HOGESCHOOL ROTTERDAM

PEE30

ER3A, ER3B

Vorige les

- Je *kan* alles als een systeem zien.
- Welke eigenschappen heeft elk systeem?
  - Functie
  - Interface (systeem bevindt zich in supersysteem)
  - Implementatie (systeem bestaat uit subsystemen)
- 5 Basisprincipes van systeemtheorie:
  - Specifieker ➔ Slechter aan te passen
  - Groter ➔ Meer onderhoud
  - Hiërarchie (van supersystemen, systemen en subsystemen)
  - Veranderen in de tijd (implementatie veranderd sneller dan functie)
  - Interactie is complex en subtiel

Vorige les

- Wat is een model?
  - Abstractie met bepaald doel, meestal schematisch (plaatje + tekst)
- Stappenplan modelleren systeem:
  - **Wat?** (functie)
  - **Waarmee?** (interface)
  - **Hoe?** (implementatie)

Vorige les

- We modelleren de functie en implementatie van het systeem apart.
- Met vier elementen kunnen we het functionele model bouwen.
  - Proces
  - Data Flow
  - Data Store
  - Terminator
- Het data context diagram modelleert de interface en de belangrijkste functie van een systeem.

Cursushandleiding

Zij er vragen over:

- Competenties?
- Leerdoelen?
- Toetsing en beoordeling?
- Verplichte literatuur?
- Deadlines en speciale Activiteiten?
- Planning?

HET FUNCTIONELE ONTWERP MET DATA FLOW DIAGRAMMEN

Overzicht systeemmodel

Functional requirements model:

Wat doet het systeem?

Waarmee doet het systeem dat?

1. Teken het data context diagram (DCD)
2. **Teken de onderliggende data flow diagrammen (DFD)**
3. Vul de data dictionary (DD)
4. Stel de proces specificaties op (PSPEC)
5. Onderscheid de control processen en control flows
6. Stel de control specificaties op (CSPEC)

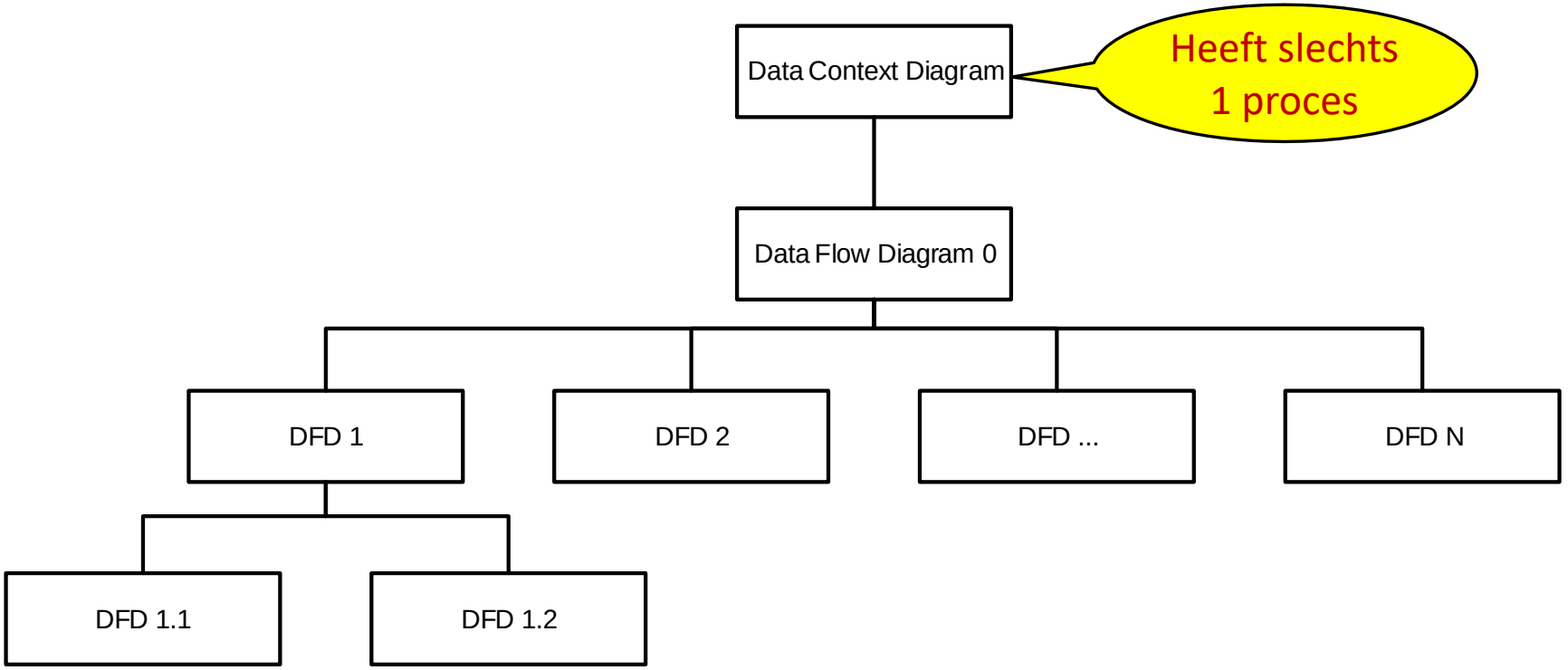
Architectural model:

Hoe doet het systeem dat?

