

Tinlab advanced algorithms

W. Oele

6 februari 2020

Advanced algorithms: inhoud

- Systemen: wat zijn dat?
 - Context: Requirements vs. specifications
 - World vs. machine
- Vier variabelen model
- Modelleren:
 - labeled state transition diagrams
 - Kripke structuren
 - Uppaal
- Verifiëren:
 - temporele logica
 - computation tree en computation tree logic

Programma: week 1/2

Maandag:

- introductie en sanity check
- tools klaarzetten
- theorie: requirements, specificaties en het 4 variabelen model
- toetsing:
 - verslag
 - eindopdracht

Vrijdag:

- herhaling
- oefenen
- werken aan verslag

Programma: week 2/3

Maandag:

- modelleren: soorten systemen
- modelleren: Kripke structuren
 - state
 - invarianten
 - guards
- introductie Uppaal

Vrijdag:

- herhaling
- oefenen
- werken aan verslag

Programma: week 3/4

Maandag:

- Kripke structuren
 - model criteria
 - modelleren: omgaan met tijd
 - Zeno behaviour
 - deadlock

Vrijdag:

- Uppaal
- oefenen
- werken aan verslag

Programma: week 4/5

Maandag:

- Kripke structuren
 - parallele compositie
 - synchronisatie
 - Uppaal

Vrijdag:

- Uppaal
- oefenen
- werken aan verslag

Programma: week 5/6

Maandag:

- computation trees
- computation tree logic
- explicit model checking/verificatie

Vrijdag:

- ctl: liveness
- ctl: fairness/freedom of starvation
- ctl: Uppaal
- oefenen
- werken aan verslag

Programma: week 6/7

Maandag:

- herhaling
- rampenbestrijding
- introductie eindopdracht
- werken aan verslag

Vrijdag:

- herhaling
- rampenbestrijding
- werken aan eindopdracht
- werken aan verslag

Programma: week 7/8

- uitloop
- herhaling
- rampenbestrijding
- werken aan eindopdracht en verslag

Programma: week 8

- presentaties en verdediging eindopdracht

Voorkennis en vaardigheden

Voorkennis:

- propositielogica
- predicaatenlogica
- verzamelingenleer
- grafentheorie

Vaardigheden:

- werken met \LaTeX
- werken met git

Leermiddelen

Voorkennis:

- reader logica
- reader theoretische informatica
- korte inleiding verzamelingenleer
- Schaum's outlines: discrete mathematics

Boek:

- *Model checking*, Edmund. M. Clarke , Orna Grumberg, Doron A. Peled
- diverse wetenschappelijke artikelen

Toetsing

Persoonlijk verslag:

- telt mee voor eindbeoordeling
- individueel
- wordt iedere week aan gewerkt
- wordt regelmatig *bekeken*...
- wordt beoordeeld met voldoende/onvoldoende

Verslag en presentatie eindopdracht:

- telt mee voor eindbeoordeling
- project beoordelingssystematiek

Het persoonlijk verslag

Het verslag:

- schrijf je individueel
- in \LaTeX
- bevat uitleg, voorbeelden en uitwerkingen van oefeningen
- bevat referenties naar wetenschappelijke artikelen, blogs, etc.
- doelgroep: medestudenten, docenten en \neg digibeten
- versiebeheer: git
- baseer je op de template

Het persoonlijk verslag

- Het verslag vormt een uitgebreide “samenvatting” van de leerstof.
- Als je je verslag een jaar later terugleest, moet je zonder problemen je weg kunnen vinden in het onderwerp verificatie en model checking.
- *“Je schrijft je eigen leerboekje”*

Het persoonlijk verslag

Toegestaan en aangeraden:

- “samenwerken”
- oefenen
- de kunst afkijken
- elkaars teksten lezen en corrigeren
- elkaar uitleggen
- \LaTeX constructies overnemen

Harde voorwaarde: Je schrijft *zelf*.

