

# Problem solving aspecten bij de constructie van content models

---

Auteur:

**Bas Vossen**  
**S0813656**

Begeleider:

**prof. dr. E. Barendsen**

Datum:

**27-09-2012**

## INHOUDSOPGAVE

---

Probleemstelling .....	3
Theoretisch kader .....	4
Problemen .....	4
Verschillende soorten problemen .....	4
Probleemoplossing .....	6
Strategieën voor probleemoplossing .....	7
Schema activatie .....	7
Search-based problem solving .....	8
Analogie .....	9
Problem Solving cycli .....	10
Cognitive-Metacognitive Framework .....	10
Plan-Do-Check-Act Cyclus .....	11
Modelleren .....	13
Meten van effectiviteit .....	14
Context .....	15
Methode .....	19
Toepassen van de methode .....	19
Search-Based Problem Solving .....	20
Cognitive-Metacognitive Framework .....	20
Plan-Do-Check-Act Cyclus .....	21
Betrouwbaarheid .....	22
Resultaten .....	23
Toepassen van de methode .....	23
Search-Based Problem Solving Vs. Schema Activation .....	24
Cognitive-Metacognitive Framework .....	24
Plan-Do-Check-Act cyclus .....	26
Betrouwbaarheid .....	27
Conclusie en discussie .....	28
Bibliografie .....	31
Boeken .....	31
Artikelen .....	31
Appendix .....	32
Toepassen van de methode .....	32
Posttest proefpersoon 1 .....	32
Posttest Proefpersoon 2 .....	34
Cognitive-Metacognitive Framework .....	35

Pretest Proefpersoon 1 .....	35
Posttest Proefpersoon 1 .....	39
Pretest Proefpersoon 2 .....	40
Posttest Proefpersoon 2 .....	44
PDCA Cyclus.....	45
Pretest Proefpersoon 1 .....	45
Posttest Proefpersoon 1 .....	49
Pretest Proefpersoon 2 .....	50
Posttest Proefpersoon 2 .....	54

## PROBLEEMSTELLING

---

In juni 2011 hebben Harrie Passier en Bastiaan Heeren, beide docenten aan de Open Universiteit Nederland, een artikel gepubliceerd over een zelf opgestelde systematische aanpak voor het modelleren van XML content models gebaseerd op het herschrijven van reguliere expressies. De reden voor het opstellen van deze nieuwe aanpak (genaamd 'XML: theorie en toepassing') was hun ontevredenheid over het bestaande lesmateriaal: *"Over the years, a lot of course material has been developed to explain to undergraduate students the fundamentals of XML, and schema languages such as DTD and XMLSchema. Typically, the syntax of these languages is discussed and examples are given. How to find a schema for some XML content is often not covered by the material. As a result, students have problems to start with modeling a complex schema, many of their inferred XML schemas are too liberal, and some are even incorrect."* (Passier en Heeren, 2011). Passier en Heeren hebben, om deze methode te evalueren, een experiment uitgevoerd: *"Five students have participated in a small experiment consisting of a pretest, an online lecture of two hours, a posttest, and an interview. (...)All students scored one or two points higher on the posttest with respect to the pretest; the mean score increased from 4.6 to 6.0, where the maximum score was nine points."*

In het eerste opzicht ziet dit er positief uit. Toch lijkt deze vorm van evalueren niet waterdicht. Zoals in het artikel van Passier en Heeren al bediscussieerd wordt kan er niet worden uitgesloten dat de proefpersonen beter op de posttest scoren door het feit dat ze op dat moment meer in aanraking zijn gekomen met de vaardigheden dan bij de pretest, of dat het echt komt door het succes van de nieuwe methode. Verder kunnen we op deze manier niet beredeneren waarom de methode werkt, gezien er puur wordt gekeken naar de concrete resultaten: maakten de proefpersonen betere modellen na de cursus, behaalden ze betere resultaten? Passier en Heeren zijn op zoek naar antwoord op de volgende onderzoeksvraag:

*Heeft de cursus 'XML: theorie en toepassing' invloed op het denkproces bij fundamentele XML vaardigheden?*

Om deze vraag te kunnen beantwoorden willen ze de volgende aspecten behandelen:

- Wordt de strategie van de cursus toegepast?
- Op welke manier bevordert de cursus het denkproces?

Het doel van deze Bachelorscriptie is een opzet maken voor een grootschaliger onderzoek waarbij er niet alleen wordt gekeken naar de resultaten maar ook naar het denkproces van de proefpersonen. Dit kan wellicht duidelijk maken welke aspecten van de methode belangrijk en effectief zijn. Om dit te verwezenlijken is er in deze scriptie gekeken op welke manier bovenstaand onderzoek het beste uitgevoerd kan worden. Hierbij is natuurlijk belangrijk welke manieren geschikt zijn voor het onderzoek, maar ook of deze betrouwbaar zijn of niet. De onderzoeksvraag van deze scriptie luidt als volgt:

**Op welke manier kan de invloed van de cursus 'XML: theorie en toepassing' op het denkproces bij fundamentele XML vaardigheden in kaart gebracht worden?**

Om deze vraag te kunnen beantwoorden zullen we antwoord geven op de volgende deelvragen:

- Welke manieren zijn geschikt voor de evaluatie van de cursus?
- Zijn deze manieren bruikbaar?
- Zijn deze manieren betrouwbaar?

---

## THEORETISCH KADER

---

---

### PROBLEMEN

---

Iedereen komt elke dag wel in aanraking met problemen, of dit nu grote of kleine zijn: 'Wat zal ik vanmiddag eten?', 'Hoe kan ik het beste de barbecue van volgende week vrijdag organiseren?', 'Hoe ga ik mijn kapotte computer repareren?' enzovoorts. Het proces wat we volgen om deze problemen op te lossen noemen we *problem solving* oftewel probleemoplossing. Om te weten hoe probleemoplossing precies in zijn werk gaat is het goed om te weten wat een probleem nu precies is. Robertson (2001) zegt: "(...) *a distinguishing feature of a problem is that there is a goal to be reached and how you get there is not immediately obvious. To put it at its simplest, you have a problem when you are required to act but you don't know what to do.*". Duncker (1945) geeft een uitgebreidere uitleg van het fenomeen probleem: "*A problem arises when a living creature has a goal but does not know how this goal is to be reached. Whenever one cannot go from the given situation to the desired situation simply by action, then there has to be recourse to thinking. Such thinking has the task of devising some action, which may mediate between the existing and desired situations.*". Duncker geeft dus aan dat een probleem geen probleem is als je precies weet wat je moet doen. Zo weet je precies wat je moet doen om de som '5 + 2' op te lossen en valt dit dus niet onder een probleem.

---

### VERSCHILLENDE SOORTEN PROBLEMEN


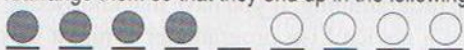
---

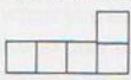
Er zijn verschillende manieren waarop men problemen kan indelen. Robertson (2001) noemt een aantal van deze categorieën. Allereerst kan er gekeken worden naar *knowledge-lean vs. knowledge-rich problems*. Knowledge-lean problems zijn problemen met simpele regels en een duidelijke omschrijving waarbij weinig kennis nodig is om het op te lossen. Denk hierbij bijvoorbeeld aan een Sudoku-puzzel of de Torens van Hanoi puzzel. Knowledge-rich problems zijn problemen waarbij wel veel kennis nodig is om het op te lossen. Denk hierbij bijvoorbeeld aan het maken van een stuk code in C++. Voor programmeurs met veel ervaring met deze programmeertaal zal dit vrij gemakkelijk zijn. Programmeurs met weinig ervaring met C++ zullen er echter moeite mee hebben, en voor mensen zonder programmeerervaring zal de opdracht niet te maken zijn.

Een tweede categorie bevatten de *well-defined vs. ill-defined problems*. Well-defined problems zijn problemen waarbij de beginstaat van het probleem, de doelstaat van het probleem, de operatoren (de stappen die gevolgd moeten worden om het probleem op te lossen) en de restricties duidelijk zijn gegeven. Denk hierbij weer aan een Sudoku-puzzel of de Torens van Hanoi puzzel. Problemen waarbij deze vier zaken niet duidelijk gegeven zijn worden ill-defined problems genoemd. Denk hierbij bijvoorbeeld aan raadsels die in een verhaalvorm gegoten zijn.

Verder kunnen we een onderscheid maken tussen *semantic-lean* vs. *semantic-rich problems*: "Semantics is the study of meanings, and the more you know of a word or concept (the more associations and relations you form) the bigger will its semantic space become. When someone is presented with puzzles such as the Dots problem or a matchstick puzzle (zie Figuur 1) for the first time, the puzzle is *semantically lean* (or *semantically poor*) as far as the solver is concerned; that is, there is very little knowledge or prior experience that the solver can call on to help solve the problem. The same would be true of someone learning to play chess for the first time. The learner does not have a great deal of experience on which to draw. For a chess expert, on the other hand, the game of chess is *semantically rich*. The expert can bring to bear a rich body of knowledge and experience about varieties of opening moves, defensive positions, strategies, thousands of previously encountered patterns of pieces, whole games even.", aldus Robertson (2001).

FIGUUR 1

3. The Dots Problem:  
 Starting with an arrangement of dots given below:  
  
 rearrange them so that they end up in the following arrangement:  
  
 A dot can only move to an empty space adjacent to it. A dot can jump over only ONE dot of either colour. White dots can only move to the right and grey dots to the left.

4. Given 16 matches arranged in 5 squares like this:  
  
 move 3 matches to form 4 squares.

Als laatste noemt Robertson nog de categorie *insight problems*. Dit zijn problemen waarbij een cruciale stap nodig is voor de oplossing. Zodra deze stap genomen is wordt de oplossing direct of binnen korte tijd overduidelijk (ook wel de "'Aha!' or 'Eureka' experience" genoemd).

## PROBLEEMOPLOSSING

---

Zoals eerder al genoemd is problem solving het proces dat we volgen om problemen op te lossen. Robertson definieert problem solving als volgt: "*Problem solving starts off from an initial given situation or statement of a problem (known as the initial state of the problem). Based on the problem situation and your prior knowledge you have to work towards a solution. When you reach it you are in the goal state of the problem. On the way from the initial state to the goal state you pass through a number of intermediate problem states.*". In het algemeen zijn er twee types probleemoplossing, namelijk *strong methods* en *weak methods*. "*Strong methods are domain-specific, learned methods that are pretty much guaranteed to get a solution; that is, strong methods are used when you already know how to go about solving the problem. (...) Weak methods are general-purpose problem-solving strategies that solvers fall back on when they don't know what to do directly to solve the problem.*" (Robertson, 2001).

Er zijn verschillende manieren om te kijken naar probleemoplossing. Zo zijn er verscheidene strategieën voor probleemoplossing bekend die gebruikt worden om het oplossingsproces te bevorderen. Deze strategieën kunnen in kaart worden gebracht om de inhoudelijke stappen die een probleemoplosser maakt zichtbaar te maken. Deze stappen kunnen bewust (bijvoorbeeld na training) maar ook onbewust genomen worden (uit intuïtie). Er is dan ook een verschil in intelligentie waar te nemen tussen deze verschillende strategieën. Experts op het gebied van probleemoplossing zullen intelligente strategieën bewust toepassen terwijl beginners vaak onbewust (en daardoor minder intelligente) strategieën toepassen.

Verder kan er ook gekeken worden naar cycli die voorkomen bij het proces van probleemoplossing. Deze verschillen van de strategieën van probleemoplossing door het feit dat hierbij niet wordt gekeken naar inhoudelijke stappen maar naar de structuur van het oplossingsproces. Door deze structuur in kaart te brengen kan duidelijk worden hoe efficiënt en systematisch het oplossen van een probleem verloopt.

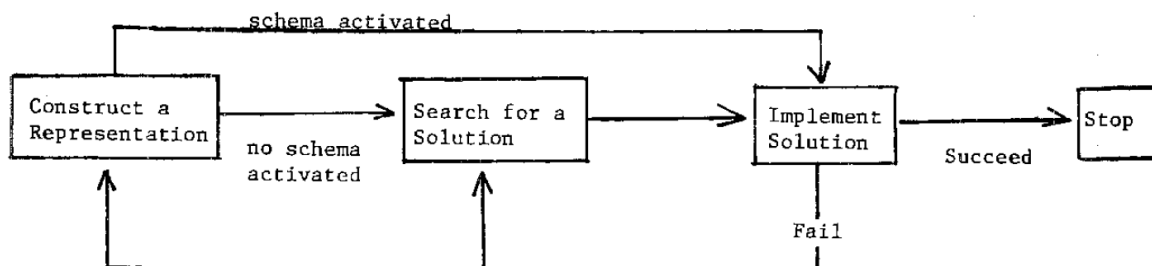
## STRATEGIEËN VOOR PROBLEEMOPLOSSING

Door de grote verscheidenheid aan problemen en een nog grotere verscheidenheid aan mensen bestaat er geen standaard strategie voor probleemoplossing. Zo zal een schaakexpert op een hele andere manier een probleem oplossen dan een stratenmaker. Het is dan ook logisch dat er verschillende soorten strategieën door verschillende mensen worden toegepast. Sommige mensen passen deze strategieën bewust toe (zoals ervaren problem solvers), terwijl sommigen hier niet eens bij nadenken (zoals je vader die een feestje wil organiseren). Mary L. Gick (1986) heeft een aantal van deze strategieën in kaart gebracht.

### SCHEMA ACTIVATIE

Gick (1986) laat in haar artikel 'Problem-Solving Strategies' zien dat er een groot verschil is in hoe leken op het gebied van probleemoplossing ('novices') complexe problemen oplossen, en hoe ervaren probleemoplossers ('experts') dit doen. Een groot verschil tussen de twee niveaus is het al dan niet gebruiken van schema activatie (zie Figuur 4). Een schema is een soort stappenplan dat probleemoplossers kunnen gebruiken om een probleem sneller en efficiënter op te lossen. *The schema is a cluster of knowledge related to a problem type. It contains information about the typical problem goal, constraints, and solution procedures useful for that type of problem* (Gick & Holyoak, 1983). Bij het toepassen van schema activatie is het mogelijk om het zoeken naar een oplossing over te slaan en direct door te gaan naar het implementeren van de oplossing middels het schema. Denk hierbij bijvoorbeeld aan het gebruiken van de ABC-formule bij het oplossen van een kwadratische vergelijking in plaats van zelf de losse stappen één voor één uit te denken. Een dergelijk schema moet echter wel geactiveerd worden: als men de ABC-formule kent zal deze toegepast worden zodra duidelijk wordt dat een kwadratische vergelijking opgelost moet worden. Als men de ABC-formule niet kent (of bijvoorbeeld niet weet dat hij toegepast moet worden bij het oplossen van een kwadratische vergelijking) dan zal het schema niet geactiveerd worden en zal men overgaan naar zogeheten search-based problem solving. *"The essential contrast at this point is that schema-driven problem solving leads more directly to the implementation of strategies and solution procedures, whereas search-based problem solving requires search before a solution procedure can be implemented."*, aldus Gick. Met andere woorden, schema activatie zorgt voor een snellere oplossing van het probleem, waardoor probleemoplossing efficiënter verloopt.