1. Teken het architecture context diagram (ACD)
2. Teken het architecture interconnect diagram (AID)

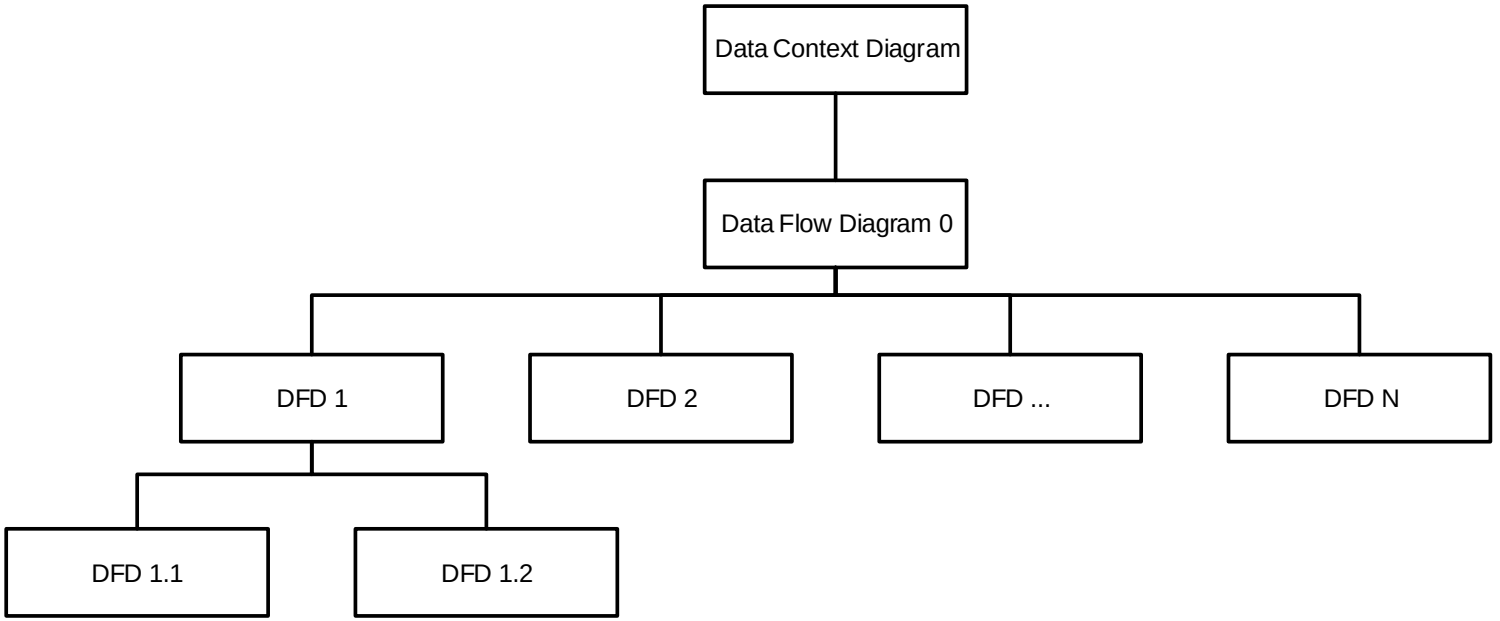
Hiërarchie in Data Flow Diagrammen

- Hiërarchisch overzicht van het systeem en omgeving.
- Onderliggende processen worden genummerd.



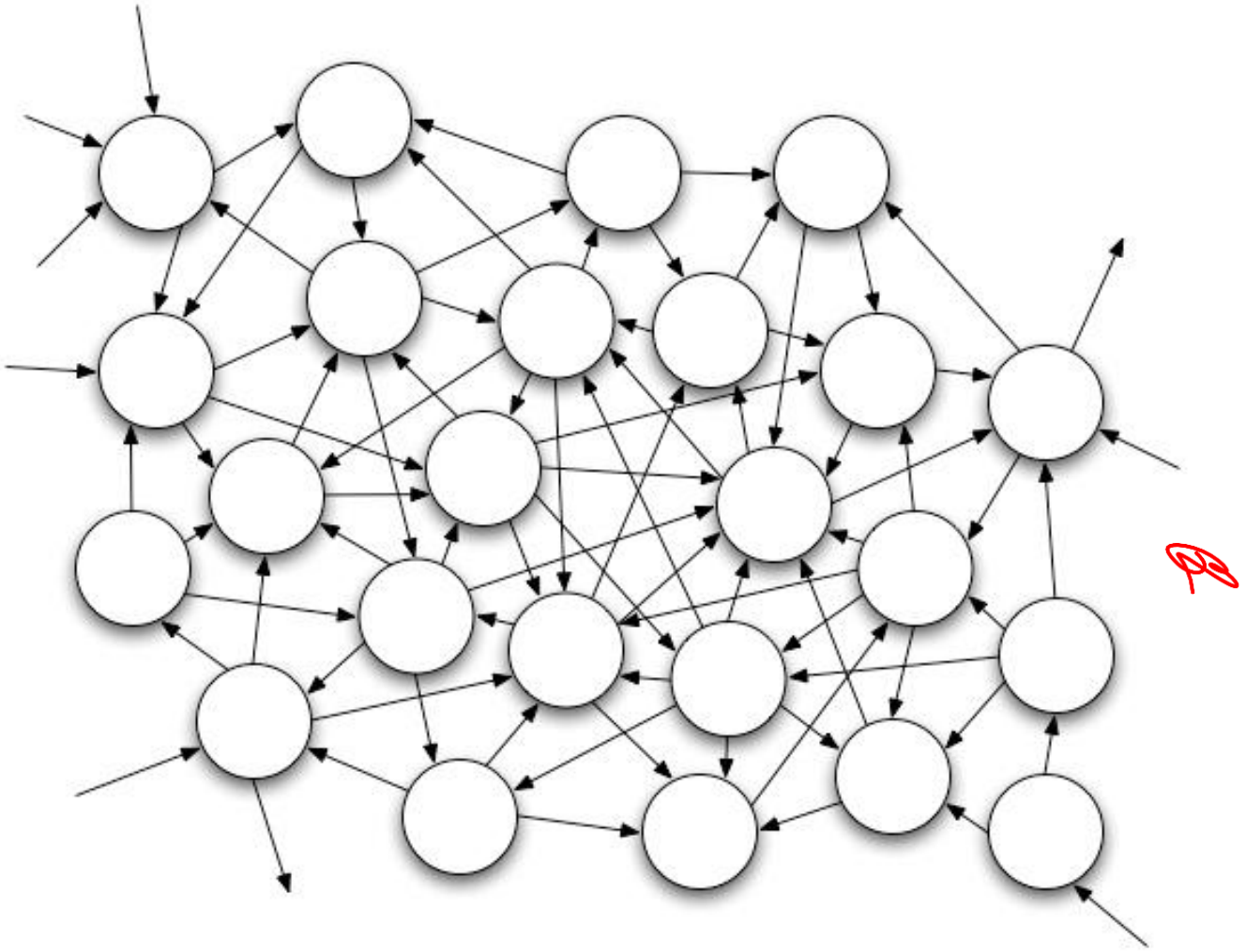
Te beantwoorden vragen:

- Hoeveel levels moet je systeem krijgen?
- Hoeveel processen moet een level hebben?
- Moeten alle processen evenveel onderliggende levels krijgen?
- Is het model logisch consistent?



