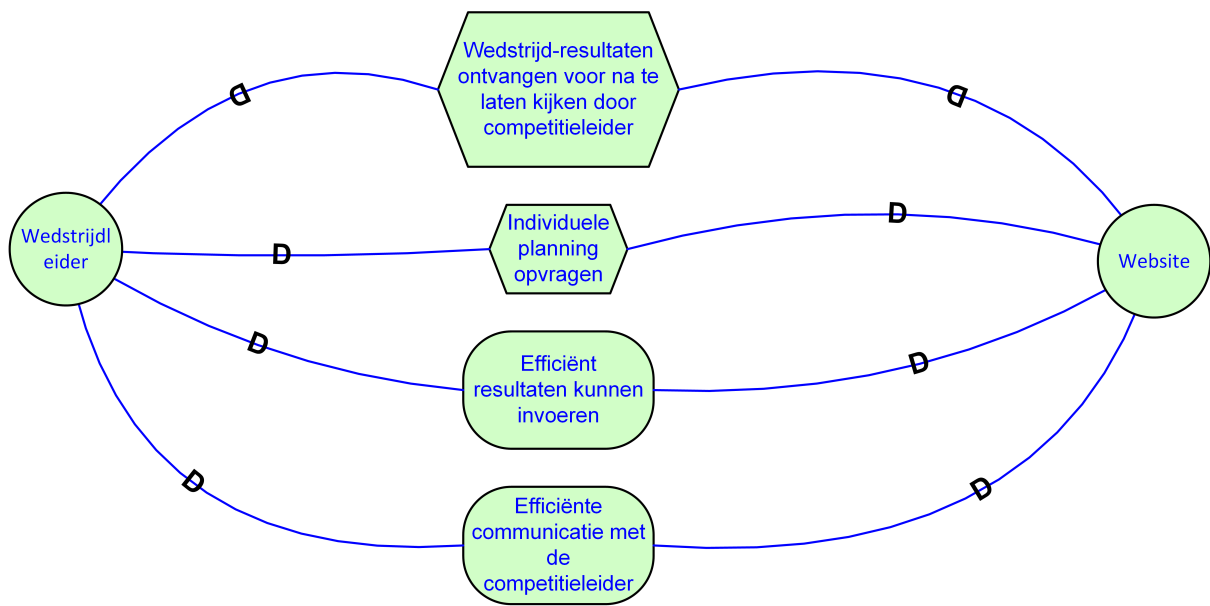


Figuur 3: And/Or decompositie van G3

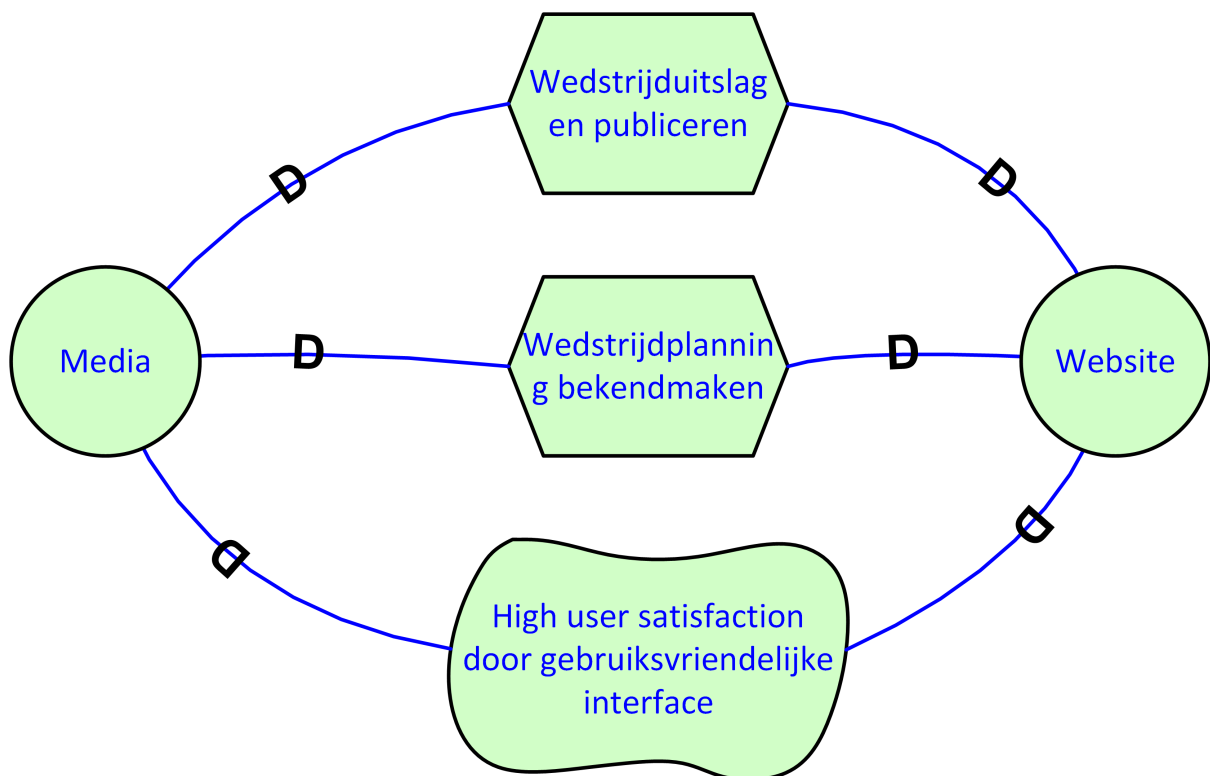