FIGUUR 2





De meest voorkomende vorm van search-based problem solving is means-ends analysis. Bij means-ends analysis wordt gekeken naar de doelsituatie en de huidige situatie om vervolgens stap voor stap de verschillen tussen de huidige situatie en de doelsituatie te verkleinen (Gick, 1986). Zo kunnen we bijvoorbeeld het geven van een feestje op een means-ends analysis manier oplossen. De huidige situatie kan zijn 'het hebben van een lege kamer', de doelsituatie kan 'een feest met vrienden en veel bier in een versierde kamer' zijn. Om het verschil tussen deze twee situaties kleiner te maken kun je in de winkel slingers kopen. Als deze opgehangen zijn is de huidige situatie veranderd tot 'het hebben van een versierde kamer'. Vervolgens kun je een paar kratten bier gaan halen om het verschil met het einddoel nog kleiner te maken, enzovoorts. Het grote probleem met deze aanpak is dat er enkel en alleen moet worden gekeken naar het huidige doel en het einddoel, en dat voorgaande doelen vergeten kunnen worden (als je de slingers hebt gekocht kun je vergeten waar je ze hebt gekocht, hoe duur ze waren enz.).

Op deze manier kun men dus een doel bereiken zonder ooit te herinneren hoe dit doel behaald is: *"Consequently, one can achieve a solution to a problem using means-ends analysis without ever remembering how that solution was achieved. That is, means-ends analysis does not foster the learning of problem states and associated moves that will result in a new state. Furthermore, even if the problem solver devoted some attention to states and associated moves, the means-ends analysis strategy imposes such a working memory load that it would probably hinder the learning of states and associated moves"* (Sweller, 1985). Op deze manier wordt het gebruik van schema activatie tegengewerkt en zullen novices niet zelf het niveau van experts kunnen behalen.

Wat samenhangt met deze twee vormen van probleemoplossing is *working forward* en *working backwards*. Novices hebben de neiging om zich te focussen op het doel om vervolgens via verschillende stappen terug te werken naar het probleem om zo een oplossing te vinden (*working backwards*). Dit zien we terug bij means-ends analysis. Experts hebben echter de neiging om via verschillende stappen vanuit het probleem te werken naar het doel (*working forward*). Dit zien we terug bij schema activatie: vanuit het probleem wordt het stappenplan van het schema toegepast om uiteindelijk tot de doelstaat te komen. Hierbij is het dus niet nodig om van tevoren te weten hoe de doelstaat er precies uitziet.

Robertson (2001) noemt ook nog Hill Climbing. Hij vergelijkt deze aanpak met de situatie dat de probleemoplosser verdwaald is in een dichte mist (en dus niet om je heen kan kijken) en omhoog wil klimmen om te zien waar hij is. Hij kan kiezen tussen noord, oost, zuid en west. Als hij bijvoorbeeld een stap noord zet en voelt dat hij afdaalt trek hij zijn voet terug, draait naar het oosten en probeert het nog eens. Als hij voelt dat hij bergop gaat neemt hij deze stap. Dit wordt herhaald tot de probleemoplosser op een gegeven moment merkt dat hij aan alle vier de kanten zou afdalen: de top is bereikt. Robertson geeft als voorbeeld het afstellen van je automotor: *"This kind of problem-solving heuristic is useful if, say, the revs on your car seem to be too high and you try to adjust the petrol/air mixture. You make fine adjustments by turning a screw one way; if the revs increase then you start turning it the other way. If the revs slow down so much that the car is in danger of stalling then you turn the screw back the other way. You continue in this way until the engine sounds as if it is running neither too fast nor too slow."* Het grote nadeel aan deze aanpak is dat men, eenmaal bij de top aangekomen, geen garantie heeft dat men op de hoogste heuvel terecht is gekomen. Verder is Hill Climbing alleen mogelijk als men constante feedback krijgt om te meten of men dichterbij het doel komt of juist niet. Immers, als alles om iemand heen plat is zal diegene nooit weten welke kant hij op moet gaan om bij de top uit te komen waardoor de zoektocht willekeurig wordt.

---

## ANALOGIE

---

Verder noemt Gick (1986) nog een aantal andere strategieën voor probleemoplossing. Zo is analogie een strategie waarbij het huidige (vaak onbekende) probleem vergeleken wordt met een vergelijkbaar probleem waar de probleemoplosser reeds een helder beeld van heeft: *“The use of analogy is an important teaching device that makes the unfamiliar familiar and may affect the ease of solving problems in a new domain.”* Denk hierbij aan het feit dat men bij een probleem met elektriciteit kan denken aan stromend water in plaats van stroom. *“The target problem used by Gick and Holyoak (1980, 1983) was Duncker’s (1945) radiation problem, in which a doctor must find a way to use high-intensity rays to destroy an inoperable stomach tumor without destroying surrounding healthy tissue. The radiation problem followed a story that involved a general who had to capture a fortress using his entire army without destroying villages surrounding the fortress. The general divided the army into small groups, each of which would attack the fortress simultaneously from different roads surrounding the fortress. The analogous “convergence” solution to the radiation problem was to apply low-intensity rays from different angles around the tumor.”* (Gick, 1986). Gick zegt dat deze methode goed kan werken, mits er een beetje naartoe gestuurd wordt door een begeleider (proefpersonen doen dit bijna nooit uit zichzelf).

---

## PROBLEM SOLVING CYCLI

---

### COGNITIVE-METACOGNITIVE FRAMEWORK

---

Garofalo en Lester (1985) hebben een framework samengesteld dat de vier stappen van probleemoplossing weergeeft: *orientation*, *organisation*, *execution* en *verification* (zie

Figuur 3). Garofalo en Lester merken op dat bij verschillende problemen de nadruk ligt op andere stappen. Er zijn zelfs gevallen waar niet elke stap nodig is. Zo is er bijvoorbeeld geen planning nodig als iemand een simpele rekensom wil oplossen en zal de nadruk liggen op *execution* en *verification*. In het volgende voorbeeld, uit het artikel van Garofalo en Lester (1985), zal de nadruk echter meer liggen op *orientation* en *organisation* en niet op de *execution*:

*Ron jogs 30 km a week. He jogs 7 km on Saturday and 8 km on Sunday. On the remaining 5 days of the week Ron jogs the same number of kilometers each day. How many kilometers does Ron jog on Friday?*

Figuur 3

**ORIENTATION:** Strategic behavior to assess and understand a problem

- A. Comprehension strategies
- B. Analysis of information and conditions
- C. Assessment of familiarity with task
- D. Initial and subsequent representation
- E. Assessment of level of difficulty and chances of success

**ORGANIZATION:** Planning of behavior and choice of actions

- A. Identification of goals and subgoals
- B. Global planning
- C. Local planning (to implement global plans)

**EXECUTION:** Regulation of behavior to conform to plans

- A. Performance of local actions
- B. Monitoring of progress of local and global plans
- C. Trade-off decisions (e.g., speed vs. accuracy, degree of elegance)

**VERIFICATION:** Evaluation of decisions made and of outcomes of executed plans

- A. Evaluation of *orientation* and *organization*
  - 1. Adequacy of representation
  - 2. Adequacy of organizational decisions
  - 3. Consistency of local plans with global plans
  - 4. Consistency of global plans with goals
- B. Evaluation of *execution*
  - 1. Adequacy of performance of actions
  - 2. Consistency of actions with plans
  - 3. Consistency of local results with plans and problem conditions
  - 4. Consistency of final results with problem conditions

Het Cognitive-Metacognitive Framework analyseert de metacognitieve aspecten van het oplossen van wiskundige problemen. Metacognitie is *knowledge and beliefs about cognitive phenomena, and the regulation and control of cognitive actions*. (Garofalo en Lester, 1985). Met andere woorden is metacognitie dus kennis over de eigen kennis. Garofalo en Lester zeggen dat dit onder andere samenhangt met het plannen hoe de probleemoplosser een probleem gaat oplossen. Als we bij problem solving dus gestructureerde, grote cycli kunnen waarnemen kunnen we concluderen dat de probleemoplosser van tevoren in zijn of haar hoofd een soort overzicht of planning heeft hoe hij of zij het probleem zou gaan oplossen. Ongestructureerde, korte cycli zijn dan ook een indicatie van weinig metacognitie en dus een slecht overzicht van de oplossing.

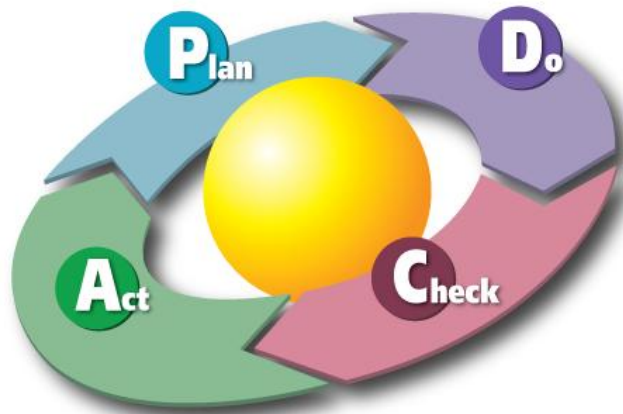
Garofalo en Lester (1985) hebben dit framework gebaseerd op de *four-phase description of problem solving activity* van Polya (1957). Dit framework zorgt er via de fases *understanding, planning, carrying out the plan* en *looking back* voor dat heuristische processen die zorgen voor succesvolle probleemoplossing geïdentificeerd worden. Het verschil tussen het framework van Polya en dat van Garofalo en Lester is vooral dat Garofalo en Lester letten op de rol van metacognitie, terwijl Polya dit alleen maar impliciet doet (Garofalo en Lester, 1985).

---

### PLAN-DO-CHECK-ACT CYCLUS

---

De Plan-Do-Check-Act cyclus, ook wel PDCA cyclus, Shewhart cyclus of Deming cyclus genoemd, is een vierstappenplan ontworpen door Walter A. Shewhart (1939). Shewhart heeft deze cyclus ontworpen met de gedachte dat het voor de evolutie van een bedrijf essentieel is om constante evaluatie en bijbehorende anticipatie te ondergaan. Het vierstappenplan gaat dus in principe constant door en wordt daarom vaak als een cirkel gerepresenteerd.



**Plan** is de eerste stap in de PDCA cyclus. In deze stap worden het probleem en het doel geïdentificeerd en geanalyseerd (Johnson, 2002). Er moet worden besloten wat er gedaan moet worden om het doel te realiseren. Er zal ook besloten worden hoe het doel het beste gerealiseerd kan worden. In deze stap wordt er dus vooral gekeken naar de huidige data en gespeculeerd hoe dit omgezet kan worden tot het doel.

De tweede stap is **Do**. In deze stap wordt de actie daadwerkelijk uitgevoerd. Ook wordt er data verzameld om de laatste twee stappen te verwezenlijken.

Vervolgens wordt de **Check** stap toegepast. Na het implementeren van de oplossing zal deze geëvalueerd moeten worden. Er wordt gekeken of de voorspelde resultaten (uit de Plan stap) overeenkomen met de daadwerkelijke resultaten.

**Act** is de laatste stap in de cyclus. Als uit de Check stap blijkt dat de oplossing niet succesvol is zullen de fouten in kaart gebracht moeten worden en moet de nieuwe kennis geformaliseerd worden. Met deze nieuwe kennis kan een nieuwe cyclus gestart worden om een nieuwe oplossing te vinden. Als de oplossing wel succesvol bleek kan deze gestandaardiseerd en gedocumenteerd worden voor toekomstig gebruik. Ook dan kan er een nieuwe cyclus gestart worden, maar deze keer voor het opnieuw verbeteren van de situatie.

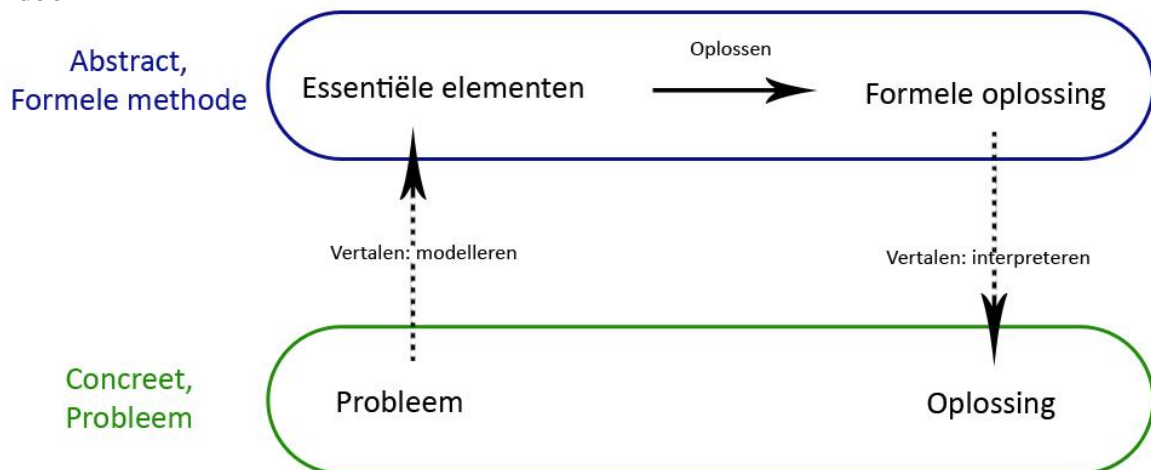
Bulsuk (2009) zegt dat het volgen van de PDCA cyclus zorgt voor een hogere kwaliteit van resultaten en processen, een constante verhoging van efficiëntie, en assistentie in het logisch en systematisch oplossen van problemen. *Simply put, PDCA is a way to reduce reliance on Murphy's Law, and move from a reactive problem fixing model to a proactive one* (Bulsuk, 2009).

Tussen de PDCA cyclus en het Cognitive-Metacognitive Framework zijn veel overeenkomsten te vinden. De Plan stap van de PDCA cyclus komt vrijwel overeen met de Orientation en Organisation stap van Garofalo en Lester. De Do stap komt overeen met de Execution stap en de Check stap komt vrijwel overeen met de Verification stap. Het grote verschil zit echter in Act stap. Garofalo en Lester houden geen rekening met de continue herhaling van de cyclus die door Shewhart wel geïmplementeerd is. Verder kijken Garofalo en Lester, net als Shewhart, naar de evaluatie van de oplossing, maar anticiperen vervolgens niet op deze evaluatie (wat in de Act stap van de PDCA cyclus wel het geval is).

## MODELLEREN

Een belangrijk aspect bij problem solving is modelleren. Flower en Hayes (1984) beschrijven een model als volgt: "A model is a metaphor for a process: a way to describe something, such as the composing process, which refuses to sit still for a portrait. As a hypothesis about a dynamic system, it attempts to describe the parts of the system and how they work together." Het maken van een model gaat vaak vanzelf: het nadenken over hoe we een puzzel zouden gaan oplossen is in feite al modelleren. Er wordt zo in ons hoofd namelijk een mentale representatie van de puzzel gevormd en deze wordt vervolgens binnen deze mentale representatie (het model) opgelost. Robertson (2001) zegt dat als we een statement van een probleem lezen we een intern model maken van deze probleemsituatie door de relevante stukken informatie bij elkaar te zetten (en de irrelevante stukken dus weg te laten). Hij legt ook uit waarom dit zo belangrijk is: "An internal model corresponds to a specific concrete situation in the external world and allows us to reason about the external situation. To do so you used information about the problem presented in the problem statement. The process of understanding, then, refers to constructing an initial mental representation of what the problem is, based on the information in the problem statement about the goal, the initial state, what you are not allowed to do, and what operator to apply, as well as you own personal past experience." Figuur 3, ontworpen door Erik Barendsen en Harrie Passier, laat duidelijk zien hoe het modelleerproces wordt toegepast bij problem solving.

FIGUUR 4



## METEN VAN EFFECTIVITEIT

---

Het lijkt erg simpel om de effectiviteit van probleemoplossing te meten: geef een proefpersoon een probleem om op te lossen en kijk of de uitkomst klopt en hoe lang de proefpersoon erover doet. Deze manier is goed om te kijken óf deze vorm van probleemoplossing werkt, maar hierdoor blijft het nog steeds volkomen onduidelijk hóe deze vorm van probleemoplossing werkt. Gezien problem solving het proces is dat we volgen om problemen op te lossen is het belangrijk om, naast concrete zaken zoals de correctheid van de oplossing, het denkproces van de proefpersoon bloot te stellen en dus te achterhalen wat de proefpersoon denkt tijdens het oplossen van het probleem. Dit is te verwezenlijken met de *Think Aloud theory* van Ericsson en Simon (1993). Met deze theorie wordt een proefpersoon simpelweg gevraagd om hardop te denken en alles te zeggen wat in hem of haar opkomt tijdens het oplossen van het probleem.