Hoeveel levels?

- Tot je dit niet meer ziet:



Hoeveel processen moet een level hebben?

- Maximaal 9
  - 7 +/- 2 regel (Miller's Law)

- Minstens 1 (context diagram)
  - Voor lagere DFD's minstens 2, want anders heeft extra laag geen zin.

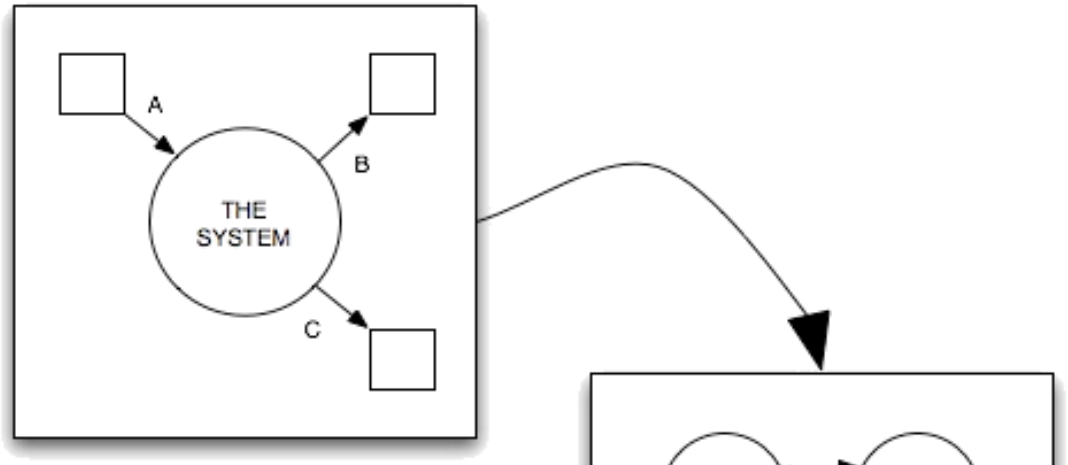
Aantal onderliggende levels gelijk?

- Nee
  - Processen kunnen complex of eenvoudig zijn, dus hoeft niet.
  - Te veel onbalans is een reden om splitsing op hogere niveaus te heroverwegen.

Logisch consistent

- Dataflow die binnenkomt in een proces moet het onderliggende DFD van dat proces binnenkomen.
- Dataflow die een DFD binnenkomt moet ook in het bovenliggende proces binnenkomen.
- Dataflow die een proces verlaat moet ook het onderliggende DFD van dat proces verlaten.
- Dataflow die een DFD verlaat moet ook het bovenliggende proces verlaten.

Is dit model logisch consistent?



CONTEXT
DIAGRAM

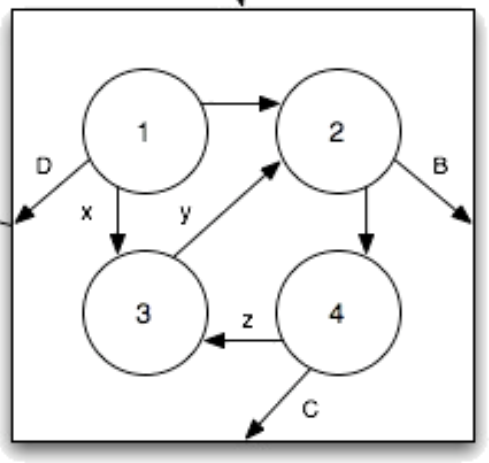


FIGURE 0

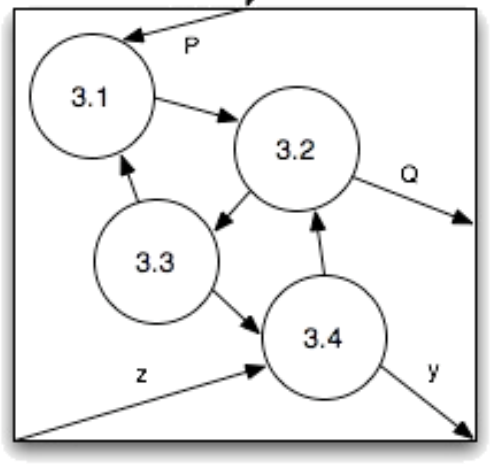
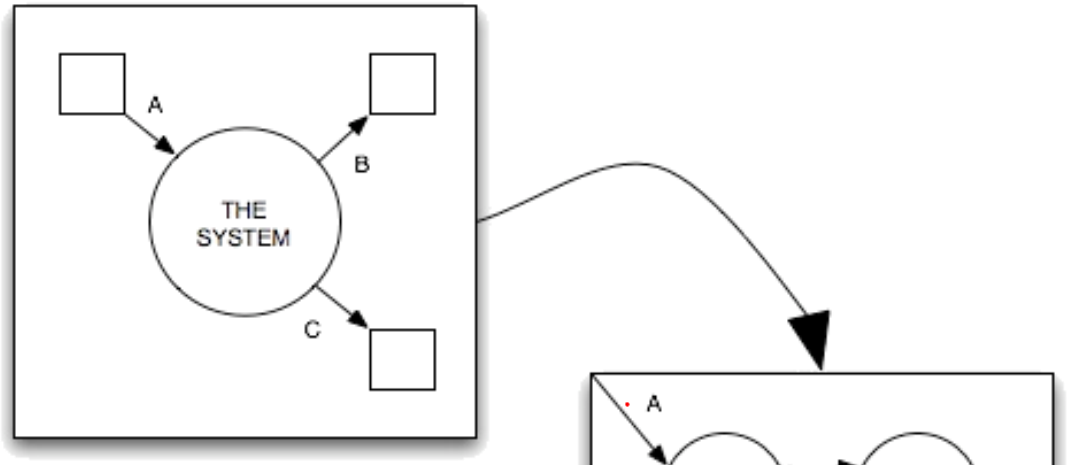


FIGURE 3

Is dit model logisch consistent?



