Verslag Tinlab Advanced Algorithms

J. I. Weverink
...

7 april 2021

Inhoudsopgave

1 Inleiding

Zie hier een referentie naar Royce [?] en nog een naar Clarke [?]...

2 Requirements

2.1 Requirements

Requirements zijn beschrijvingen over hoe een product zou moeten functioneren. Zo verandert de betekenis van een requirement als de machine in een andere omgeving wordt geplaats. De requirements voor de verwaring van een ruimte bijvoorbeeld: Binnen moet het altijd warm zijn. In nederland kunnen we zeggen dat 25°C als warm wordt aangezien. Terwijl op de noordpool dat op een lager punt zal zijn.

Anders gezegd zijn requirements geen harde eisen. Dit komt doordat de requirements zijn geformuleerd van het perspectief van de opdrachtgever. De opdrachtgever kan de requirtements geven zonder kennis te hebben van de machine die het moet gaan uitvoeren. De requirements die zijn opgesteld geven dan ook geen grenzen aan die overscheden kunnen worden.

Onder requirements zijn er verschillende soorten requirements. Zo zijn system requirements opgesteld voor het hele systeem en bevatten subsystemen die die kunnen bestaan uit software en hardware. Hier moet uiteindelijk alles ervoor zorgen dat deze requirement wordt gehaald. Software requirement zijn niet bedoeld voor het hele systeem, maar behappen alleen de de software van het systeem. Software requirement zijn niet bedoeld voor het hele systeem, maar behappen alleen de software van het systeem. De software requirements kunnen gaan over over de functionele eisen, gebruikers eisen en zakelijke vereisten. Requirements zijn onder te verdelen in verschillende delen:

- Functional Requirement
- Performance requirement
- Usability requirement
- User requirement
- Interface requirement
- Modes requirement
- Adaptability requirement
- Physical requirement
- Design requirement
- Enviromental requirement
- Logistical requirement

TODO SOFTWARE REQUIREMENTS

2.1.1 Functionele eisen en niet-functionele eisen

Functionele eisen geven aan wat het systeem moet doen en kunnen. Niet-functionele eisen geven de eigenschappen aan van het systeem, zoals snelheid, veiligheid en

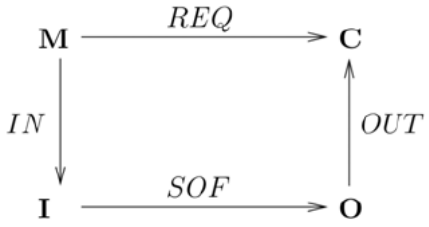
bruikbaarheid. Met andere woorden functionele eisen geven informatie over het "wat", niet-functionele eisen geven informatie over het "hoe"

2.2 specificaties

Uitspraak over gedeelde fenomenen of doelen die de machine moet bereiken middels de onderdelen waar die machine uit bestaat of middels de fenomenen waar de machine controle over heeft.
Doorgaands preciezer, meetbaar, exact geformuleerd.
Wel precies, beschrijven hoe de machine vanuit zijn mogelijkheden de doelen opgesteld in de requirements kan bereiken.

2.3 Het vier variabelen model

Sensoren Software Actuatoren Omgeving



M - Monitored Variable statespace **C** - Controlled Variable statespace
I - Input Variable statespace **O** - Output Variable statespace

[?]

2.3.1 Monitored variabelen

Wat gemeten wordt vanuit de omgeving:

- Temperatuur
- Licht intensiteit
- Luchtvochtigheid
- Wat je allemaal met sensoren kunt meten

2.3.2 Controlled variabelen

Kunnen worden "bestuurd"door actuatoren:

- Temperatuur
- Licht intensiteit
- Wat je allemaal kan beinvloeden

2.3.3 Input variabelen

De data, die staan voor de gemeten waardes vanuit de omgeving, die als input door de software worden gebruikt.

2.3.4 Output variabelen

De data die de software levert als output. Waar de actuatoren op moeten handelen.

2.4 Rampen

2.4.1 Ramp 1

Beschrijving

Datum en plaats

Oorzaak

2.4.2 Ramp 2

Beschrijving

Datum en plaats

Oorzaak

2.4.3 Ramp 3

Beschrijving

Datum en plaats

Oorzaak

2.4.4 Ramp 4

2.4.5 Ramp 5

2.4.6 Ramp 6

3 Modellen

Een goed model heeft een duidelijk object dat gemodelleerd moet worden, er is duidelijk **wat** er beschreven moet worden.
Een goed model heeft een duidelijk doel. -waarom modelleren we? (voor communicatie of verificatie, analyse, etc.)
Een goed model is traceerbaar: elk onderdeel is te herleiden tot de onderdelen van het ëchte"systeem.
Een goed model is waarheidsgetrouw: relevante onderdelebn van het model komen

terug in de werkelijkheid.
een goed model is eenvoudig, maar niet te eenvoudig
Een goed model is uitbreidbaar en herbruikbaar: in de toekomst is het eenvoudig verder te werken met dit model en kunnen zelfs *klassen* van vergelijkbare systemen gemaakt worden
Een goed model deelt geen jargon/semantiek met andere documenten en modellen.

Richtlijnen (tegenstrijdig heden:
Waarheidgetrouw vs simpelheid duidelijheid vs. gedeeld jargon/semantiek

3.1 De Kripke structuur

3.2 Soorten modellen

3.3 Tijd

3.4 Guards en invarianten

Guards zijn voorwaarden waaraan moet worden voldaan voordat een status kan worden gemaakt.

3.5 Deadlock

3.6 Zeno gedrag

4 Logica

4.1 Propositielogica

4.2 Predicatenlogica

4.3 Kwantoren

4.4 Dualiteiten

5 Computation tree logic

5.1 De computation tree

5.2 Operator: AG

De betekenis van AG is makkelijk te onthouden A = Always, G = Globally. Dit houdt in dat het niet uit maakt waar je bent, je zal altijd van welke positie dan ook bij een gedefinieerd punt uitkomen.

5.3 Operator: EG

De betekenis van AG is makkelijk te onthouden E = Exists, G = Globally.

5.4 Operator: AF

De betekenis van AG is makkelijk te onthouden A = Always, F = Eventually.

5.5 Operator: EF

5.6 Operator: AX

5.7 Operator: EX

5.8 Operator: p U q

5.9 Operator: p R q

5.10 Fairness

5.11 Liveness