Ericsson en Simon (1993) hebben drie verschillende vormen van hardop denken gedefinieerd. Als eerste is er directe verbalisatie waarbij proefpersonen herhalen wat hun 'innerlijke stem' zegt. Zo kan men een telefoonnummer opzoeken en het zo lang mogelijk in het korte termijn geheugen houden door dit nummer constant te herhalen. Dit is relatief makkelijk: het nummer is in het korte termijn geheugen opgenomen als verbale code. Met andere woorden hoeft het niet meer vertaald te worden van concept naar woorden. Deze manier van hardop denken heeft geen positieve en geen negatieve invloed op probleemoplossing.

Bij de tweede vorm van hardop denken wordt er van de proefpersonen verwacht dat ze het denkproces actief vertalen naar woorden en vervolgens uitspreken. Zo kan een proefpersoon bijvoorbeeld een denkbeeldige taak krijgen om op te lossen waarbij hij of zij elke stap moet verbaliseren. Hoe moeilijker de taak, des te lastiger de proefpersoon het zal vinden om hardop na te denken. De bronnen die gebruikt worden om het concept te vertalen naar woorden worden immers ook gebruikt voor het oplossen van het denkbeeldige probleem. Deze vorm van hardop denken vertraagt probleemoplossing dan ook.

De laatste vorm van hardop denken lijkt op de vorige vorm maar bevat het kunnen uitleggen waarom de genomen stappen gekozen zijn. Deze vorm van hardop denken heeft een positief effect op problem solving volgens Ahlum-Heath en DiVesta (1986) en Berry (1983), maar er zijn ook nadelen. Zo zorgt het uitleggen van de stappen voor een onderbreking van het denkproces en is het moeilijk te bepalen of hetgeen de proefpersoon zegt ook daadwerkelijk de processen zijn die hij of zij gevolgd heeft. Hierdoor zijn de eerste twee vormen van hardop denken de meest voorkomende vormen bij probleemoplossing.

## CONTEXT

---

Passier en Heeren werken als docent bij de Open Universiteit Nederland: “*De Open Universiteit ontwikkelt, verzorgt en bevordert, samenwerkend in netwerken en allianties, hoogwaardig en innovatief hoger afstandsonderwijs. Als dé universiteit voor leven-lang-leren stelt zij de uiteenlopende leerbehoeften van mensen in hun levensloop én de noodzaak tot aanzienlijke verhoging van het kennisniveau van de samenleving centraal.*” (www.ou.nl) Het grootste gedeelte van de cursussen bestaat uit afstandsonderwijs. Dit betekent dat, op enkele bijeenkomsten na, de student de stof tot zich kan nemen door zelfstudie, online lectures en een digitale leeromgeving. De OU is qua indeling erg flexibel: een opleiding bestaat uit een aantal modules die je los kunt kopen en zelf kan indelen. Hierdoor is een opleiding bij de OU aantrekkelijk voor mensen die bijvoorbeeld al een baan hebben maar zich willen specialiseren.

Passier en Heeren hebben een lesmethode ontworpen over hoe men verschillende content models kan maken aan de hand van het herschrijven van reguliere expressies. Als voorbeeld kunnen we kijken naar het volgende XML element:

<!ELEMENT movie (title, producer+, length)>

Het content model van een XML element geeft aan welke kindelementen mogen voorkomen en in welke volgorde. In het voorbeeld is (title, producer+, length) het content model van het element 'movie'. Met andere woorden moet een film in dit geval één titel hebben, gevolgd door één of meerdere producers en één lengte. Er zijn veel verschillende manieren om dergelijke content models op te schrijven. Passier en Heeren hebben gekozen voor de notatie met reguliere expressies. Hierbij representeren kleine letters (a, b, c enz.) de atomen van het element en wordt de cardinaliteit van deze atomen aangegeven door '?' (komt nul of één keer voor), '\*' (komt nul of meerdere keren voor) en '+' (komt één of meerdere keren voor). Het lege atoom wordt aangegeven door 'ε' en een keuze wordt aangegeven door '|'. Zo betekent de expressie 'ab\*' dat a één keer voorkomt én b nul of meerdere keren voor mag komen en betekent 'a|b' dat a óf b voor mag komen.

Omdat er verschillende manieren zijn om expressies aan te geven is het mogelijk om ze te herschrijven zonder dat de inhoud verandert. Zo kan 'a?' (a mag nul of één keer voorkomen) ook geschreven worden als 'ε | a' (het element is leeg -m.a.w. a komt nul keer voor- óf a komt één keer voor). Passier en Heeren geven een lijstje van de mogelijke herschrijf regels:

$$\begin{array}{ll}
 R | S & = S | R & (1a) \\
 R | R & = R & (1b) \\
 \epsilon R & = R & (1c) \\
 R \epsilon & = R & (1d)
 \end{array}
 \qquad
 \begin{array}{ll}
 R? & = \epsilon | R & (2a) \\
 R^* & = \epsilon | RR^* & (2b) \\
 R^+ & = RR^* & (2c)
 \end{array}$$

$$\begin{array}{ll}
 RS | RT & = R(S | T) & (3a) \\
 RT | ST & = (R | S)T & (3b) \\
 R^* R & = RR^* & (3c) \\
 (RS)^* R & = R(SR)^* & (3d)
 \end{array}$$

$$\begin{array}{ll}
 R^n & \Rightarrow R^* & (\text{if } n \geq 0) & (4a) \\
 R^n & \Rightarrow R^+ & (\text{if } n \geq 1) & (4b) \\
 R^* S^* & \Rightarrow (R | S)^* & & (4c)
 \end{array}$$



This presents a list of rewrite rules that operate on REs. The first set of rules (1a - 1d) expresses some basic properties of the choice and sequence operators. The second set of rules (2a - 2c) shows that all occurrences of  $R?$  and  $R+$  can be removed from an expression and that  $R^*$  can be expanded one step. The third set of rules is for making expressions deterministic. More rewrite rules can be added to the collection, for example by combining existing ones. When modeling XML content with REs, it is convenient to have a rich set of rules that covers common patterns. Note that the rules can be applied in both directions (i.e., also from right to left) because both sides are equal. (Passier en Heeren, 2011)

Passier en Heeren hebben een stappenplan opgesteld om een precies content model te maken. Deze is als volgt:

**Stap 1: Maak SF**

**Stap 2: Verwijder dubbele elementen**

**Stap 3: Maak deterministisch:**

- **Hergroep:**  $R / S = S / R$
- **Factoriseer:**  $RS / RT = R(S/T)$   
of  $XR / X = XR?$

**Stap 4: Verwijder alle  $\epsilon$ 's**

- $\epsilon / R = R?$

Deze stappen kunnen we het beste duidelijk maken aan de hand van het volgende voorbeeld gebaseerd op het lesmateriaal van Passier en Heeren (2011):

```
<recs>
  <rec>
    <a/><b/><a/><b/><a/><b/>
  </rec>
  <rec>
    <a/><b/>
  </rec>
  <rec>

  </rec>
  <rec>
    <a/><b/><a/><b/>
  </rec>
  <rec>
    <a/><b/><a/><b/><a/><b/>
  </rec>
</recs>
```

Op het eerste gezicht zien we vier verschillende recs, namelijk ababab, ab, een lege rec, abab en nogmaals ababab. Met deze ruwe data kunnen we al een eerste precies model opstellen, namelijk **ababab | ab | ε | abab | ababab**. Dit eerste ruwe model noemen we een Startform, ook wel afgekort tot SF. De tweede stap is het verwijderen van dubbele elementen. We zien dat ababab twee keer voorkomt, dus we kunnen het model aanpassen tot **ababab | ab | ε | abab**. Stap drie is het deterministisch maken van het content model. Een content model is deterministisch als voor elk element in een XML-file geldt, dat we zonder vooruit te kijken kunnen bepalen welk element in het content model matcht met dat element (Passier en Heeren, 2011). Zo is bijvoorbeeld ab | ac niet deterministisch (immers, als we het element 'a' zien kunnen we niet bepalen of dit de 'a' van 'ab' of van 'ac' is zonder vooruit te kijken) en is a(b|c) wel deterministisch. Dit is te verwezenlijken door allereerst de elementen te hergroeperen met de herschrijfgregel  $R | S = S | R$ . Zo zal ons nieuwe content model als volgt zijn: **ε | ab | abab | ababab**. Vervolgens moeten we de elementen factoriseren, in dit geval met de herschrijfgregel  $XR | X = XR?$  (gezien de structuur  $RS | RT$  niet voorkomt in ons content model). Als we dit stap voor stap doen krijgen we eerst **ε | ab (ab | abab)?**, vervolgens **ε | ab (ab (ab)?)?**. De laatste stap is het wegwerken van alle lege elementen door middel van de herschrijfgregel  $ε | R = R?$ . Het uiteindelijke precieze content model is dus **(ab (ab (ab)?)?)?**.

Deze methode is een vorm van een schema: een stappenplan om het oplossen van het probleem sneller en efficiënter te laten verlopen. Als dit stappenplan succesvol kan worden aangeleerd zal in het vervolg schema activatie optreden bij het oplossen van dergelijke problemen (mits dit schema correct getriggerd wordt). Gezien dit schema wordt gebruikt voor een domein specifiek probleem waarbij het duidelijk is naar wat voor soort oplossing gezocht wordt rekenen we dit tot strong method probleemoplossing.

Zoals eerder al genoemd bestaat de cursus 'XML: theorie en toepassing' uit een pretest voorafgaande aan de leerstof, de eigenlijke cursus (zelfstudie en bijeenkomsten) en een posttest. Passier heeft van twee studenten die de cursus hebben gevolgd de pre- en posttest opgenomen waarbij ze de opdracht hebben gekregen te werk te gaan via het think aloud protocol. In deze opnames wordt er alleen aandacht geschonken aan de vaardigheid om precieze content modellen op te stellen aan de hand van reguliere expressies. Passier en Heeren (2011) beschrijven het verschil tussen precieze en correcte content models als volgt: *“We distinguish between precise content models (describing exactly the set of allowed sequences of XML elements, but nothing more), and correct models (describing at least the sequences of XML elements we want to have).”* In de cursus zelf wordt een precies content model als volgt gedefinieerd: *“Als  $R$  een content model is en  $T$  representeert alle sequenties die we willen hebben, dan is  $R$  een precies model als geldt:  $L(R) = T$ ”.*

De eerste proefpersoon is een omgeschoolde biologe met aardig wat programmeerervaring, de tweede proefpersoon is eigenaar van een ICT bedrijf in Spanje. Helaas is het Dhr. Passier door tijdnood niet gelukt om meer opnames van proefpersonen aan te leveren. Doordat de data dus vrij summier is kunnen we dan ook geen harde conclusies trekken uit de resultaten van dit onderzoek. Gezien dit onderzoek een opzet is tot een grootschaliger onderzoek is dit echter geen probleem: het is een begin. Met het programma Elluminate Live heeft Passier audio opnames kunnen maken. Verder zijn er ook opnames gemaakt van het digitale whiteboard van Elluminate Live waarop de studenten de opdrachten hebben uitgewerkt. De pre- en posttest bestaan uit soortgelijke opdrachten als Figuur 5.

FIGUUR 5

**Opgave 1**

Gegeven de volgende vier instanties van het element land:

```
<land>  
  <president>...</president>
```

```
</land>
```

```
<land>  
  <koning>...</koning>
```

```
</land>
```

```
<land>  
  <koning>...</koning>  
  <koningin>...</koningin >
```

```
</land>
```

```
<land>  
  <koningin>...</koningin >
```

```
</land>
```

De puntjes staan voor string-data. Dat is hier verder niet van belang.

**Opdracht:** Geef een zo precies mogelijk contentmodel van element land

**Antwoord:**

We leiden af:

```
president | koning | koning, koningin | koningin  
=  
president | koning, koningin? | koningin
```

---

## METHODE

---

Van beide proefpersonen zijn twee opnames gemaakt: één van de pretest en één van de posttest. Gezien de pretest plaatsvond voordat de proefpersonen aan de cursus begonnen en de posttest nadat de proefpersonen de cursus hadden afgerond hebben we door deze te vergelijken kunnen zien of de cursus invloed heeft op het maken van content models middels reguliere expressies. Bij elk van deze sessies hebben we kunnen luisteren naar wat de proefpersoon denkt (middels de audio opname) en hebben we kunnen kijken wat de proefpersoon precies heeft uitgewerkt (middels het digitale whiteboard). Het herkennen van patronen en strategieën is echter vrij lastig bij het beluisteren van audio opnames dus hebben we allereerst de opnames getranscribeerd tot teksten. Zo kunnen we op een visuele manier patronen en strategieën weergeven. We hebben er voor gekozen om dit middels kleurcodes te verwezenlijken: per invalshoek hebben we in kaart gebracht wat we proberen te herkennen in de transcripties om dit vervolgens in kleuren onder te verdelen. Proberen we bijvoorbeeld drie stappen te herkennen in de transcripties, dan kunnen we de eerste stap de kleur rood toekennen, de tweede groen en de derde blauw. Herkennen we in de transcriptie vervolgens een aantal zinnen die de eerste stap representeren dan markeren we dit stukje tekst rood. Hetzelfde geldt natuurlijk voor de tweede en derde stap (met bijbehorende kleuren). Op deze manier zijn eventuele patronen snel en duidelijk zichtbaar en is er overzichtelijk te zien welk stuk transcriptie bij welke te herkennen stap hoort. Op deze manier hebben we geprobeerd te herkennen of de methode daadwerkelijk wordt toegepast, hoe het Cognitive-Metacognitive Framework tot uiting komt, of er search-based probleemoplossing wordt toegepast en in hoeverre de PDCA cyclus tot uiting komt.

---

### TOEPASSEN VAN DE METHODE

---

Allereerst hebben we proberen te herkennen of de proefpersonen de methode van Passier en Heeren daadwerkelijk toepassen. Aangezien de proefpersonen in de pretests nog niet in aanraking zijn gekomen met de methode hebben we dit alleen proberen te herkennen in de posttests. We hebben de stappen van de methode als volgt onderverdeeld in kleuren:

**Stap 1: Maak SF**

**Stap 2: Verwijder dubbele elementen**

**Stap 3: Maak deterministisch:**

- **Hergroeppeer:**  $R | S = S | R$
- **Factoriseer:**  $RS | RT = R(S|T)$   
of  $XR | X = XR?$

**Stap 4: Verwijder alle  $\epsilon$ 's**

- **$2a. \epsilon | R = R?$**

Herkennen we in de transcriptie dus het maken van een startform dan markeren we deze zinnen rood, het hergroeperen van de elementen licht blauw enzovoorts. Het komt vaak voor dat niet alle stappen van toepassing zijn. Zo zal het verwijderen van dubbele elementen simpelweg niet van toepassing zijn als er geen dubbele elementen in de startform voorkomen. Deze stappen zullen bij de analyse grijs gekleurd zijn.

Gezien het toepassen van de methode heel concreet en zonder ambiguïteit kan worden toegepast hebben we allereerst gekeken naar de uitwerkingen van de proefpersonen op het digitale whiteboard. In deze uitwerking is het al dan niet toepassen van de stappen makkelijk en snel te herkennen. Na het eenmaal voorzien van kleurcodes aan deze uitwerkingen hebben we de bijbehorende denkstappen in de transcriptie opgezocht en deze gemarkeerd met dezelfde kleurcode.