CONTEXT
DIAGRAM

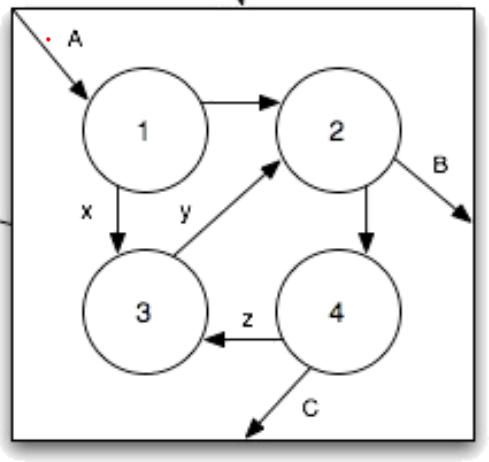


FIGURE 0

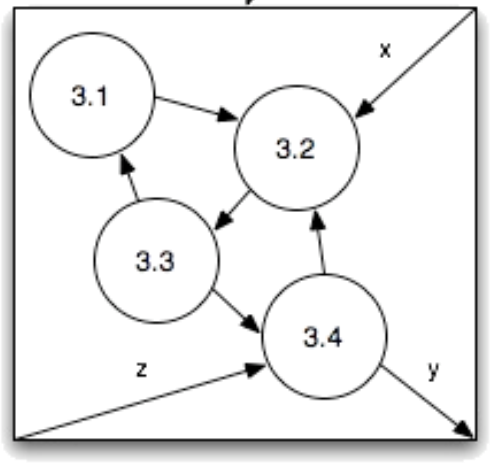
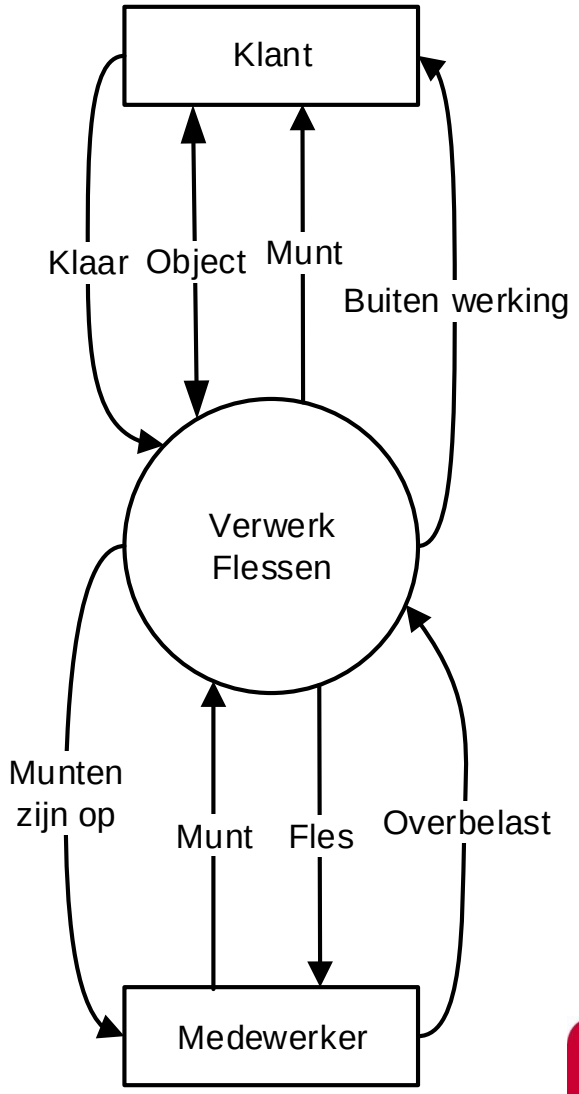


FIGURE 3

HOGESCHOOL ROTTERDAM

Flessenverwerker

- Welke in- en uitgaande flows heeft het dataflowdiagram DFD0?



Flessenverwerker

De flessenverwerker moet een systeem worden waarbij klanten flessen met statiegeld in kunnen voeren en daarvoor een tegoed aan geld terug krijgen.

In feite kunnen klanten allerlei objecten invoeren. Alleen als het object een fles van waarde is, wordt het object verplaatst naar de andere kant van de verwerker (mag niet door de klant te bereiken zijn) waar een medewerker de fles uit het systeem kan halen.

Indien het object een fles van waarde is wordt ook een tegoed (afhankelijk van de prijs van de fles) geaccumuleerd voor de klant. Indien het object geen fles van waarde is wordt het object teruggeven aan de klant. Uiteindelijk kan de klant aangeven dat hij/zij klaar is met het invoeren van objecten, om zo het tegoed uitgekeerd te krijgen in geld (munten).

Als de medewerkers overbelast zijn moeten ze dit aan kunnen geven, op dit moment kunnen er geen nieuwe objecten worden ingevoerd. Ook indien er geen geld meer in het systeem zit, moet dit eerst worden aangevuld alvorens het systeem nieuwe objecten accepteert.