### 4.3 Dependencies



Figuur 4: First dependency model in i\*



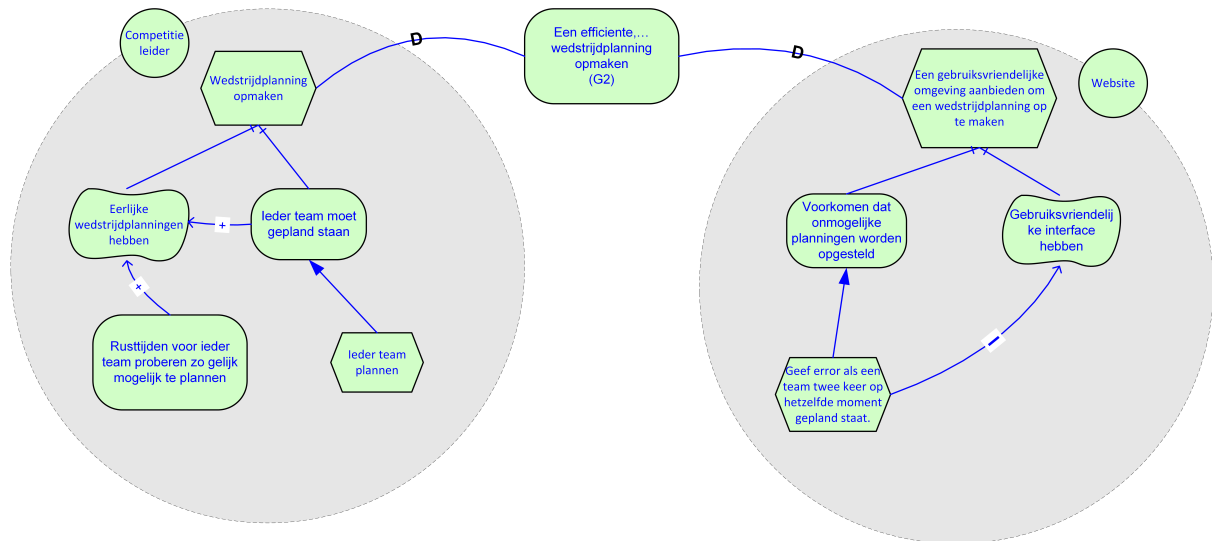
Figuur 5: Second dependency model in i\*