---

## SEARCH-BASED PROBLEM SOLVING

---

Er zijn veel verschillende vormen van search-based problem solving. We hebben in de transcripties de volgende proberen te herkennen: means-end analysis, working backwards en hill climbing. Aangezien bij means-end analysis en working backwards wordt gewerkt vanuit de doelstaat hebben we proberen te herkennen dat de proefpersonen constant vergelijkingen maken tussen de huidige staat en de doelstaat. Bij hill climbing hebben we ook getracht om deze vergelijkingen waar te nemen, maar hebben we ook specifiek het aftasten van alle mogelijkheden van de proefpersoon proberen te herkennen. Als de proefpersoon bijvoorbeeld vier verschillende mogelijkheden ziet om het probleem op te lossen hebben we gekeken of hij deze mogelijkheden één voor één gaat proberen: "Werkt deze oplossing? Nee, dit is fout. Is de tweede oplossing beter? Nee, dit is ook niet goed. Ik zal de derde oplossing proberen. Dit ziet er goed uit, nu kan ik verder gaan." enzovoorts.

---

## COGNITIVE-METACOGNITIVE FRAMEWORK

---

Als tweede is gekeken naar het Cognitive-Metacognitive Framework om te kijken of er een vooruitgang is in de structuur van de cycli. Zoals eerder gezegd kunnen we de efficiëntie en planning van probleemoplossing meten aan de hand van deze structuur. Als we gestructureerde, grote cycli kunnen waarnemen kunnen we concluderen dat de probleemoplosser van tevoren in zijn of haar hoofd een soort overzicht of planning heeft hoe hij of zij het probleem zou gaan oplossen. Ongestructureerde, korte cycli zijn dan ook een indicatie van weinig metacognitie en dus een slecht overzicht van de oplossing. We hebben de pretests en de posttests van de proefpersonen geanalyseerd en vergeleken om te zien of de structuur van de cycli verbeterd is door de methode van Passier en Heeren. Om dit in kaart te brengen hebben we de vier stappen van de cyclus, net als bij voorgaand perspectief, verdeeld met kleurcodes:

### Orientation

### Organisation

### Execution

### Verification

**Orientation** is de eerste stap van de cyclus. In deze stap wordt de aanwezige informatie geanalyseerd. Er is hierbij geen sprake van speculatie of aanzet tot handelen maar komt vaak neer op het 'op een rijtje zetten' van alle informatie. In de interviews komt dit vaak neer op het maken van de Startform (waar wordt uitgeschreven welke elementen allemaal aanwezig zijn) en het opnieuw overzien van de situatie als de proefpersoon vastloopt. Bij **organisation** worden doelen geïdentificeerd en wordt gepland hoe deze doelen bereikt kunnen worden. Deze stap verschilt van orientation door het wel aanwezig zijn van speculatie en aanzet tot handelen. Redenaties en deducties horen ook thuis in deze stap. De derde stap is **execution**, het daadwerkelijke handelen om het probleem op te lossen. In dit onderzoek komt dit vooral neer op het typen zelf en het commentaar wat de proefpersonen geven tijdens het typen. Commentaar over wat ze later zullen typen hoort hier dus niet thuis.

De laatste stap is **verification** waarbij de execution stap wordt geëvalueerd. Hier wordt vaak herhaald was in de organisation stap is besloten en vervolgens of dit klopt of juist niet.

Deze stappen hebben we in de getranscribeerde tekst terug proberen te vinden aan de hand van signaalwoorden en vervolgens gemarkeerd. 'Ik begin met uitschrijven' duidt bijvoorbeeld op orientation, 'ik zou verwachten dat...' of 'Ik kan het beste zus en zo doen...' is een goed voorbeeld van organisation, het typen van de proefpersoon is duidelijk execution enzovoorts.

---

## PLAN-DO-CHECK-ACT CYCLUS

---

Naast de structuur van het Cognitive-Metacognitive Framework hebben we ook gekeken naar de structuur van de Plan-Do-Check-Act cyclus. Zoals eerder genoemd zorgt het gestructureerd toepassen van de PDCA cyclus voor een hogere kwaliteit van resultaten en processen, een constante verhoging van efficiëntie, en assistentie in het logisch en systematisch oplossen van problemen. Net als bij het Cognitive-Metacognitive Framework hebben we bij de PDCA cyclus de pretests en de posttests van de proefpersonen geanalyseerd en vergeleken om te zien of de structuur van de cyclus verbeterd is door de methode van Passier en Heeren. Om dit in kaart te brengen hebben we de vier stappen van de cyclus, net als bij voorgaande perspectieven, verdeeld met kleurcodes:

**Plan**

**Do**

**Check**

**Act**

De **Plan** stap van de PDCA cyclus kunnen we vergelijken met de orientation en organisation stap van het Cognitive-Metacognitive Framework. In deze stap wordt de aanwezig informatie geanalyseerd wat vaak neerkomt op het maken van de Startform (waar wordt uitgeschreven welke elementen allemaal aanwezig zijn). Verder worden de doelen geïdentificeerd en wordt gepland hoe deze doelen bereikt kunnen worden. Redenaties en deducties horen ook thuis in deze stap. De **Do** stap beslaat het daadwerkelijke handelen om het probleem op te lossen. In dit onderzoek komt dit vooral neer op het typen zelf en het commentaar wat de proefpersonen geven tijdens het typen. Commentaar over wat ze later zullen typen hoort hier dus niet thuis. De derde stap is de **Check** stap waar puur wordt gekeken of de acties van de Do stap correct zijn of niet. Hierbij wordt de huidige situatie vergeleken met de voorspelde situatie van de Plan stap. Bij de **Act** stap wordt geëvalueerd wat er precies fout is en wat er gedaan moet worden om deze fout op te lossen.

Net als bij het Cognitive-Metacognitive Framework hebben we deze stappen in de getranscribeerde tekst terug proberen te vinden aan de hand van signaalwoorden en vervolgens gemarkeerd. 'Ik begin met uitschrijven' of 'Ik kan het beste zus en zo doen...' is een goed voorbeeld van de Plan stap, het typen van de proefpersoon is duidelijk de Do stap, "Dit klopt niet" behoort tot de Check stap enzovoorts.

---

## BETROUWBAARHEID

---

Gezien de analyse van de transcripties door één persoon uitgevoerd is, is er discussie mogelijk over de mate van betrouwbaarheid van deze analyse. De resultaten kunnen zo namelijk onderhevig zijn aan interpretatie, voorkeur of visie van de analist. Om de betrouwbaarheid van de analyse te handhaven hebben we de inter-observer reliability bij de verschillende onderdelen vastgesteld. Hiervoor hebben we een tweede analist gevraagd om de analyses (het toepassen van de methode, het herkennen van search-based problem solving en het in kaart brengen van de Cognitive-Metacognitive Framework en de PDCA cyclus) uit te voeren op ingekorte stukken van de transcripties. Dit komt erop neer dat er voor elk perspectief minstens één opdracht van de pre- of posttests geanalyseerd is door de tweede analist. De resultaten van deze analyses hebben we vervolgens vergeleken met de resultaten van de oorspronkelijke analyses. Per opdracht hebben we het totaal aantal genomen stappen van de proefpersoon geteld en gekeken in hoeverre de analyses van deze stappen overeenkomen tussen de tweede en de oorspronkelijke analyse.

# RESULTATEN

## TOEPASSEN VAN DE METHODE

Na het analyseren van de posttests van de proefpersonen en het aanbrengen van de kleurcodes is het duidelijk dat de methode van Passier en Heeren wordt toegepast op de opdrachten. Alle acties van de proefpersonen zijn in te delen als stappen van de methode. Niet alle stappen zijn echter bij elke opdracht terug te vinden. Dit komt omdat niet elke stap bij elke opdracht van toepassing is: als er bijvoorbeeld geen dubbele elementen of  $\epsilon$ 's zijn hoeven deze ook niet te worden verwijderd. De genomen stappen worden ook in de goede volgorde genomen, hoewel opvallend is dat alleen stap 4 af en toe te vroeg wordt toegepast. Dit lijkt echter geen invloed te hebben op de strategie en de uitkomst. Hier volgt een stuk geanalyseerde transcriptie ter illustratie:

Ik begin uit te schrijven welke dat er zijn. A B C komma B C of A B C D. En dan hebben we hier nog eentje, B A C. Even kijken, 1 2 3 4, ja die heb ik. En: dan ga ik ze op volgorde zetten. ((typt)) A B C ((typt)) A B C D ((typt)) B C ((typt)) en B A C. En dan kan ik bij de twee linkse A B buiten haakjes halen. En dat heeft daarbij A B C of C D. En bij die andere kan ik B buiten haakjes halen, kan ik ook al (.) de A vraagteken tussen zetten. Die A B haakje ((typt)) A B daar kan ik C buiten haakje halen door te zeggen dat er D met een vraagteken, C D met een vraagteken (.)

**Stap 1: Maak SF**

**Stap 2: Verwijder dubbele elementen**

**Stap 3: Maak deterministisch:**

- **Hergropeer:**  $R / S = S / R$
- **Factoriseer:**  $RS / RT = R(S/T)$   
of  $XR / X = XR?$

**Stap 4: Verwijder alle  $\epsilon$ 's**

- $2a. \epsilon / R = R?$

$abc / bc / bac / abcd$   
 $abc / abcd / bc / bac$   
 $ab(c/cd) / ba?c$   
 $(abcd?) / ba?c$



---

## SEARCH-BASED PROBLEM SOLVING VS. SCHEMA ACTIVATION

---

Enigszins onverwacht is het niet gelukt om een vorm van search-based problem solving te herkennen in de pretests van de proefpersonen. De herkenning van de indicatoren die we hadden opgesteld bleef uit in de transcripties. Er is gekeken naar means-end analysis, working forward vs. working backward en hill climbing. Hill climbing bleek vrij snel al niet van toepassing te zijn op de pretests. Bij hill climbing is namelijk constante feedback nodig om te meten of je dichterbij het doel komt of niet. Deze feedback ontbreekt bij het oplossen van de opdrachten uit de pretest. Working backwards is ook niet herkend. Dit is hoogstwaarschijnlijk te wijten aan het feit dat de doelstaat van de opdrachten niet concreet genoeg is: working backwards werkt alleen als de begin- en eindstaat duidelijk zijn maar hoe je van de ene naar de andere komt nog onduidelijk is. Gezien dit niet het geval was bij de opdrachten van de pretest (immers, de antwoorden waren nog niet bekend) hebben de proefpersonen deze aanpak niet kunnen aannemen. Hetzelfde geldt voor means-end analysis: dit kan alleen plaatsvinden als de doelstaat duidelijk gedefinieerd is. Zoals verwacht is er in de pretests ook geen vorm van schema activatie gevonden. Dit zou, bij gebrek aan andere probleemoplossing perspectieven, dus betekenen dat de proefpersonen in de pretests een ongecontroleerde en ongestructureerde vorm van probleemoplossing toepassen waarbij er puur op intuïtie beslissingen genomen werden.

We hebben echter wel duidelijk schema activatie kunnen waarnemen in de posttests. Middels het herkennen van het toepassen van de methode hebben we gezien dat de methode daadwerkelijk wordt toegepast in de posttests. Gezien deze methode een domein specifiek schema is kunnen we aannemen dat in de posttests wel schema activatie voorkomt.

---

## COGNITIVE-METACOGNITIVE FRAMEWORK

---

Bij het proberen te herkennen van het Cognitive-Metacognitive Framework hebben we gekeken naar de pretests en de posttests van de proefpersonen. Na het analyseren en het aanbrengen van de kleurcodes hebben we een verschil kunnen waarnemen tussen de resultaten van de pretests en de resultaten van de posttests. In de pretests zien we dat de verschillende stappen van de problem solving cyclus veelal in de foute volgorde worden genomen. Ook zien we dat er per opdracht meerdere kleine cycli worden gevormd. De proefpersonen springen als het ware van de hak op de tak zonder al te veel structuur. Ter illustratie het volgende stuk geanalyseerde transcriptie van de pretests:

Je hebt een president of een koning en een koningin, die kunnen dan wel allebei tegelijk aanwezig zijn. (.) Als ik er helemaal haakjes omheen zet (.) dan is er in ieder geval een van de twee aanwezig. Nu heb ik hem bij koning weggehaald ((typt)) Dat kan ook niet (.) Nou ik weet het niet ((typt)) Ik denk dat ik hem zo laat. Als ik het geheel tussen haakjes zet, dan blijft er eh: altijd eentje (.) En de dingen die er tussenin staan zijn in principe allemaal optioneel (.) Ja of het is een keuze, en daar kan er maar eentje van zijn (.) oh dat kan, ja vast wel. ((typt)) Zo hoortie, ik zette er teveel vraagtekentjes in. Je hebt de president of een koning en een koningin en dan zou die koning en die koningin, die zou een van de twee kan dat zijn. (.) Dan krijg je er drie eh: (.) Of ja, dan kom ik weer met die deterministische (.) ((typt)) Zo hoortie.

<b>Orientation</b>
<b>Organisation</b>
<b>Execution</b>
<b>Verification</b>

In de posttests zien we echter een grote verbetering op dit gebied. Hier werden de stappen grotendeels in de goede volgorde genomen en werden er minder, maar grotere cycli gevormd. De oplossing verging een stuk meer conform de structuur van het Cognitive-Metacognitive Framework. Het valt op dat er meerdere malen te zien is dat het oplossen van het probleem begint met orientation en eindigt met verification, maar dat er tussenin een afwisseling van organisation en execution wordt waargenomen. Dit valt te verklaren door het feit dat de proefpersonen tegelijkertijd hardop nadenken (organisation) en typen (execution) waardoor deze twee stappen parallel lopen. Het parallel lopen van stappen is echter niet aan te geven in de transcripties waardoor het lijkt alsof er een constante afwisseling plaatsvindt. Ter illustratie het volgende stuk geanalyseerde transcriptie van de posttests:

Ehm, ik begin gewoon met het uitschrijven, dat is toch het makkelijkste. ((typt)) En dan moet ik even kijken want alleen C of D C ((typt)) (onverstaanbaar) alleen een D, alleen een C, een D en een C. Dit is [onverstaanbaar] tussen haakjes C, B C haakje vraagteken. En de D is al dan niet aanwezig ((typt)) dus dan wordt het D vraagteken C en het geheel mag ook leeg zijn. En dan [onverstaanbaar]. De D hoeft er niet te zijn en dan is het C of de D is er wel met een D C. (.) [onverstaanbaar]. Ja.

---

## PLAN-DO-CHECK-ACT CYCLUS

---

Ook bij het analyseren middels de PDCA cyclus hebben we de pretests en de posttests met elkaar vergeleken na het analyseren en het aanbrengen van de kleurcodes. Net als bij het Cognitive-Metacognitive Framework kunnen we bij de PDCA cyclus een verschil waarnemen, hoewel ze dit keer minder duidelijk zijn. Het is nog steeds zo dat in de pretests kleinere cycli worden gemaakt dan in de posttests, maar de volgorde is nagenoeg correct. Het komt echter soms voor dat een stap wordt overgeslagen. Dit uit zich meestal in het voorkomen van of de Check stap of de Act stap, maar niet allebei. Ter illustratie het volgende stuk geanalyseerde transcriptie van de pretests:

Je hebt een president of een koning en een koningin, die kunnen dan wel allebei tegelijk aanwezig zijn. (.) Als ik er helemaal haakjes omheen zet (.) dan is er in ieder geval een van de twee aanwezig. Nu heb ik hem bij koning weggehaald ((typt)) Dat kan ook niet (.) Nou ik weet het niet ((typt)) Ik denk dat ik hem zo laat. Als ik het geheel tussen haakjes zet, dan blijft er eh: altijd eentje (.) En de dingen die er tussenin staan zijn in principe allemaal optioneel (.) Ja of het is een keuze, en daar kan er maar eentje van zijn (.) oh dat kan, ja vast wel. ((typt)) Zo hoortie, ik zette er teveel vraagtekentjes in. Je hebt de president of een koning en een koningin en dan zou die koning en die koningin, die zou een van de twee kan dat zijn. (.) [onverstaanbaar] Dan krijg je er drie eh: (.) Of ja, dan kom ik weer met die deterministische (.) ((typt)) Zo hoortie.