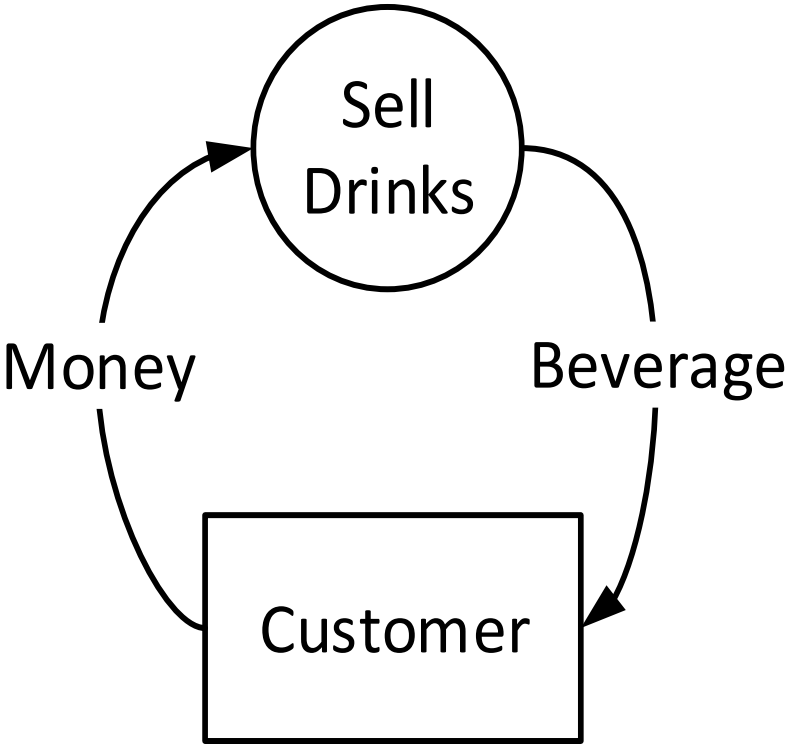
TOOLS VOOR HET MAKEN VAN DATA FLOW DIAGRAMMEN

Tools voor het maken van DFD

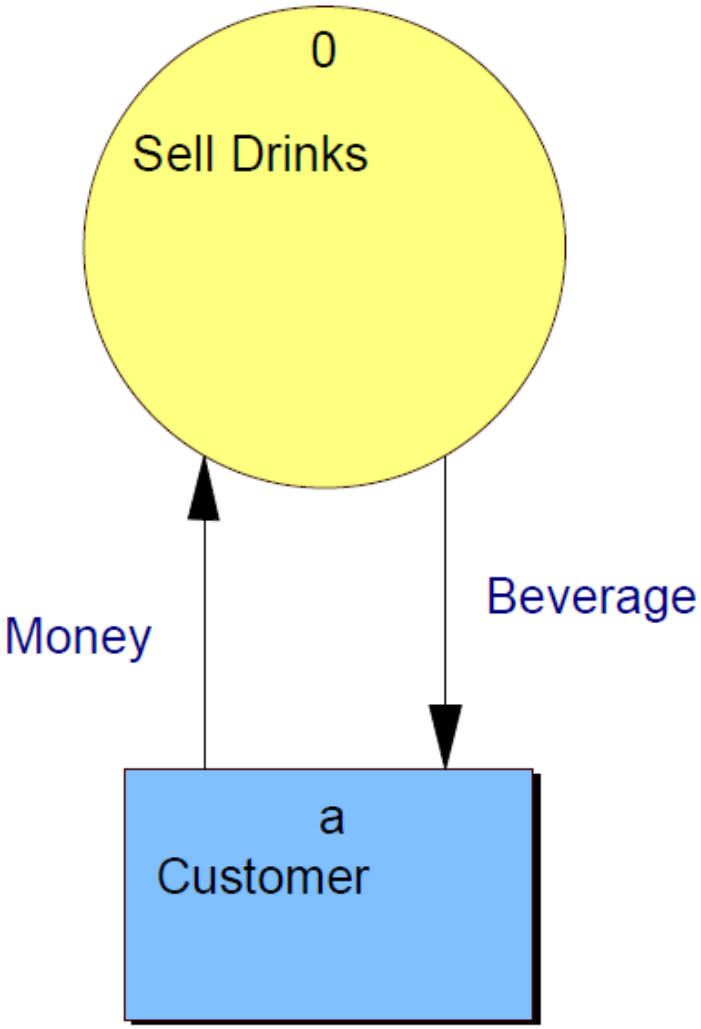
- Tekentools:
  - Microsoft Visio (gratis beschikbaar op HR)
  - Dia Diagram Editor (free: http://dia-installer.de/)
  - Draw.io of Lucidchart
- CASE-tools (Computer-aided software engineering):
  - Select Yourdon (Single License US $1898): http://www.selectbs.com/analysis-and-design/select-yourdon
  - Enterprise Architect (Single License US $599): http://sparxsystems.com/enterprise_architect_user_guide/12.1/business_engineering/analysis_models2.html
  - **QSEE multi-CASE tool** (free): http://www.leedsbeckett.ac.uk/qsee/
  - ...

Een CASE-tool kan je DFD checken!

Voorbeeld drankautomaat



Voorbeeld drankautomaat



QSEE Technologies is part of

LEEDS BECKETT UNIVERSITY

HOGESCHOOL ROTTERDAM

Overzicht systeemmodel

Functional requirements model:

Wat doet het systeem?

Waarmee doet het systeem dat?

1. Teken het data context diagram (DCD)
2. Teken de onderliggende data flow diagrammen (DFD)
3. **Vul de data dictionary (DD)**
4. **Stel de proces specificaties op (PSPEC)**
5. Onderscheid de control processen en control flows
6. Stel de control specificaties op (CSPEC)

Architectural model:

Hoe doet het systeem dat?