De eindopdracht

De eindopdracht:

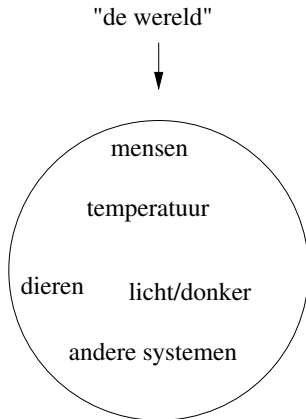
- maak je in groepjes
- bevat alle onderdelen van de project beoordelingssystematiek
- wordt tijdens een presentatie verdedigd

Beoordeling vindt plaats op basis van het verslag en de presentatie.

Deze les

- context: world vs. machine
- requirements en specifications
- systems engineering vs. software engineering
- het vier variabelen model

Context



Context

Van Lamsweerde/Jackson:

- *The problem world is some problematic part of the real world that we want to improve by building some machine solution.*

Context

Van Lamsweerde/Jackson:

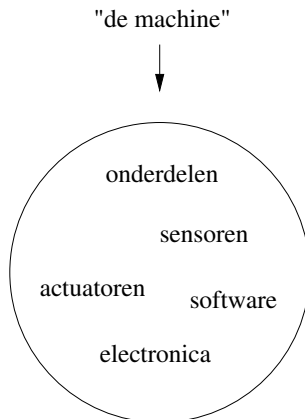
- *The problem world is some problematic part of the real world that we want to improve by building some machine solution.*
- *The aim of a software development project is to improve this world by building some machine expected to solve the recognized problem.*

Context

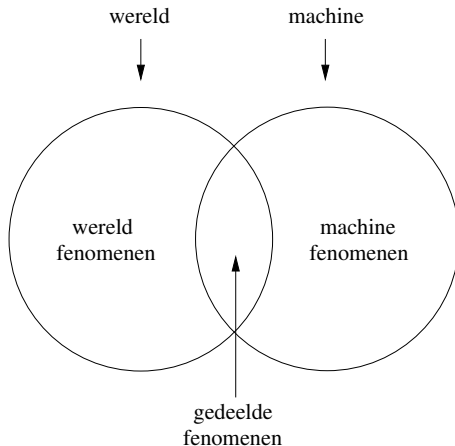
Van Lamsweerde/Jackson:

- *The problem world is some problematic part of the real world that we want to improve by building some machine solution.*
- *The aim of a software development project is to improve this world by building some machine expected to solve the recognized problem.*
- *The requirements engineering process is concerned with the machines effect on the surrounding world and the assumptions we make about this world.*

Context



Context



Requirements vs. specificaties

System requirement:

- uitspraak over wereld fenomenen (gedeeld of niet) of *doelen* die bereikt moeten worden.

Requirements vs. specificaties

System requirement:

- uitspraak over wereld fenomenen (gedeeld of niet) of *doelen* die bereikt moeten worden.
- met enige regelmaat informeel, niet precies geformuleerd.

Requirements vs. specificaties

System requirement:

- uitspraak over wereld fenomenen (gedeeld of niet) of *doelen* die bereikt moeten worden.
- met enige regelmaat informeel, niet precies geformuleerd.

Software requirement/specificatie:

- 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.

Requirements vs. specificaties

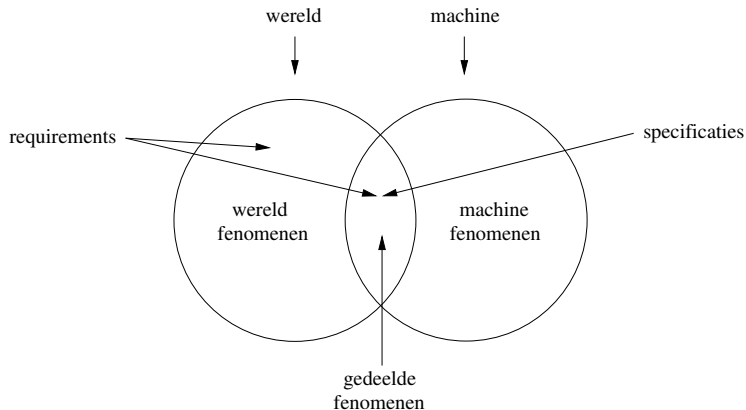
System requirement:

- uitspraak over wereld fenomenen (gedeeld of niet) of *doelen* die bereikt moeten worden.
- met enige regelmaat informeel, niet precies geformuleerd.