Plan
Do
Check
Act

In de posttests zien we vooral een verbetering in de verhouding tussen de Plan stap en de Do stap. Ook zien we minder, maar grotere cycli dan bij de pretests. Net als bij de pretests valt bij de posttests op dat de Check stap wel eens wordt overgeslagen. Dit komt omdat het vergelijken van de huidige situatie met de voorspelde situatie van de Plan stap vaak uit blijft. Waarschijnlijk valt dit te wijten aan het feit dat tijdens de Plan stap wel wordt gekeken naar wat er gedaan moet worden om het probleem op te lossen, maar niet hoe de situatie er dan uit zal zien. Als er geen situatie voorspeld is kan deze ook niet vergeleken worden met de huidige situatie. Ter illustratie het volgende stuk geanalyseerde transcriptie van de posttests:

Ik begin uit te schrijven welke dat er zijn. A B C komma B C of A B C D. En dan hebben we hier nog eentje, B A C. Even kijken, 1 2 3 4, ja die heb ik. En: dan ga ik ze op volgorde zetten. ((typt)) A B C ((typt)) A B C D ((typt)) B C ((typt)) en B A C. En dan kan ik bij de twee linkse A B buiten haakjes halen. En dat heeft daarbij A B C of C D. En bij die andere kan ik B buiten haakjes halen, kan ik ook al (.) de A vraagteken tussen zetten. Die A B haakje ((typt)) A B daar kan ik C buiten haakje halen door te zeggen dat er D met een vraagteken, C D met een vraagteken (.) En dan heb ik hier een fout gemaakt. Want die hebben alledrie A B C. Dus hij is veel makkelijker. Ik zet hier B met een vraagteken. Of de andere, B A vraagteken C. (.) Ja, dit is goed.

In dit voorbeeld is de invloed van de Act stap goed te zien: tijdens het evalueren van de huidige situatie wordt een fout ontdekt waarna weer een nieuwe (zij het korte) cyclus wordt uitgevoerd.

---

## BETROUWBAARHEID

---

Bij de analyse van het toepassen van de methode, en daarmee de analyse van schema activatie, was de inter-observer reliability 100%: de analyse van alle genomen stappen kwamen overeen met de oorspronkelijke analyse. Hetzelfde gold voor het herkennen van search-based problem solving. Bij het in kaart brengen van het Cognitive-Metacognitive Framework kwam 83,33% van de analyse van de denkstappen overeen met de oorspronkelijke analyse. Bij het in kaart brengen van de PDCA cyclus was dit ook 83,33%. Na het bespreken van deze analyse met de tweede analist bleek dit echter niet helemaal correct te zijn. Door het feit dat de tweede analist, voor het behouden van de compactheid van deze test, geen toegang had tot de audio opnames bleek hij sommige stappen verkeerd geïnterpreteerd te hebben. Zo kwam het bijvoorbeeld voor dat het herhalen van de proefpersoon wat zij zojuist heeft opgeschreven werd opgevat als het opnoemen van informatie die gegeven was bij de opdracht. Dit kwam door het ontbreken van zaken als intonatie en het zien van bijbehorende uitwerking op het digitaal whiteboard. Bij de PDCA cyclus resulteerde dit bijvoorbeeld in het categoriseren van tekst tot de Plan stap in plaats van de Check stap. Met deze nieuwe informatie heeft de tweede analist zijn analyse nogmaals bekeken en correcties aangebracht. Na deze correcties was de inter-observer reliability van het Cognitive-Metacognitive Framework verbeterd tot 90%. De inter-observer reliability van de PDCA cyclus werd verbeterd tot 94,44%. De bezwaren die de tweede analist op de categorisatie van de oorspronkelijke analist had waren inhoudelijk bezwaren: de analisten waren het niet met elkaar eens bij welke categorie de stappen hoorden. Dit betekent dat de bezwaren niet gebaseerd waren op onduidelijkheden in de methode waardoor deze niet herschreven hoefde te worden.

## CONCLUSIE EN DISCUSSIE

---

De hoofdvraag van dit onderzoek is **Op welke manier kan de invloed van de cursus 'XML: theorie en toepassing' op het denkproces bij fundamentele XML vaardigheden in kaart gebracht worden?**

Om deze vraag te kunnen beantwoorden zullen we antwoord geven op de volgende deelvragen:

- Welke manieren zijn geschikt voor de evaluatie van de cursus?
- Zijn deze manieren bruikbaar?
- Zijn deze manieren betrouwbaar?

Om de eerste deelvraag te beantwoorden hebben we gekeken of schema activatie, means-end analysis, working backwards en hill climbing te herkennen zijn als probleemoplossing strategieën in de transcripties. Verder hebben we getracht om de stappen van het Cognitive-Metacognitive Framework en de PDCA cyclus te herkennen. We hebben schema activatie en de stappen van het Cognitive-Metacognitive Framework en de PDCA cyclus weten te herkennen in de transcripties van de pre- en posttests van de proefpersonen. Hieruit kunnen we concluderen dat het trachten te herkennen van schema activatie een geschikte manier is om de cursus van Passier en Heeren te evalueren gezien het toepassen hiervan een indicatie is voor een snellere oplossing van het probleem, waardoor probleemoplossing efficiënter verloopt. Zoeken naar een verbetering van het toepassen van het Cognitive-Metacognitive Framework is ook geschikt gezien dit een indicatie van een betere planning tot oplossing en hogere efficiëntie is. Als laatste is zoeken naar een verbetering van het toepassen van de PDCA cyclus ook geschikt voor de evaluatie van de cursus van Passier en Heeren gezien dit een indicatie is van een hogere efficiëntie en hogere kwaliteit van resultaten en processen.

We hebben verder nog willen kijken of er in de pretests search-based problem solving voorkomt door means-end analysis, working backwards en hill climbing te herkennen. Dit is niet gelukt door het feit dat deze vormen van probleemoplossing niet helemaal aansluiten bij de aard van de problemen uit dit onderzoek. Bij search-based problem solving is namelijk een duidelijke doelstaat nodig voor feedback en sturing. Bij de problemen uit dit onderzoek is van tevoren echter niet duidelijk hoe de doelstaat er precies uitziet (het antwoord is immers nog niet bekend). Gezien de doelstaat niet helder is kan deze ook niet gebruikt worden voor feedback en blijft de te herkennen structuur van means-end analysis ook uit. Deze vorm van evaluatie is dus niet geschikt.

Om de tweede deelvraag te beantwoorden hebben we de manieren uit de eerste deelvraag toegepast op de transcripties van de opnames van de proefpersonen. Passier en Heeren willen onderzoek doen naar het volgende probleem: *Heeft de cursus 'XML: theorie en toepassing' invloed op het denkproces bij fundamentele XML vaardigheden?* Om deze vraag te beantwoorden kunnen we kijken naar de volgende aspecten:

- Wordt de strategie van de cursus toegepast?
- Op welke manier bevordert de cursus het denkproces?

Het eerste aspect is heel kort te beantwoorden: ja, de strategie van de cursus wordt toegepast door de proefpersonen. In de pretests is het duidelijk dat structuur in de vorm van de strategie van de cursus ontbreekt. In de posttests komt echter onmiskenbaar naar voren dat de strategie wordt toegepast: de verschillende stappen van de cursus zijn allemaal terug te zien in de getranscribeerde opnames.

Het tweede aspect is wat uitgebreider. Er is allereerst gekeken naar schema activatie. In de pretests was geen vorm van schema activatie te herkennen. Gezien de strategie in de cursus op zichzelf een schema is hebben we met de eerste deelvraag bewezen dat schema activatie wel plaatsvindt in de posttests. Gick (1986) zegt het volgende over schema activation: "*The essential contrast at this point is that schema-driven problem solving leads more directly to the implementation of strategies and solution procedures, whereas search-based problem solving requires search before a solution procedure can be implemented.*". Met andere woorden, schema activatie zorgt voor een snellere oplossing van het probleem, waardoor probleemoplossing efficiënter verloopt. Op dit gebied is de cursus dus bevorderend voor de efficiëntie van probleemoplossing.

Verder is er gekeken naar de het Cognitive-Metacognitive Framework. Bij de meest efficiënte vorm van probleemoplossing zou een opvolging van orientation naar organisation naar execution naar verification te zien zijn met één cyclus per probleem. Bij de analyse is naar voren gekomen dat de pretests over het algemeen bestaan uit vele kleine cycli die ongestructureerd en vaak in verkeerde volgorde voorkomen. Dit is een indicatie van een lage mate van metacognitie waarbij de probleemoplosser dus een slecht overzicht van het probleem heeft. Een slechte planning tot oplossing en inefficiëntie is hiervan het gevolg. In de posttests zien we echter een verbetering in de structuur van deze cycli. Hier zijn grote cycli waarneembaar die gestructureerd voorkomen. Dit is een indicatie van een hogere mate van metacognitie waarbij de probleemoplosser dus een beter overzicht van het probleem heeft. Een betere planning tot oplossing en hogere efficiëntie is hiervan het gevolg.

Als laatste is gekeken naar de PDCA cyclus. Bij deze analyse zijn de verschillen tussen de pretests en de posttests wat minder, maar toch zijn er verbeteringen waarneembaar. Net als bij het Cognitive-Metacognitive Framework zijn de cycli uit de pretests kleiner van stuk maar groter in aantal. De posttests bestaan uit minder maar grotere cycli. Verder zien we dat de structuur van de cycli verbeterd is tot parallel lopende Plan en Do stap afgesloten door een Check en Act stap. Hierbij kan echter de Check stap nog wel eens wegvallen. Dit komt omdat het vergelijken van de huidige situatie met de voorspelde situatie van de Plan stap vaak uit blijft. Waarschijnlijk valt dit te wijten aan het feit dat tijdens de Plan stap wel wordt gekeken naar wat er gedaan moet worden om het probleem op te lossen, maar niet hoe de situatie er dan uit zal zien. Als er geen situatie voorspeld is kan deze ook niet vergeleken worden met de huidige situatie. Dit alles komt waarschijnlijk omdat de PDCA cyclus vooral is bedoeld voor het behandelen van grotere projecten bij bedrijven en daarom wat minder aansluit op het huidige onderzoek. Het is desalniettemin geschikt voor dit onderzoek, maar bij een dergelijke kleinschalige problemen kunnen sommige dingen wegvallen. Met de verbeteringen zorgt de methode van Passier en Heeren voor een hogere kwaliteit van resultaten en processen, een constante verhoging van efficiëntie en assistentie in het logisch en systematisch oplossen van problemen.

Met het eerste aspect is dus bewezen dat de methode uit de cursus daadwerkelijk wordt toegepast. Met aspect is gebleken dat het succes van de cursus ligt bij het verbeteren van efficiëntie door schema activatie, het optimaliseren van metacognitie en het verhogen van de kwaliteit van de resultaten en efficiëntie. Hierdoor kunnen we concluderen dat de cursus 'XML: theorie en toepassing' een positieve invloed heeft op het denkproces bij fundamentele XML vaardigheden.

Het onderzoek in deze scriptie is echter bedoeld om een opzet te maken voor een grootschaliger onderzoek op dit gebied. Hierdoor is het moeilijk om op basis van de huidige, summier data harde conclusies te trekken. Er valt te discussiëren hoe betrouwbaar de resultaten van dit onderzoek zijn door het feit dat er maar twee proefpersonen aan hebben deelgenomen. Het is voor het vervolgonderzoek dan ook belangrijk om het aantal proefpersonen drastisch te verhogen. Hierdoor wordt het onderzoek betrouwbaarder en zullen uitbijters makkelijker te detecteren zijn zodat men hier op kan anticiperen. Verder moet er zorgvuldiger gekeken worden naar de pre- en posttests. In het huidige onderzoek zitten deze niet helemaal op één lijn. Zo zijn de opgenomen pretests een stuk omvangrijker dan de opgenomen posttests (vier opdrachten tegenover één of twee opdrachten). Verder zitten er nuanceverschillen tussen de twee tests. Zo verschilt de mate van abstractheid: de pretest bevat zowel vragen waarbij veel geabstraheerd moet worden als vragen waarbij dit niet hoeft. De posttest bevat alleen vragen waarbij dit niet hoeft. Het is momenteel dan ook onmogelijk om vast te stellen of de nieuwe strategie invloed heeft op de vaardigheid om te abstraheren, simpelweg omdat dit niet is opgenomen in de posttests. Samenvattend moet er bij het vervolgonderzoek dus op gelet worden dat de pre- en posttests beter op elkaar aansluiten. Als laatste is het lastig om te monitoren of de proefpersonen de stappen die ze hardop denken ook daadwerkelijk nemen. Het is namelijk vaak lastig om het denkproces precies te vertalen naar woorden. Er kunnen dus nuanceverschillen zitten tussen het eigenlijke denkproces en hetgeen de proefpersonen hardop denken. Dit is echter een risico wat altijd aanwezig zal zijn bij think aloud sessies.

Om de laatste deelvraag te beantwoorden hebben we de inter-observer reliability bij de verschillende onderdelen vastgesteld. Gezien deze 100% was bij het analyseren van schema activatie, 90% bij het in kaart brengen van het Cognitive-Metacognitive Framework en 94,44% bij het in kaart brengen van de PDCA cyclus kunnen we concluderen dat deze manieren van evaluatie betrouwbaar zijn. Hierbij moet wel gezegd worden dat het analyseren van alleen de transcripties onvoldoende is: het vooraf beluisteren van de audio opnames bleek essentieel te zijn. Deze laatste conclusie is gevormd uit de discussie tussen de twee analisten. In een vervolgonderzoek is het dus wenselijk om te kijken of de tweede analist daadwerkelijk voor een hogere inter-observer reliability zorgt na het van tevoren beluisteren van de audio opnames. Verder is het wenselijk om deze resultaten te ontdoen van de factor toeval (door bijvoorbeeld Cohen's Kappa toe te passen).

Afsluitend kunnen we concluderen dat door te kijken naar het al dan niet aanwezig zijn van schema activatie en vorderingen in het toepassen van het Cognitive-Metacognitive Framework en de PDCA cyclus we de invloed van de cursus 'XML: theorie en toepassing' op het denkproces bij fundamentele XML vaardigheden in kaart kunnen brengen.

## BIBLIOGRAFIE

---

### BOEKEN

---

Robertson, S. Ian, (2001), *'Problem Solving'*, University of Luton, UK

Shewhart, Walter A., (1939), 'Statistical method from the viewpoint of quality control', The Graduate School, the Department of Agriculture, Washington D.C.

### ARTIKELN

---

Passier, H., Heeren, B., (2011), *'Modeling XML Content Explained'*

Garofalo, J., Lester, Frank K. Jr., (1986), 'Journal for Research in Mathematics Education, *National Council of Teachers in Mathematics* Vol. 16, No. 3, pp. 163-176

Flower, L., Hayes, J. R. (1984), 'A cognitive process theory of writing', *College Composition and Communication* Vol. 32, No. 4, pp. 365-387.