Figuur 6: Third dependency model in i\*



## 4.4 Rationale



Figuur 7: Rationale model in i\*

## 5 Scenarios

### 5.1 “Planning opstellen”

De competitieleider wil een nieuwe match tussen 2 ploegen in de planning zetten. Hij gaat naar de website, drukt op “log in” en voert zijn persoonlijke inloggegevens in. De competitieleider heeft de optie om de wedstrijdplanning aan te passen of een nieuwe match toe te voegen op de website. Hij gaat naar de webpagina waar hij een nieuwe wedstrijd kan toevoegen via de knop “wedstrijd toevoegen”. Vervolgens krijgt hij een overzicht van de beschikbaarheden van de ploegen, plaatsen, tijdstippen, shoutcasters, scheidsrechters. (Hij ziet een paar velden waar hij de naam van de ploegen, de plaats, het tijdstip, de scheidsrechter en de shoutcasters nodig voor de match kan invullen.) Hij vult de namen van de 2 ploegen in die hij tegen elkaar wilt laten spelen. Hij selecteert een plaats tussen de mogelijkheden opgeslagen in de database. De competitieleider voegt vervolgens een beschikbaar tijdstip toe, alsook een scheidsrechter uit de database die op het gegeven tijdstip beschikbaar is. De competitieleider wijst ook shouters toe voor de wedstrijd. De competitieleider slaat de nieuwe gegevens op en sluit de website.

## **5.2 “Planning updaten”**

1. De competitieleider opent de website
2. De competitieleider drukt op “log in”
3. De competitieleider meldt zich aan met zijn persoonlijke inloggegevens
4. De competitieleider klikt op de knop “wedstrijdplanning aanpassen”
5. De competitieleider klikt op de knop “alle geplande wedstrijden weergeven”
6. De competitieleider klikt op de planning die hij wilt aanpassen
7. De competitieleider klikt op de knop “invoeren ploegnaam”
8. De competitieleider voert de naam van de ploeg in, in het veld “Ploegnaam”
9. De competitieleider klikt op de knop “wijzigingen opslaan”
10. De competitieleider klikt op de knop “afmelden”
11. De competitieleider sluit de competitiewebsite

## **5.3 “Planning wijzigen”**

De competitieleider probeert op de planning op de website een timeslot te wijzigen dat al voorbij is:

1. De competitieleider opent de website
2. De competitieleider drukt op “log in”
3. De competitieleider meldt zich aan met zijn persoonlijke inloggegevens
4. De competitieleider klikt op de knop “wedstrijdplanning aanpassen”
5. De competitieleider klikt op de knop “alle geplande wedstrijden weergeven”
6. De competitieleider klikt op de planning die hij wilt aanpassen
7. De competitieleider klikt op de knop “wijzig tijdstip”
8. De competitieleider klikt op een timeslot dat reeds voorbij is.
9. Er opent een waarschuwingsvenster waar opstaat: ‘dit timeslot kan niet aangepast worden: datum is al voorbij’.
10. De planning blijft onveranderd.