Software requirement/specificatie:

- 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.
- doorgaans preciezer, meetbaar, exact geformuleerd.

Requirements vs. specificaties



System engineering vs. software engineering

Systems engineering:

- *Hoe verkrijgen we het juiste systeem?*
- lastige vraag: de buitenwereld geeft niets van zichzelf prijs.

System engineering vs. software engineering

Systems engineering:

- *Hoe verkrijgen we het juiste systeem?*
- lastige vraag: de buitenwereld geeft niets van zichzelf prijs.

Software engineering:

- *Hoe krijgen we de software op de juiste manier in het systeem?*

Software wordt in de systems engineering beschouwd als een onderdeel van het systeem.

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

- Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.)

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

- Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.)
- actuatoren: veranderen iets in de omgeving (mechanische, electrisch, pneumatisch, etc.)

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

- Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.)
- actuatoren: veranderen iets in de omgeving (mechanische, electrisch, pneumatisch, etc.)

Software:

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

- Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.)
- actuatoren: veranderen iets in de omgeving (mechanische, electrisch, pneumatisch, etc.)

Software:

- Kan niet direct communiceren met de buitenwereld.

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

- Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.)
- actuatoren: veranderen iets in de omgeving (mechanische, electrisch, pneumatisch, etc.)

Software:

- Kan niet direct communiceren met de buitenwereld.
- Snapt derhalve niets van de buitenwereld.

Systemen, requirements en specificaties

Systemen gaan een zekere interactie aan met hun omgeving:

- Sensoren: meten fenomenen uit de omgeving (temperatuur, druk, licht, geluid, etc.)
- actuatoren: veranderen iets in de omgeving (mechanische, electrisch, pneumatisch, etc.)

Software:

- Kan niet direct communiceren met de buitenwereld.
- Snapt derhalve niets van de buitenwereld.
- Kan alleen maar bestaan in en communiceren met het systeem.

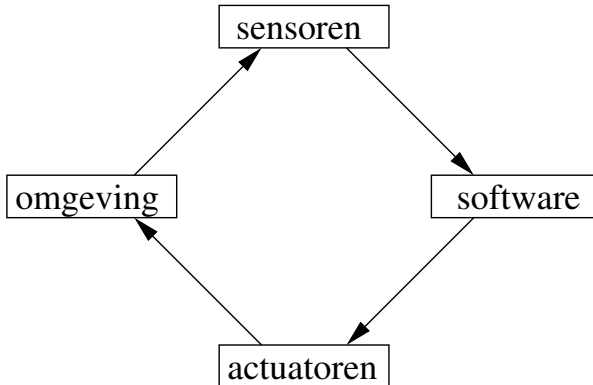
Systemen, requirements en specificaties

De verschillen tussen:

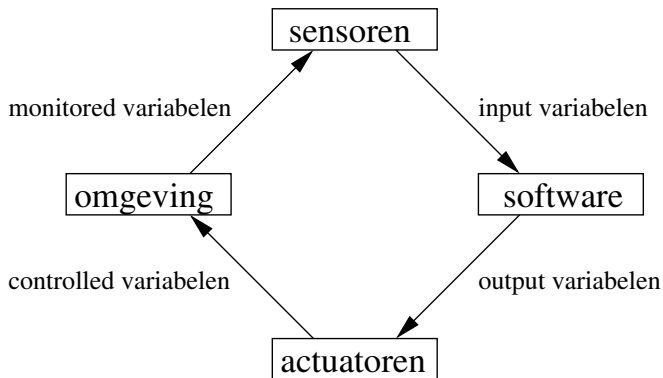
- de buitenwereld en het systeem in de buitenwereld
- requirements en specifications
- systeem en de software *in* dat systeem

kunnen we preciezer formuleren middels het *vier variabelen model* (Parnas & Madey, 1995)