Gick, Mary L., (1986), Problem-Solving Strategies, *Educational Psychologist*, 21:1-2, 99-120

Sweller, J., (1985), *'Cognitive load during problem solving: Effects on learning'*

Ericsson, K.A., Simon, H.A., (1993), *Protocol analysis: Verbal reports as data* (rev. ed.), Cambridge, MA: MIT Press

Ahlum-Heath, M.E., DiVesta, F.J., (1986), The effects of conscious controlled verbalization of a cognitive strategy on transfer in problem solving, *Memory and Cognition*, Vol. 14, pp. 281-285

Berry, D.C., (1983), Metacognitive experience and transfer of logical reasoning, *Quarterly Journal of Experimental Psychology*, Vol. 35A, pp. 39-49

Bulsuk, Karn G., (2009), *'Taking the First Step with the PDCA (Plan-Do-Check-Act) Cycle'*

Johnson, C. N., (2002), *'The benefits of PDCA. Quality Progress'*, 35(5), 120



## APPENDIX

### TOEPASSEN VAN DE METHODE

#### POSTTEST PROEFPERSOON 1

P: Ik begin uit te schrijven welke dat er zijn. A B C komma B C of A B C D. En dan hebben we hier nog eentje, B A C. Even kijken, 1 2 3 4, ja die heb ik. En: dan ga ik ze op volgorde zetten. ((typt)) A B C ((typt)) A B C D ((typt)) B C ((typt)) en B A C. En dan kan ik bij de twee linkse A B buiten haakjes halen. En dat heeft daarbij A B C of C D. En bij die andere kan ik B buiten haakjes halen, kan ik ook al (.) de A vraagteken tussen zetten. Die A B haakje ((typt)) A B daar kan ik C buiten haakje halen door te zeggen dat er D met een vraagteken, C D met een vraagteken (.) En dan heb ik hier een fout gemaakt. Want die hebben alledrie A B C. Dus hij is veel makkelijker. Ik zet hier B met een vraagteken. Of de andere, B A vraagteken C. (.) Ja, dit is goed.

**Stap 1: Maak SF**

**Stap 2: Verwijder dubbele elementen**

**Stap 3: Maak deterministisch:**

- **Hergroep:**  $R / S = S / R$
- **Factoriseer:**  $RS / RT = R(S/T)$   
of  $XR / X = XR?$

**Stap 4: Verwijder alle  $\epsilon$ 's**

- $2a. \epsilon / R = R?$

$abc / bc / bac / abcd$   
 $abc / abcd / bc / bac$   
 $ab(c/cd) / ba?c$   
 $(abcd?) / ba?c$

H: Maar ik zie hem, ja nu zie ik hem. Ja, en wat denk je nu?

P: Hij is klaar, dus nu zou ik eigenlijk even moeten kijken van welke mogelijkheden heeft hij. En dat is A B C of de D erbij A B C D of B zonder A C en nu met A C. En dan denk ik dat ik ze allemaal, dan denk ik dat hij goed is. (.)

H: Ja?

P: Ja

H: Ok. Keurig. Ik vind het heel grappig om te zien dat er inderdaad een heel ander denkproces bij je gaande is wat veel meer gestroomlijnd is.

P: Het is veel minder chaotisch.

(...)

P: Ehm, ik begin gewoon met het uitschrijven, dat is toch het makkelijkste. Eerst een lege ((typt)) En dan moet ik even kijken want hij is C of D C ((typt)) Of hij is [onverstaanbaar] alleen een D, alleen een C, een D en een C. Dit is dan een (.) buiten: [onverstaanbaar] tussen haakjes C, D C haakje vraagteken. En de D is al dan niet aanwezig ((typt)) dus dan wordt het D vraagteken C en het geheel mag ook leeg zijn. En dan [onverstaanbaar]. De D hoeft er niet te zijn en dan is het C of de D is er wel met een D C. (.) En dan is het volgens mij, deze is goed. Ja.

$\epsilon / c / dc$   
 $(c / dc)?$   
 $(d?c)?$

**Stap 1: Maak SF**

**Stap 2: Verwijder dubbele elementen**

**Stap 3: Maak deterministisch:**

- Hergroep:  $R / S = S / R$
- Factoriseer:  $RS / RT = R(S/T)$   
of  $XR / X = XR?$

**Stap 4: Verwijder alle  $\epsilon$ 's**

- $2a. \epsilon / R = R?$

H: Ja, hij is volgens mij ook goed. Ja dat ging best, hoe vond je het gaan?

P: Het gaat een stuk beter, ik kan nou

---

POSTTEST PROEFPERSOON 2

---

W: Even kijken als ik hier begin en dus als we de mogelijkheden allemaal uit gaan schrijven. A B A B of A B of een lege daar gebruik ik een euro voor, want die stelt toch weinig voor, en als laatste A B A B. Dus hier beginnen we mee. Jij ziet wat ik schrijf?

H: Ja.

W: Dus wat we hiermee kunnen doen ehm, is die, wacht even hoor, als ik hieronder ga schrijven, ok dan als ik die lege, ik doe even twee stappen tegelijk anders moet ik zo veel schrijven. *Wat ik doe ik haal die lege er gewoon af en dan kan ik dit schrijven op deze manier. En dan zetten we hier een vraagtekentje achter van die is er of één keer of niet.* Dan ben ik dat lege ding dus kwijt. Wat ik dan hiermee kan doen is een hele mooie A B buiten haakjes halen en dan krijg ik binnen de haakjes hier een A B A B, dan krijg ik of een lege of een A B en dit haakje en dit vraagtekentje blijft gewoon staan. *Dan kan ik op diezelfde manier dus weer die lege erbij halen en dan kan ik nog een keer (.) de (.) A B erbuiten gaan halen en als ik dan doorga als het goed is dan hou ik dit over. (.)*

**Stap 1: Maak SF**

**Stap 2: Verwijder dubbele elementen**

**Stap 3: Maak deterministisch:**

- *Hergroep:  $R / S = S / R$*
- *Factoriseer:  $RS / RT = R(S/T)$*   
*of  $XR / X = XR?$*

**Stap 4: Verwijder alle  $\epsilon$ 's**

- *$2a. \epsilon / R = R?$*

*ababab | ab |  $\epsilon$  | abab  
(ababab | ab | abab)?  
(ab(abab |  $\epsilon$  | ab))?  
(ab(ab(ab)?)?)?*

H: Ja je, wat denk je nu?

W: Ja dit was het volgens mij?

H: Dit was het? Je bent nu helemaal klaar met de opgave?

W: Ja ik denk dat dit het is. Ja ik ben klaar met het oplossen, het enige wat we nog moeten doen is kijken of inderdaad alles dekt.

H: Ja is prima, ga maar door, dat is een essentieel onderdeel

W: Ja nou, ja uiteraard. Maar dus de eerste, A B A B A B zit er duidelijk dus in (.) Ja, de A B van die tweede die zit er ook in. Als ik even, als ik de voorlaatste en het daarvoorstaande vraagtekentje weglaat zeg maar dan hou ik dus A B over. De lege die zit er ook in voor het laatste vraagtekentje [onverstaanbaar] en de A B A B zit er ook in. (.) Als ik dit tweede vraagtekentje: ja, ze zijn er allemaal.

H: Ja ok, dat is correct.

## COGNITIVE-METACOGNITIVE FRAMEWORK

---

### PRETEST PROEFPERSOON 1

---

P: Ik zie eh . vier elementen en (.) de ene heeft president en de andere (.) hebben koning, koningin en nog eentje koning. Ehm (.) content zou ik dus verwachten dat de president is optioneel kan [onverstaanbaar] bepalen voor president of koning (.) wel koning koningin. Die zitten achter elkaar. Dus dan zou ik (.) dan begin ik het op te schrijven. (.) En dan haakje openen (.) ehm (.) ((typt)) president ((typt)) komma koning

H: Je mag trouwens ook wel afkortingen nemen hoor want het wordt misschien wel erg lang als je het achter elkaar moet intikken.

J: Oh ja, dat kan ook. Ehm en dan zet ik er de (.) optionele dingen achter eh. Ja een beetje raar want ze zijn allemaal optioneel.((typt)) Dat is wat ik dan in eerste instantie denk. Die koning en koningin die horen bij elkaar (.) en ze komen allemaal maar één keer voor (.) Ik zou wel zeggen van [onverstaanbaar] president en de koning of koningin je hebt ze niet tegelijk (.)

H: Ja, ga door, met denken.

P: Dus nu heb ik ze allemaal los optioneel (.) dus (.) ehm: (.) je hebt de keuze, het moeten eigenlijk keuzes zijn. (.) Maar dan is het een keuze tussen koning koningin, die kunnen wel bij elkaar zijn, en hier hoort dan een 'of' tussen. (.) ((typt)) zoiets zou ik denken. Ze zijn [onverstaanbaar] alle drie optioneel (.) maar dat zou betekenen dat er niks in kan staan. Dat zou [onverstaanbaar] natuurlijk mogelijk zijn, maar (.) de vraagtekentjes staan verkeerd.

H: Uhuh

J: (.) ((typt)) ow (.) dat krijg ik [onverstaanbaar] geprutst. (.) Hij moet aanwezig zijn dus het geheel moet iets hebben (.) Ja de vraagtekentjes moeten weg misschien wel. Even kijken (.) dat kan ik niet aangeven, dus of die of de andere dus (.) nou ik zal hem toch zo laten staan denk ik. Het is niet goed met die vraagtekens maar ik weet eigenlijk wat ik eraan moet veranderen.

H: Sorry, wat zei je als laatste?

P: Nou het is volgens mij niet goed maar ik weet zo gauw niet wat ik eraan moet veranderen. (.) Dat moet ik opzoeken. (.)

H: Moet je iets opzoeken?

P: Nou nee nu niet eh: (.)

H: Maar je bent nu klaar met de opgave wat jou betreft?

P: Ja ik zou m even zo laten staan. Er is iets niet goed maar ik weet even niet precies wat, eh: (.) Echt, ja (.) Ik heb nu het idee van hij kan helemaal leeg zijn, en dat is niet eh niet de bedoeling eh (.)

H: Uhuh (.) ja (.) Wat denk je dan?

P: Ja, ik ben, je moet (.) ehm hier een haakje achter zetten en dan (.) Je hebt een president of een koning en een koningin, die kunnen dan wel allebei tegelijk aanwezig zijn. (.) Als ik er helemaal haakjes omheen zet (.) dan is er in ieder geval een van de twee aanwezig. Nu heb ik hem bij koning weggehaald ((typt)) Dat kan ook niet (.) Nou ik weet het niet ((typt)) Ik denk dat ik hem zo laat. Als ik het geheel tussen haakjes zet, dan blijft er eh: altijd eentje (.) En de dingen

die er tussenin staan zijn in principe allemaal optioneel (.) Ja of het is een keuze, en daar kan er maar eentje van zijn (.) oh dat kan, ja vast wel. ((typt)) Zo hoortie, ik zette er teveel vraagtekentjes in. Je hebt de president of een koning en een koningin en dan zou die koning en die koningin, die zou een van de twee kan dat zijn. (.) [onverstaanbaar] Dan krijg je er drie eh: (.) Of ja, dan kom ik weer met die deterministische (.) ((typt)) Zo hoortie. Die zijn ook [onverstaanbaar] (.) Even kijk of ik nu de haakjes goed heb. Het idee is dat een land met een president of (.) een konings eh: huis. En dan heb je een koning met een koningin of eigenlijk alleen de koning of alleen de koningin (.)

H: Uhuh

P: Zo laat ik hem

H: Ja hij is klaar?

P: Zo zou ik denken dat het eh: moest zijn ja. De haakjes kloppen nog niet zie ik. ((typt)) die horen niet tussen haakjes. (.) Zo.

H: Ja?

P: Ja, zo hoortie denk ik

H: Ok. Dan gaan we naar de tweede opgaven. Ik zal daarvoor een nieuwe pagina maken. Die heb je hier. En het is hetzelfde recept dus het woord is aan jou. Eh, je ziet hier eigenlijk weer een zelfde soort termen, nu drie instanties van het element letters en ook hier is de vraag om het bijbehorende content model te maken dus je kan ook hier gelijk beginnen met haakje openen haakje sluiten en dat wordt het content model dus letters hoeft je ook niet eens op te nemen. (.)

P: Hij kan leeg zijn (.) Dat betekent dat er ook een eh (.) [onverstaanbaar] sterretje aan het eind, buiten de haakjes (.) A, en dan (.) A zonder B en B voor een A dus we maken B optioneel en A is in allebei aanwezig dus een A, en dat is er maar één dus die hoeft geen dingetje. [onverstaanbaar] (.) Dit is klaar.

H: Dit is m?

P: Ja (.)

H: Kan je mij nog wat meer uitleg geven? Want ik hoorde je wel wat zeggen, maar in een keer staat, plop, deze expressie op het scherm die ik niet helemaal kan knopen aan jou gedachtegang.

P: Ik denk hij kan leeg zijn, dus het content model staat een sterretje achter, nul of meer keer. (.) Ja hij is niet meer keer trouwens. Het is een vraagteken. Ik pas m aan. ((typt)) Want het geheel is optioneel. Hij kan dan leeg zijn en er kan B, die mag aanwezig zijn maar die hoeft niet aanwezig te zijn. Daarom een B met een vraagteken erachter. En het volgende element is een A, als tweede element. (.) [onverstaanbaar]

H: Uhuh, ben je klaar?

P: Ja

H: Ok, dan gaan we naar de volgende. Het is misschien wel handig voor deze opgave, iets soortgelijks, je krijgt hier al een content model. Je mag gewoon 'straat' met een s'je afkorten en huisnummer met een -h. (.)

P: (.) De eerste vraag is wat we mis[onverstaanbaar] met postadres ...-straat, optioneel, huisnummer postcode of (.) huisnummer postcode woonplaats optioneel. (.)

H: Patricia, wat denk je nu?

P: ((zucht)) geen idee wat er mis is. Eh: (.) Ik moet zeggen dat ik hier niet helemaal snap wat de vraag is, waarom, wat is er mis. Het is niet compleet. Hier bij postcode en postadres, je hebt of de straat met de (.) de ene heeft een straat en de ander heeft een woonplaats dus die zijn niet hetzelfde, maar (.) ik kan me niet voorstellen dat dat de vraag is.

H: Voor A kan ik je dan cadeau geven dat die niet deterministisch is.

P: Dat is precies waar ik moeite mee heb. Bepaalde (.) Poeh (.) Als je een huisnummer hebt en een postcode dan kan je bepalen of je dan in de bovenste zit met straat of in de onderste met een woonplaats. (.) Dat bedoel je met deterministisch. (.) Ja ik heb hier een beetje moeite met dat concept.

H: Ja, ga door.

P: (.) Ja ik zit bij B te kijken dat er, wat je voor, wat je bedoelt dat het semantisch equivalent is. Ik heb het idee de eerste die, je hebt alleen de straat het huisnummer en de postcode en de tweede heeft huisnummer postcode woonplaats maar hij kan allebei kan hij zonder huisnummer postcode zijn (.) Ik zou zelf in eerste instantie zeggen ik haal huisnummer en postcode ook weg maar dan is hij natuurlijk niet meer hetzelfde. (.) Ik zeg, ik zou zelf zeggen je hebt ((typt)) straat, optioneel (.) ((typt)) huisnummer, postcode, die moeten aanwezig zijn en dan heb je ((typt)) woonplaats. **Maar** die optie staat ertussen (.) Dus als je ze aan elkaar zet dan wordtie (.) nou heb ik er [onverstaanbaar] van gemaakt. (.)

H: Uhuh (.)

P: En toch (.)

H: Wat denk je nu Patricia? (.)

P: Nou, ik heb een beetje, de 'of' moet hier ergens anders neerzetten denk ik (.) die (.) Ik zie wel dat ik hem ruimer heb gemaakt door hem als één neer te zetten (.) want je kunt nu ook alle vier de dingen hebben. En dat is wat ik weg wil halen. (.) Als ik nou (.) Als ik zeg dat je de haakjes om de (.) straat huisnummer en postcode doet en de straat optioneel maakt, dan nog heb je dat woonplaats, kijk daar (.)