1. Teken het architecture context diagram (ACD)
2. Teken het architecture interconnect diagram (AID)

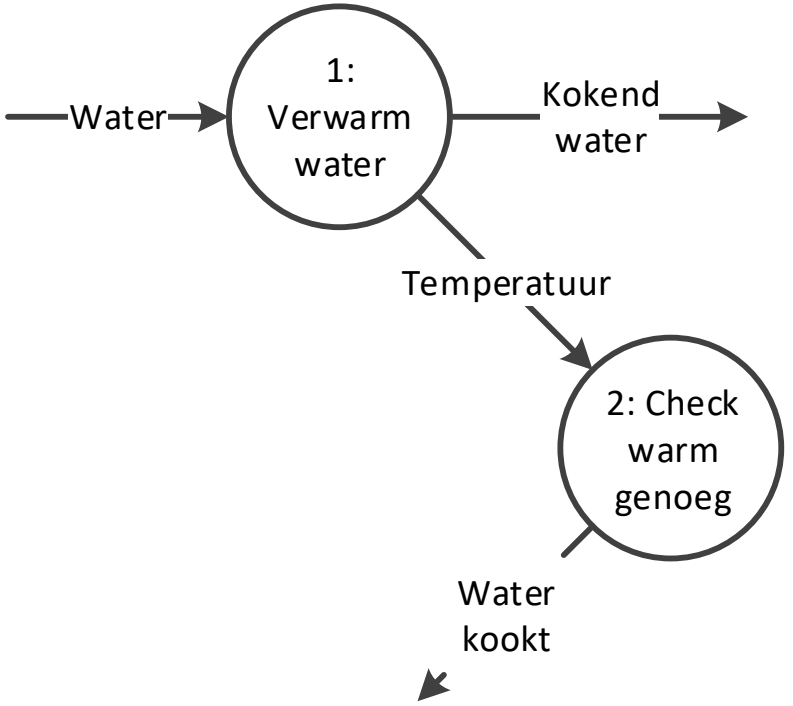
DATA DICTIONARY

Data dictionary (zie Yourdon H10)

- Korte beschrijvingen van **alle** flows en stores:

Data flow “Temperatuur”	
Deze flow geeft de temperatuur weer.	
Grootheid:	Temperatuur
Eenheid:	˚C
Bereik:	0 - 120

Kan op verschillende manieren, dit is slechts een voorbeeld.



composition of aggregate data

Data dictionary notation (Yourdon 10.2):

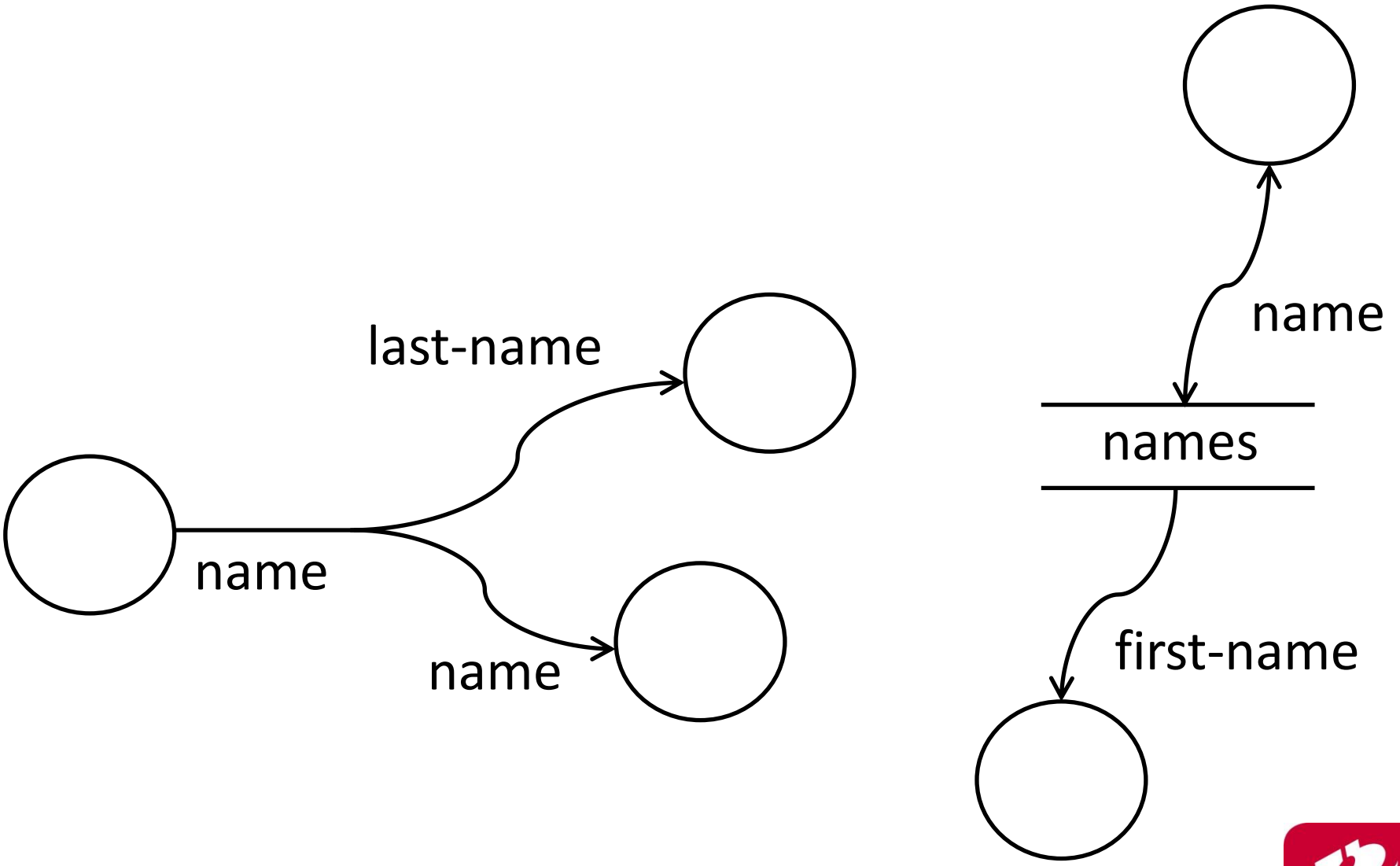
- = is composed of
- + and
- ( ) optional (may be present or absent)
- { } iteration: (min){ }(max)
- [ ] select one of several alternative choices
- | separates alternative choices in the [ ] construct
- ** comment
- @ unique identifier (key field) for a store