Het vier variabelen model



Het vier variabelen model



Het vier variabelen model

Systemen (met daarin software) en de bijbehorende vier variabelen:

- Monitored variabelen: door sensoren gekwantificeerde fenomenen uit de omgeving

Het vier variabelen model

Systemen (met daarin software) en de bijbehorende vier variabelen:

- Monitored variabelen: door sensoren gekwantificeerde fenomenen uit de omgeving
- Controlled variabelen: door actuatoren “bestuurde” fenomenen uit de omgeving

Het vier variabelen model

Systemen (met daarin software) en de bijbehorende vier variabelen:

- Monitored variabelen: door sensoren gekwantificeerde fenomenen uit de omgeving
- Controlled variabelen: door actuatoren “bestuurde” fenomenen uit de omgeving
- Input variabelen: data die de software als input gebruikt

Het vier variabelen model

Systemen (met daarin software) en de bijbehorende vier variabelen:

- Monitored variabelen: door sensoren gekwantificeerde fenomenen uit de omgeving
- Controlled variabelen: door actuatoren “bestuurde” fenomenen uit de omgeving
- Input variabelen: data die de software als input gebruikt
- Output variabelen: data die de software levert als output

System en software requirements

System requirements definiëren een relatie tussen de monitored variabelen en de controlled variabelen:

System en software requirements

System requirements definiëren een relatie tussen de monitored variabelen en de controlled variabelen:

$$Sysreq \subseteq M \times C$$

System en software requirements

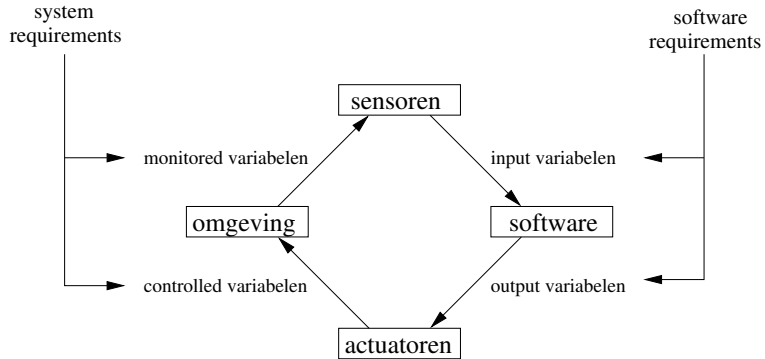
System requirements definiëren een relatie tussen de monitored variabelen en de controlled variabelen:

$$Sysreq \subseteq M \times C$$

Software requirements/specificaties definiëren een relatie tussen de input en output variabelen:

$$Softreq \subseteq I \times O$$

System en software requirements



Artikelen

- The world and the machine - M. Jackson
- Therac-25 and the security of the computer controlled equipment - E. Bozdag
- Requirements - A. van Lamsweerde
- What are requirements? - S. Easterbrook

Vragen voor het verslag

- Wat is het verschil tussen system- en softwareengineering?
- Wat zijn requirements en specifications?
- Er bestaat ook een 6-variabelen model... Wat is dat?
- Wat voor soorten requirements zijn er zoal te vinden?
- Hoe verkrijgt men requirements?
- Wat voor requirement elicitation technieken zijn er zoal?
- Wat is het verschil tussen functionele en niet-functionele requirements?
- Wat verstaat men onder *mode confusion*?
- Wat verstaat men onder de automatiseringsparadox?

Vragen voor het verslag

Zoek uit:

- wat er mis ging met de Therac
- wat er mis ging tijdens vlucht 1951
- wat de kernramp in Tsjernobyl in 1986 veroorzaakte

Zoek naast bovenstaande rampen nog drie gevallen uit, waarin een systeem faalde en beoordeel deze volgens het onderscheid

- hardware/software/gebruik
- 4 variabelen model
- requirements vs. specifications