H: Ja, wat denk je Patricia?

P: Ik zie zo gauw niet hoe ik ze moet splitsen. Ehm: (.) Of je moet er een derde bijzetten (.) Straat huisnummer postcode ((typt)) en (.) huisnummer en postcode (.) Dan moet je er drie maken (.) ((typt)) huisnummer postcode, en dan heb je ((typt)) haakje, huisnummer [onverstaanbaar] postcode woonplaats. En dan haal ik alle vraagtekentjes weg [onverstaanbaar] die drie keuzes. Nou heb ik hem helemaal uitgeschreven. Je hebt een straat huisnummer postcode, of huisnummer postcode of huisnummer postcode woonplaats (.)

H: Ja?

P: Ja. (.) Dan zou ik zeggen dattie zo goed is.

H: Is die zo (.) Is die zo af?

P: Ja (.) ja. (.)

H: Waar denk je nu aan?

P: (.) Ja ik zit nu te bepalen of ik, als ik de derde dat ik weet (.) Ja ik zou toch denken dattie klaar is. Zo lever ik hem in, ja hij is af.

H: Hij is zo af?

P: Ja.

H: Ok, nou dan gaan we naar de (.) laatste oefening.

P: (.) Semantische equivalentheid. Ik ga eerste bepalen wat die eigenlijk doet. Hij heeft A en B of A B A B of hij is leeg (.) of A B en C. (.) Nou, ik zou denken, hij heeft ((typt)) A komma B, die heeftie altijd en die herhaalt hij, dus ((typt)) twee keer. Als die hem in dezelfde volgorde ((typt)) dus die horen wel bij elkaar. Daar zou ik een ster achter zetten want hij, oh nee. Geen ster maar een plus. Hij zit overal in, dus hij komt minstens één keer voor maar hij kan ook twee keer voorkomen. Die twee keer die [onverstaanbaar] als meer keer. En dan kan er een C erin zijn, en het geheel, na die A B kan, ehm, C komen en, heb je weer iets (.) maar het kan ook leeg zijn. [onverstaanbaar] vraagteken. Nu heb ik hem wel ruimer gemaakt, want A B kan meer keer dan twee keer voorkomen. Maar, het principe (.) is volgens mij wel gelijk. Alleen die C, die komt ook niet (.) die komt ook niet in ieder moment voor. Dus iedere keer, dus die C zelf (.) is ook optioneel. (.) Ja (.) Ja, volgens mij issie zo klaar. (.)

H: Ok. Hij is af zo?

P: Ja.

H: Of denk je nog iets?

P: Nee, volgens mij is deze klaar.

H: Nou dan sluiten we, dan sluiten we dit weg.

---

## POSTTEST PROEFPERSOON 1

---

P: Ik begin uit te schrijven welke dat er zijn. A B C komma B C of A B C D. En dan hebben we hier nog eentje, B A C. Even kijken, 1 2 3 4, ja die heb ik. En: dan ga ik ze op volgorde zetten.

((typt)) A B C ((typt)) A B C D ((typt)) B C ((typt)) en B A C. En dan kan ik bij de twee linkse A B buiten haakjes halen. En dat heeft daarbij A B C of C D. En bij die andere kan ik B buiten haakjes halen, kan ik ook al (.) de A vraagteken tussen zetten. Die A B haakje ((typt)) A B daar kan ik C buiten haakje halen door te zeggen dat er D met een vraagteken, C D met een vraagteken (.) En dan heb ik hier een fout gemaakt. Want die hebben alledrie A B C. Dus hij is veel makkelijker. Ik zet hier B met een vraagteken. Of de andere, B A vraagteken C. (.) Ja, dit is goed.

H: Maar ik zie hem, ja nu zie ik hem. Ja, en wat denk je nu?

P: Hij is klaar, dus nu zou ik eigenlijk even moeten kijken van welke mogelijkheden heeft hij. En dat is A B C of de D erbij A B C D of B zonder A C en nu met A C. En dan denk ik dat ik ze allemaal, dan denk ik dat hij goed is. (.)

H: Ja?

P: Ja

H: Ok. Keurig. Ik vind het heel grappig om te zien dat er inderdaad een heel ander denkproces bij je gaande is wat veel meer gestroomlijnd is.

P: Het is veel minder chaotisch.

(...)

P: Ehm, ik begin gewoon met het uitschrijven, dat is toch het makkelijkste.

[onverstaanbaar] ((typt)) En dan moet ik even kijken want alleen C of D C ((typt)) (onverstaanbaar) alleen een D, alleen een C, een D en een C. Dit is [onverstaanbaar] tussen haakjes C, B C haakje vraagteken. En de D is al dan niet aanwezig ((typt)) dus dan wordt het D vraagteken C en het geheel mag ook leeg zijn. En dan [onverstaanbaar]. De D hoeft er niet te zijn en dan is het C of de D is er wel met een D C. (.) [onverstaanbaar]. Ja.

H: Ja, hij is volgens mij ook goed. Ja dat ging best, hoe vond je het gaan?



---

## PRETEST PROEFPERSOON 2

---

W: Goed, nou het is dus neem ik aan hier de bedoeling dat ik er dus een mooie DTD van ga maken waar dit, deze XML aan moet voldoen. Ik zie dus hier een, dat ik hier eigenlijk eh: vier verschillende soorten: XML heb waarbij er eentje alleen een president heeft, eentje alleen een koning en eentje alleen een koningin en dat ook nog de combinatie koning koningin mogelijk is. Ik moet dus in feite een hele mooie DTD maken waarbij dus: een: element eh: voorkomt. Dus een element land. Wat dus op verschillende manieren onderverdeeld kan zijn. Dus als ik dan ga beginnen, ik weet niet of het nou de bedoeling is dat ik de hele DTD van het begin af aan ga zitten schrijven.

H: [praten door elkaar] content model van het element land waarvan je hier dus vier instanties hebt.

W: Ok dus dat eh, dan schrijf je dus een element wat dus land heet. ((typt)) En dat is dus opgebouwd of uit een elementtype president of een elementtype koning, maar dat moet 'of' zijn dus ik moet geen komma zetten maar ik moet dat dus als een 'of' schrijven. Of een-

H: Ja je tikt het-

W: Ja ik ben het aan het intikken.

H: Ok dan zie ik het vanzelf verschijnen.

W: Of een koning, of dus een koning gevolgd door een koningin of een koningin alleen. Ik heb eerlijk gezegd het vermoeden dat het ook nog wel makkelijker kan (.) maar om te beginnen is dit denk ik niet zo gek dus dan geef ik aan dat het element zo opgebouwd is. Dan moet ik daarna uiteraard ook de deelelementen, die moeten ook apart gedefinieerd zijn. Waarbij dus het element president, dat is gewoon een heel simpel element van het type PC data.

H: Dat hoeft je niet te doen.

W: Die hoeft ik allemaal niet aan te geven, ok. Dus dan zou ik het dus op deze manier zou ik het aangeven.

H: Dit is het model zoals het ook juist is.

W: Uhuh. Ja ik heb hier dus mijn element land wat dus vier verschillende mogelijkheden heeft. Of een president of een koning of een koning en koningin of alleen een koningin waarbij ik nou dus de onderdelen apart moet specificeren.

H: Dit is de oefenopgave nog [onverstaanbaar]

W: Je (.) Je geluid valt even weg, ik heb je niet gehoord.

H: Dit is de oefenopgave nog [onverstaanbaar]. Is dit model deterministisch?

W: Eh, dit model is niet deterministisch want op het moment dat ik een koning tegen kom dan weet ik nog niet of ik het tweede of het derde geval hier te pakken heb (.) Maar ik kan daarna dus of een koningin hebben of niet.

H: Nee, en dus?

W: En dus zou ik het deterministisch moeten maken, inderdaad. En hoe zou ik dat dan doen? (.)

H: Wat denk je nu?

W: Nou ik zit op het ogenblik, ik zit gewoon te kijken en ik denk eerlijk gezegd vrij weinig. (.)

H: Is het nu nog stil in je hoofd, of?

W: Het is nog stil in mijn hoofd inderdaad. Dat, ik zie zo gauw geen mogelijkheid om dit mooi deterministisch te maken. (.) Ja want als ik een eh: Ik zou hooguit kunnen zeggen van een eh: zowel de koning als, dat ik er een element van maak waarbij ik maar drie mogelijkheden bij heb. Eh: waarbij ik dus zeg ik heb dus of alleen een president, nee wacht even ik heb maar twee mogelijkheden dan. Ik heb of een president, dus dan zou ik dit krijgen, nee wacht even (.) En dan heb ik dus of een president, of een: eh: of (.) een koning die nul of één keer kan voorkomen. Dit geef je aan met een vraagtekentje erachter dacht ik. Gevolgd door een koningin die nul of één keer kan voorkomen en dan zou ik dus dit krijgen. En dan is het dus volgens mij wel deterministisch geworden.

H: Zijn er nog andere dingen waar je aan denkt?

W: Nee, dus eh: verder niks anders waar ik op het ogenblik aan denk. (.)

H: [onverstaanbaar]

W: Sorry?

H: Hiermee is deze opgave voor jou afgerond?

W: Eh: (.) Ja. Tenminste, ja wat ik in feite zou moeten doen is het inderdaad als een volledige DTD formuleren. Dit element XML ook als XML neerzetten en kijken of het goedgekeurd wordt, of dat door de parsing heen komt.

H: Zullen we deze afsluiten?

W: Prima.

H: Hoe gaat het? Lastiger allemaal hardop?

W: Nou ik blijf die brom op de achtergrond heel erg storend vinden.

H: Dat heeft met mijn microfoon te maken, maar dat [onverstaanbaar]. Ben je toe aan de volgende opgave?

W: Ja hoor.

H: Ok dan gaan we

W: Ik zal hem even opzoeken.

H: Lees even goed, en het gaat hier ook steeds weer om alleen het content model van dit geval het element letters.

W: ((zucht))

H: Dus als je, dit kunnen we nu [onverstaanbaar]. Dan zet ik mijn microfoon uit ja?

W: Ja ok.

H: Dus jij kan nu beginnen met de opdracht.

W: Ok goed ik zie hier dus drie instanties van het element letters waarbij de eerste (.) eh: gewoon helemaal leeg is. De tweede bevat alleen een, bevat een A en de derde bevat een B of een A (.) waarbij A en B dus ja ook weer elementen zijn. Eh: Dat wil dus zeggen dat eh, in dit geval kan ik hieruit afleiden dat als er een B is dan is er ook een A, maar eh: en dan staat er een A na. Maar de (.) maar de A kan alleen voorkomen, de B kan niet alleen voorkomen. Wat zal ik hier zeggen? Ik zou zeggen van, Ik krijg hier dus weer een, een element (.) wat letters heet. En dat kan dus zijn of empty, en als het niet empty is dan zit er in elk geval een A in en dan zit er dus al dan niet een B voor dus dat de B alleen kan niet voorkomen dus dat betekent dat ik aan moet geven dat ik een B heb met een vraagtekentje, gevolgd door een A. A is 0 of één keer, nee is precies één keer en dat vraagtekentje is nul of één dus dan zou ik hiermee, zou ik dit gedefinieerd hebben. Ik hier dus mee óf een leeg element samenstellen, of een element wat alleen een A bevat of een element wat een B bevat gevolgd door een A. (.) Waarmee ik denk dat dit verder opgelost is.

H: Je bent ermee klaar?

W: Ja.

H: [onverstaanbaar] de derde opgave ook nog [onverstaanbaar]. Ik zal de microfoon weer uitzetten en een nieuwe pagina maken.

W: Dan zal ik even naar de derde opgave gaan kijken. (.)

H: Hierbij hetzelfde, je kunt voor straat een s en postcode een p zetten.

W: Ok, er wordt gevraagd wat is er mis met het contentmodel van het element postadres. Hier het element postadres, ja wat mij hier (.) opvalt is dat er dubbele haakjes staan. Wacht even, nee er staat hier, ik heb hier twee mogelijkheden. Dat is 1 (.) eh ik heb twee mogelijkheden. Of ik heb een straat die nul of één keer voorkomt, een huisnummer en een postcode, of ik heb een huisnummer een postcode en een woonplaats. Ehm: Het probleem hierbij is dat de straat omdat hij niet voor hoeft te komen kan ik dus beginnen met een huisnummer waardoor het model hier even niet deterministisch is. Hoe zou ik dat hier kunnen oplossen? Nou ik zou dit heel makkelijk (.) nou, niet zo makkelijk. Je zou kunnen zeggen ik maak er maar één mogelijkheid van (.) Maar hiermee is wel duidelijk uitgesloten dat ik een adres krijg wat zowel een straat als een woonplaats bevat. Dus die dingen zou je wel moeten uitsluiten. Ehm: ik heb in elk geval een huisnummer en een postcode (.) je zou dus aan moeten geven op de een of andere manier dat het voorkomen van de straat zou moeten uitsluiten er een woonplaats is. En woonplaats dat het uitsluit dat er een straat is. Dus als de ene nul of één keer voorkomt dat de ander dan één of nul keer voorkomt. Hoe je dat nou aan kan geven in een DTD zou ik zo gauw niet weten. (.)

H: Ik kan je verklappen dat het ook niet [onverstaanbaar] is [onverstaanbaar]

W: Ok, daar zal ik een andere oplossing voor moeten verzinnen. (.) Nja het enige, je kan het gewoon altijd heel makkelijk uitschrijven waarbij je dus vier mogelijkheden krijgt. (.) En dan ben ik er nog niet want dan is het nog steeds niet deterministisch. (.) Dan zou ik dus, moet ik, ik heb, ik kan zeggen van nou ik heb twee mogelijkheden in feite. Ik begin met mijn straat of ik begin met een huisnummer. En dan, als ik met een straat begin dan heb ik daarna een huisnummer en een postcode en als ik met een huisnummer begin dan heb ik in elk geval een postcode en al dan niet een woonplaats. En ik denk dat ik daarmee wel deterministisch ben. Dus dan ((typt)) zal ik eens gaan kijken als ik het ga opschrijven of ik er dan wat duidelijker uit kan komen. ((typt)) Nou als ik een straat heb, dan heb ik daarna in elk geval een huisnummer en een postcode. Maar als ik dat niet heb, wat heb ik dan? Dan heb ik een huisnummer ((typt)) Dan heb ik het huisnummer en in elk geval een postcode ((typt)) en na de postcode dan heb ik dus al dan niet een woonplaats. Wacht even, het valt van mijn scherm af heb ik het idee. (.)

H: Ja je kunt het afkorten hè, met 's' 'h' en 'p'.

W: Het valt er wel af, dus je krijgt dan eh, ik tik het even opnieuw, dan wordt het wat beter leesbaar. (.) En dan zetten we op de volgende regel, dan staat daar dus of ((typt)) een huisnummer, [onverstaanbaar] in elk geval een postcode en dan krijg ik dus al dan niet een woonplaats en dan ((typt)) denk ik dat ik hiermee, nee wacht even ik ben er nog niet, ik heb, nee dan ben ik er wel (.) het is semantisch hetzelfde. Ik heb hier de mogelijkheden dat ik een straat huisnummer en postcode heb, ik heb de mogelijkheid dat ik alleen huisnummer en postcode heb en ik heb de mogelijkheid dat ik een huisnummer postcode en woonplaats heb dus ik denk dat ik hiermee alle mogelijkheden die het gegeven postadres geeft dat ik die toch gedekt heb.

H: Eh, ben je klaar met deze?

W: Ja, ik ben hier wel mee klaar denk ik.

H: Ok, dankjewel. Ik vind dat je het nog aardig gedaan hebt, vooral die laatste was erg moeilijk.