Voorbeeld

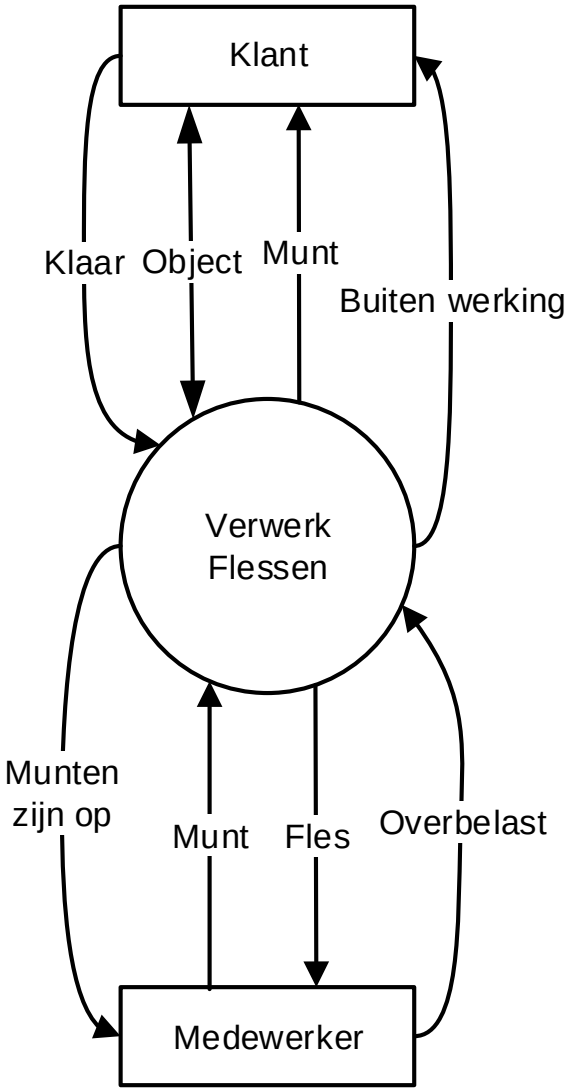
- **name** = **courtesy-title** + **first-name** + (**middle-name**) + **last-name**
- **courtesy-title** = [Mr. | Miss | Mrs. | Ms. | Dr. | Professor]
- **first-name** = 1{**legal-character**}
- **middle-name** = 1{**legal-character**}
- **last-name** = 2{**legal-character**}
- **legal-character** = [A-Z | a-z | 0-9 | - | *space*]

Implementatiegerichte zaken hier niet specificeren!

Samengestelde flow of store kan splitsen



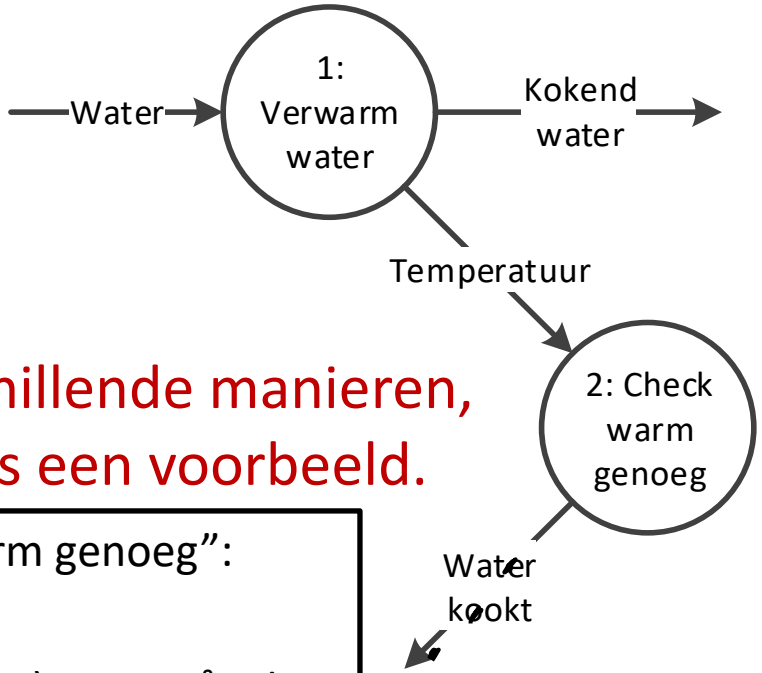
Maak DD Flessenverwerker



PROCESS SPECIFICATIONS

Process specifications (zie Yourdon H11)

- Korte *functionele* beschrijvingen van alle **niet uitgediepte** processen.
- Process Specification: PSPEC



PSPEC "Verwarm Water":

Dit proces neemt IN(Water) op en verwarmt het water zodanig dat na enkele tijd OUT(Kokend water) kan worden uitgevoerd. Tevens geeft het proces de OUT(Temperatuur) weer van het water.

Kan op verschillende manieren, dit is slechts een voorbeeld.