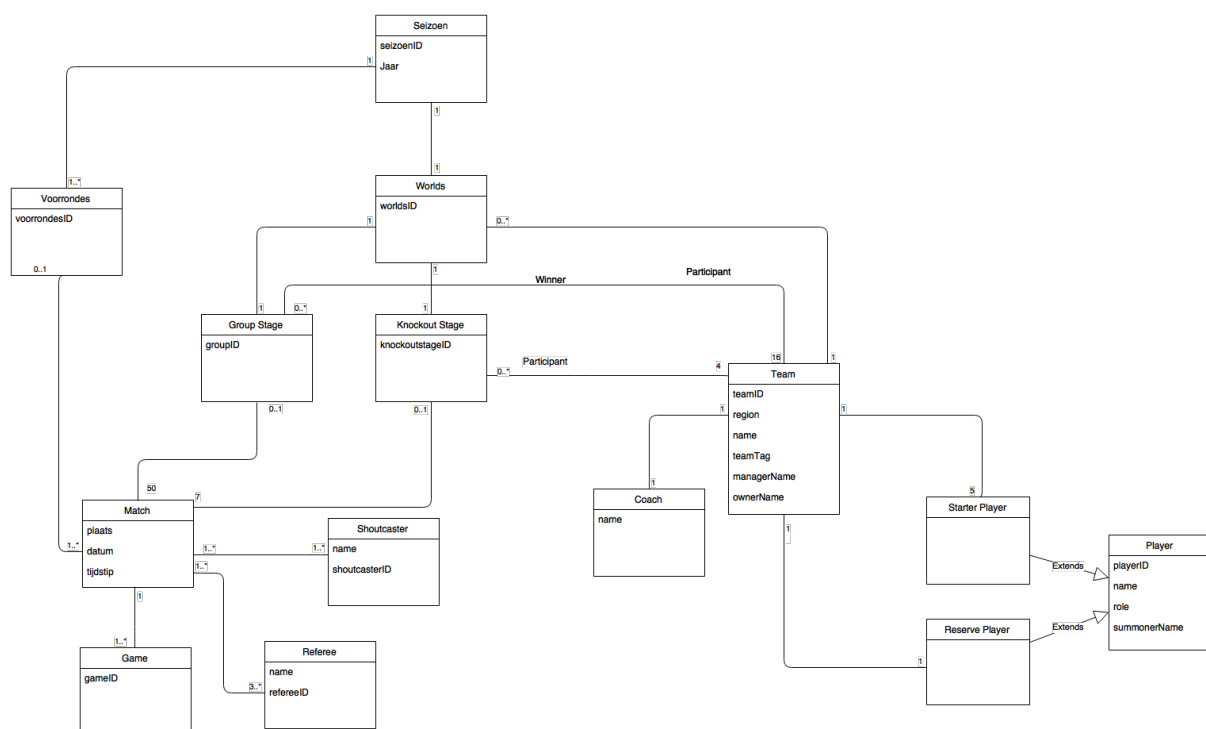
## 5.4 “Ploeg aanmaken”

Parth Naidu, de coach van ploeg Team SoloMid, wilt Sneaky, een speler van ploeg Cloud 9, toevoegen aan zijn team:

Parth Naidu, de coach van ploeg Team SoloMid, opent de website. Hij klikt op “log in” en logt vervolgens in met zijn persoonlijke inloggegevens. Vervolgens ziet hij een link naar “ploegen beheren”, waar hij dan ook op klikt. De site brengt hem dan naar een pagina waar hij zijn ploeg kan beheren, op die pagina staat ook een link om een speler toe te voegen. Hij klikt op deze link en krijgt vervolgens een formulier dat hij kan invullen om een nieuwe speler aan zijn ploeg toe te voegen. Hij vult de gegevens van Sneaky, een speler van ploeg Cloud 9, in op het formulier. Hij probeert dit ingevulde formulier in te vullen. De website weigert deze verandering door te voeren en laat een waarschuwing zien aan Parth Naidu, waar op staat: “Deze speler speelt al voor een andere ploeg, hij mag niet voor 2 ploegen tegelijk spelen. Hij moet eerst de andere ploeg verlaten vooraleer hij bij een andere ploeg kan toegevoegd worden.”

## 6 Datamodel

### 6.1 UML diagram



Figuur 8: UML representatie van het model

## 6.2 OCL Constraints

```
context Worlds invariant  
group_stage.Participant ->  
    includesAll(knockout_stage.participant)
```

Alle teams van de knockout stage moeten ook aan de group stage hebben meegedaan.

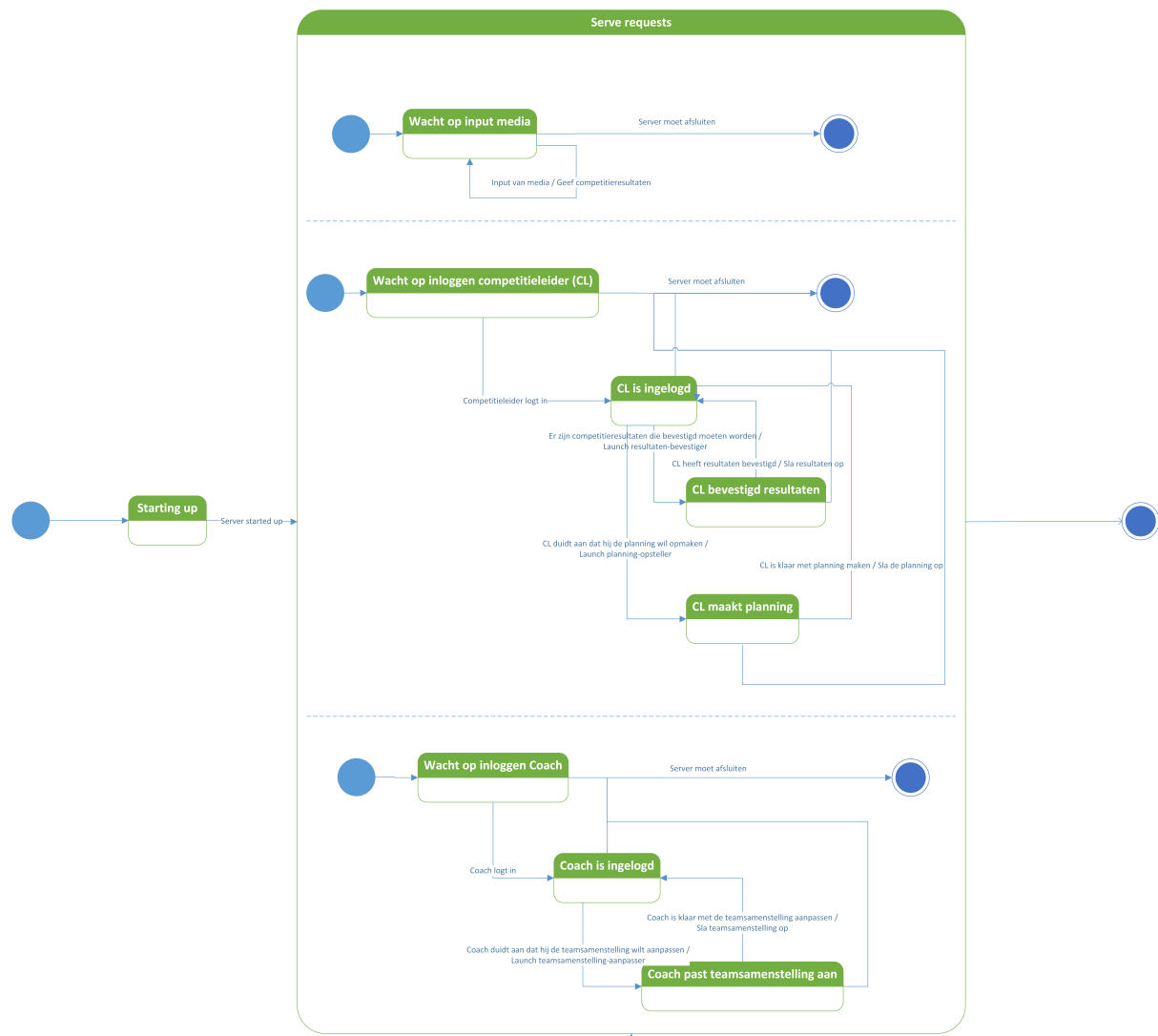
```
context Worlds invariant  
Knockout_stage.Participant -> includesAll(winner)
```

De winnaar van worlds moet hebben meegedaan in de knockout stage.

```
context Team invariant  
starter_player -> excludeAll(reserve_player)
```

Een speler is oftewel een starter player oftewel een reserve player, hij kan niet beide tegelijk zijn.

## 7 State machine



Figuur 9: State Machine