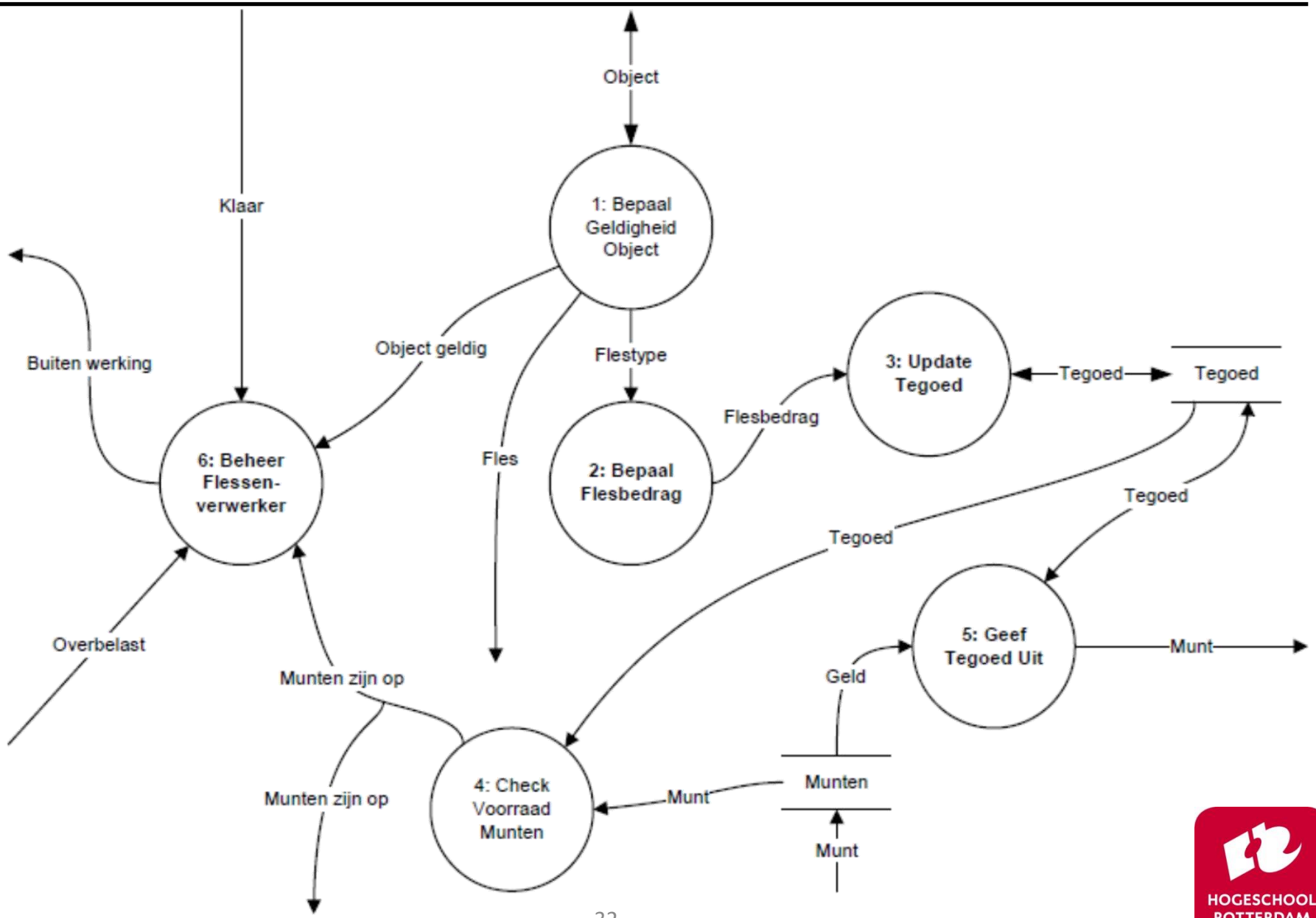
PSPEC "Check warm genoeg":

Als IN(Temperatuur) >= 100 ˚C dan
    OUT(Water kookt) = waar
Anders
    OUT(Water kookt) = onwaar

PSPEC

- Beschrijft **wat** een proces doet (om de inputs naar de outputs te transformeren). *Laat implementatie (hoe) buiten beschouwing!*
- Vele vormen mogelijk:
  - Wiskundige formules, bijvoorbeeld regeltechniek: *H(s)* = …
  - Grafiek, bijvoorbeeld overdrachtskarakteristiek
  - Taal, bijvoorbeeld Structured English of Nederlands
  - Pre- en postcondities
  - Beslissings- of waarheidstabel
  - Toestandsmachine (state machine)
  - Flowcharts
  - …
- Kies per proces een geschikte vorm.

Maak DD en PSPECs Flessenverwerker



PROJECTOPDRACHT PEE30

De opdrachtgever is een fervent fietser en dol op gadgets. Hij zou graag zo veel mogelijk willen meten aan en om zijn fiets. Denk hierbij iig aan omgevingstemperatuur, cadans, snelheid, lichtsterkte, vermogen. Eventuele extra te meten parameters zijn hartslag, VO2Max, bandendruk, remkracht, etc. Niet alleen dienen deze waarden gemeten worden, ze moeten ook weergegeven worden en opgeslagen worden om later terug te kijken.

Tevens dient een servicemonteur in staat zijn om waarden te kalibreren en een coach dient inzicht te hebben in de prestaties van de fietser.

Tijdens het fietsen wil de opdrachtgever zoveel mogelijk geautomatiseerd laten afhandelen. Denk hierbij iig aan het licht automatisch aan en uit laten gaan, het achterlicht laten knipperen als er hard geremd wordt. Wanneer de bandendruk onder een bepaalde waarde komt dient er een waarschuwing te komen.

Als de fietser zijn gemiddelde snelheid en of vermogen hoger liggen dan eerdere gemiddelden wordt deze hiervan op de hoogte gesteld, tevens als deze lager ligt.

Al deze waarden dienen verstuurd te worden naar de cloud waar dit wordt opgeslagen, dit kan alleen als er een verbinding is met het internet. Deze verbinding kan opgesteld worden via WiFi of via Bluetooth richting een GSM.

PvE

Model = DCD + DFD0 + ({DFD…}) + DD + 2{PSPEC} + ({CSPEC})

PvE = 1{Requirement}

Eisen aan eisen

- Een eis (requirement) moet … zijn:
  - ondeelbaar (atomic)
  - duidelijk (beknopt, eenvoudig en nauwkeurig) (clear)
  - juist (correct)
  - abstract (implementation-free)
  - onafhankelijk (independent)
  - haalbaar (feasible)
  - noodzakelijk (necessary)
  - testbaar (testable, verifiable)
  - traceerbaar (traceable)
  - eenduidig (unambiguous)
  - begrijpelijk (understandable)

Zie wiki good_requirements.pdf voor uitleg en voorbeelden!

Eisen aan PvE

- Een PvE moet … zijn:
  - compleet
  - consistent
  - niet redundant

- Hoe helpt het Yourdon model bij het opstellen van het PvE.
  - Functionaliteit is helder voordat opstellen PvE begint.
  - Elk proces (functie) met flows naar buiten levert requirements (zorgt voor compleetheid).
  - DD zorgt voor consistentie.

Opstellen eisen

- Het Yourdon functional requirements model definieert de functies die het systeem moet vervullen.
  - Voeg hier prestatie-eisen aan toe.
  - Voeg daarna niet functionele eisen toe.

- Schrijf meteen de acceptatietest.
  - Dit zorgt ervoor dat eisen testbaar zijn.

NAZORG LES 2

Zelfstudie

- Bestuderen Yourdon: H9 (t/m 9.3), H10 & 11
- Bestuderen DCD’s van System of Systems