---

## POSTTEST PROEFPERSOON 2

---

W: Even kijken als ik hier begin en dus als we de mogelijkheden allemaal uit gaan schrijven. A B A B of A B of een lege daar gebruik ik een euro voor, want die stelt toch weinig voor, en als laatste A B A B. Dus hier beginnen we mee. Jij ziet wat ik schrijf?

H: Ja.

W: Dus wat we hiermee kunnen doen ehm, is die, wacht even hoor, als ik hieronder ga schrijven, ok dan als ik die lege, ik doe even twee stappen tegelijk anders moet ik zo veel schrijven. Wat ik doe ik haal die lege er gewoon af en dan kan ik dit schrijven op deze manier. En dan zetten we hier een vraagtekentje achter van die is er of één keer of niet. Dan ben ik dat lege ding dus kwijt. Wat ik dan hiermee kan doen is een hele mooie A B buiten haakjes halen en dan krijg ik binnen de haakjes hier een A B A B, dan krijg ik of een lege of een A B en dit haakje en dit vraagtekentje blijft gewoon staan. ((typt)) Dan kan ik op diezelfde manier dus weer die lege erbij halen en dan kan ik nog een keer (.) de (.) A B erbuiten gaan halen en als ik dan doorga als het goed is dan hou ik dit over. ((typt))

H: Ja je, wat denk je nu?

W: Ja dit was het volgens mij?

H: Dit was het? Je bent nu helemaal klaar met de opgave?

W: Ja ik denk dat dit het is. Ja ik ben klaar met het oplossen, het enige wat we nog moeten doen is kijken of inderdaad alles dekt.

H: Ja is prima, ga maar door, dat is een essentieel onderdeel

W: Ja nou, ja uiteraard. Maar dus de eerste, A B A B A B zit er duidelijk dus in (.) Ja, de A B van die tweede die zit er ook in. Als ik even, als ik de voorlaatste en het daarvoorstaande vraagtekentje weglaat zeg maar dan hou ik dus A B over. De lege die zit er ook in voor het laatste vraagtekentje [onverstaanbaar] en de A B A B zit er ook in. (.) Als ik dit tweede vraagtekentje: ja, ze zijn er allemaal.

H: Ja ok, dat is correct.

## PDCA CYCLUS

---

### PRETEST PROEFPERSOON 1

---

P: Ik zie eh . vier elementen en (.) de ene heeft president en de andere (.) hebben koning, koningin en nog eentje koning. Ehm (.) content zou ik dus verwachten dat de president is optioneel kan [onverstaanbaar] bepalen voor president of koning (.) wel koning koningin. Die zitten achter elkaar. Dus dan zou ik (.) dan begin ik het op te schrijven. (.) En dan haakje openen (.) ehm (.) ((typt)) president ((typt)) komma koning

H: Je mag trouwens ook wel afkortingen nemen hoor want het wordt misschien wel erg lang als je het achter elkaar moet intikken.

J: Oh ja, dat kan ook. Ehm en dan zet ik er de (.) optionele dingen achter eh. Ja een beetje raar want ze zijn allemaal optioneel.((typt)) Dat is wat ik dan in eerste instantie denk. Die koning en koningin die horen bij elkaar (.) en ze komen allemaal maar één keer voor (.) Ik zou wel zeggen van [onverstaanbaar] president en de koning of koningin je hebt ze niet tegelijk (.)

H: Ja, ga door, met denken.

P: Dus nu heb ik ze allemaal los optioneel (.) dus (.) ehm: (.) je hebt de keuze, het moeten eigenlijk keuzes zijn. (.) Maar dan is het een keuze tussen koning koningin, die kunnen wel bij elkaar zijn, en hier hoort dan een 'of' tussen. (.) ((typt)) zoiets zou ik denken. Ze zijn [onverstaanbaar] alle drie optioneel (.) maar dat zou betekenen dat er niks in kan staan. Dat zou [onverstaanbaar] natuurlijk mogelijk zijn, maar (.) de vraagtekentjes staan verkeerd.

H: Uhuh

J: (.) ((typt)) ow (.) dat krijg ik [onverstaanbaar] geprutst. (.) Hij moet aanwezig zijn dus het geheel moet iets hebben (.) Ja de vraagtekentjes moeten weg misschien wel. Even kijken (.) dat kan ik niet aangeven, dus of die of de andere dus (.) nou ik zal hem toch zo laten staan denk ik. Het is niet goed met die vraagtekens maar ik weet eigenlijk wat ik eraan moet veranderen.

H: Sorry, wat zei je als laatste?

P: Nou het is volgens mij niet goed maar ik weet zo gauw niet wat ik eraan moet veranderen. (.) Dat moet ik opzoeken. (.)

H: Moet je iets opzoeken?

P: Nou nee nu niet eh: (.)

H: Maar je bent nu klaar met de opgave wat jou betreft?

P: Ja ik zou m even zo laten staan. Er is iets niet goed maar ik weet even niet precies wat, eh: (.) Echt, ja (.) Ik heb nu het idee van hij kan helemaal leeg zijn, en dat is niet eh niet de bedoeling eh (.)

H: Uhuh (.) ja (.) Wat denk je dan?

P: Ja, ik ben, je moet (.) ehm hier een haakje achter zetten en dan (.) Je hebt een president of een koning en een koningin, die kunnen dan wel allebei tegelijk aanwezig zijn. (.) Als ik er helemaal haakjes omheen zet (.) dan is er in ieder geval een van de twee aanwezig. Nu heb ik hem bij koning weggehaald ((typt)) Dat kan ook niet (.) Nou ik weet het niet ((typt)) Ik denk dat ik hem zo laat. Als ik het geheel tussen haakjes zet, dan blijft er eh: altijd eentje (.) En de dingen

die er tussenin staan zijn in principe allemaal optioneel (.) Ja of het is een keuze, en daar kan er maar eentje van zijn (.) oh dat kan, ja vast wel. ((typt)) Zo hoortie, ik zette er teveel vraagtekentjes in. Je hebt de president of een koning en een koningin en dan zou die koning en die koningin, die zou een van de twee kan dat zijn. (.) [onverstaanbaar] Dan krijg je er drie eh: (.) Of ja, dan kom ik weer met die deterministische (.) ((typt)) Zo hoortie. Die zijn ook [onverstaanbaar] (.) Even kijk of ik nu de haakjes goed heb. Het idee is dat een land met een president of (.) een konings eh: huis. En dan heb je een koning met een koningin of eigenlijk alleen de koning of alleen de koningin (.)

H: Uhuh

P: Zo laat ik hem

H: Ja hij is klaar?

P: Zo zou ik denken dat het eh: moest zijn ja. De haakjes kloppen nog niet zie ik. ((typt)) die horen niet tussen haakjes. (.) Zo.

H: Ja?

P: Ja, zo hoortie denk ik

H: Ok. Dan gaan we naar de tweede opgaven. Ik zal daarvoor een nieuwe pagina maken. Die heb je hier. En het is hetzelfde recept dus het woord is aan jou. Eh, je ziet hier eigenlijk weer een zelfde soort termen, nu drie instanties van het element letters en ook hier is de vraag om het bijbehorende content model te maken dus je kan ook hier gelijk beginnen met haakje openen haakje sluiten en dat wordt het content model dus letters hoeft je ook niet eens op te nemen. (.)

P: Hij kan leeg zijn (.) Dat betekent dat er ook een eh (.) [onverstaanbaar] sterretje aan het eind, buiten de haakjes (.) A, en dan (.) A zonder B en B voor een A dus we maken B optioneel en A is in allebei aanwezig dus een A, en dat is er maar één dus die hoeft geen dingetje. [onverstaanbaar] (.) Dit is klaar.

H: Dit is m?

P: Ja (.)

H: Kan je mij nog wat meer uitleg geven? Want ik hoorde je wel wat zeggen, maar in een keer staat, plop, deze expressie op het scherm die ik niet helemaal kan knopen aan jou gedachtegang.

P: Ik denk hij kan leeg zijn, dus het content model staat een sterretje achter, nul of meer keer. (.) Ja hij is niet meer keer trouwens. Het is een vraagteken. Ik pas m aan. ((typt)) Want het geheel is optioneel. Hij kan dan leeg zijn en er kan B, die mag aanwezig zijn maar die hoeft niet aanwezig te zijn. Daarom een B met een vraagteken erachter. En het volgende element is een A, als tweede element. (.) [onverstaanbaar]

H: Uhuh, ben je klaar?

P: Ja

H: Ok, dan gaan we naar de volgende. Het is misschien wel handig voor deze opgave, iets soortgelijks, je krijgt hier al een content model. Je mag gewoon 'straat' met een s'je afkorten en huisnummer met een -h. (.)

P: (.) De eerste vraag is wat we mis[onverstaanbaar] met postadres ...-straat, optioneel, huisnummer postcode of (.) huisnummer postcode woonplaats optioneel. (.)

H: Patricia, wat denk je nu?

P: ((zucht)) geen idee wat er mis is. Eh: (.) Ik moet zeggen dat ik hier niet helemaal snap wat de vraag is, waarom, wat is er mis. Het is niet compleet. Hier bij postcode en postadres, je hebt of de straat met de (.) de ene heeft een straat en de ander heeft een woonplaats dus die zijn niet hetzelfde, maar (.) ik kan me niet voorstellen dat dat de vraag is.

H: Voor A kan ik je dan cadeau geven dat die niet deterministisch is.

P: Dat is precies waar ik moeite mee heb. Bepaalde (.) Poeh (.) Als je een huisnummer hebt en een postcode dan kan je bepalen of je dan in de bovenste zit met straat of in de onderste met een woonplaats. (.) Dat bedoel je met deterministisch. (.) Ja ik heb hier een beetje moeite met dat concept.

H: Ja, ga door.

P: (.) Ja ik zit bij B te kijken dat er, wat je voor, wat je bedoelt dat het semantisch equivalent is. Ik heb het idee de eerste die, je hebt alleen de straat het huisnummer en de postcode en de tweede heeft huisnummer postcode woonplaats maar hij kan allebei kan hij zonder huisnummer postcode zijn (.) Ik zou zelf in eerste instantie zeggen ik haal huisnummer en postcode ook weg maar dan is hij natuurlijk niet meer hetzelfde. (.) Ik zeg, ik zou zelf zeggen je hebt ((typt)) straat, optioneel (.) ((typt)) huisnummer, postcode, die moeten aanwezig zijn en dan heb je ((typt)) woonplaats. **Maar** die optie staat ertussen (.) Dus als je ze aan elkaar zet dan wordtie (.) nou heb ik er [onverstaanbaar] van gemaakt. (.)

H: Uhuh (.)

P: En toch (.)

H: Wat denk je nu Patricia? (.)

P: Nou, ik heb een beetje, de 'of' moet hier ergens anders neerzetten denk ik (.) die (.) Ik zie wel dat ik hem ruimer heb gemaakt door hem als één neer te zetten (.) want je kunt nu ook alle vier de dingen hebben. En dat is wat ik weg wil halen. (.) Als ik nou (.) Als ik zeg dat je de haakjes om de (.) straat huisnummer en postcode doet en de straat optioneel maakt, dan nog heb je dat woonplaats, kijk daar (.)

H: Ja, wat denk je Patricia?

P: Ik zie zo gauw niet hoe ik ze moet splitsen. Ehm: (.) Of je moet er een derde bijzetten (.) Straat huisnummer postcode ((typt)) en (.) huisnummer en postcode (.) Dan moet je er drie maken (.) ((typt)) huisnummer postcode, en dan heb je ((typt)) haakje, huisnummer [onverstaanbaar] postcode woonplaats. En dan haal ik alle vraagtekentjes weg [onverstaanbaar] die drie keuzes. Nou heb ik hem helemaal uitgeschreven. Je hebt een straat huisnummer postcode, of huisnummer postcode of huisnummer postcode woonplaats (.)

H: Ja?



P: Ja. (.) Dan zou ik zeggen dattie zo goed is.

H: Is die zo (.) Is die zo af?

P: Ja (.) ja. (.)

H: Waar denk je nu aan?

P: (.) Ja ik zit nu te bepalen of ik, als ik de derde dat ik weet (.) Ja ik zou toch denken dattie klaar is. Zo lever ik hem in, ja hij is af.

H: Hij is zo af?

P: Ja.

H: Ok, nou dan gaan we naar de (.) laatste oefening.

P: (.) Semantische equivalentheid. Ik ga eerste bepalen wat die eigenlijk doet. Hij heeft A en B of A B A B of hij is leeg (.) of A B en C. (.) Nou, ik zou denken, hij heeft ((typt)) A komma B, die heeftie altijd en die herhaalt hij, dus ((typt)) twee keer. Als die hem in dezelfde volgorde ((typt)) dus die horen wel bij elkaar. Daar zou ik een ster achter zetten want hij, oh nee. Geen ster maar een plus. Hij zit overal in, dus hij komt minstens één keer voor maar hij kan ook twee keer voorkomen. Die twee keer die [onverstaanbaar] als meer keer. En dan kan er een C erin zijn, en het geheel, na die A B kan, ehm, C komen en, heb je weer iets (.) maar het kan ook leeg zijn. [onverstaanbaar] vraagteken. Nu heb ik hem wel ruimer gemaakt, want A B kan meer keer dan twee keer voorkomen. Maar, het principe (.) is volgens mij wel gelijk. Alleen die C, die komt ook niet (.) die komt ook niet in ieder moment voor. Dus iedere keer, dus die C zelf (.) is ook optioneel. (.) Ja (.) Ja, volgens mij issie zo klaar. (.)

H: Ok. Hij is af zo?

P: Ja.

H: Of denk je nog iets?

P: Nee, volgens mij is deze klaar.

H: Nou dan sluiten we, dan sluiten we dit weg.

---

## POSTTEST PROEFPERSOON 1

---

P: Ik begin uit te schrijven welke dat er zijn. A B C komma B C of A B C D. En dan hebben we hier nog eentje, B A C. Even kijken, 1 2 3 4, ja die heb ik. En: dan ga ik ze op volgorde zetten. ((typt)) A B C ((typt)) A B C D ((typt)) B C ((typt)) en B A C. En dan kan ik bij de twee linkse A B buiten haakjes halen. En dat heeft daarbij A B C of C D. En bij die andere kan ik B buiten haakjes halen, kan ik ook al (.) de A vraagteken tussen zetten. Die A B haakje ((typt)) A B daar kan ik C buiten haakje halen door te zeggen dat er D met een vraagteken, C D met een vraagteken (.) En dan heb ik hier een fout gemaakt. Want die hebben alledrie A B C. Dus hij is veel makkelijker. Ik zet hier B met een vraagteken. Of de andere, B A vraagteken C. (.) Ja, dit is goed.

H: Maar ik zie hem, ja nu zie ik hem. Ja, en wat denk je nu?

P: Hij is klaar, dus nu zou ik eigenlijk even moeten kijken van welke mogelijkheden heeft hij. En dat is A B C of de D erbij A B C D of B zonder A C en nu met A C. En dan denk ik dat ik ze allemaal, dan denk ik dat hij goed is. (.)

H: Ja?

P: Ja

H: Ok. Keurig. Ik vind het heel grappig om te zien dat er inderdaad een heel ander denkproces bij je gaande is wat veel meer gestroomlijnd is.

P: Het is veel minder chaotisch.

(...)

P: Ehm, ik begin gewoon met het uitschrijven, dat is toch het makkelijkste. [onverstaanbaar] ((typt)) En dan moet ik even kijken want alleen C of D C ((typt)) (onverstaanbaar) alleen een D, alleen een C, een D en een C. Dit is [onverstaanbaar] tussen haakjes C, B C haakje vraagteken. En de D is al dan niet aanwezig ((typt)) dus dan wordt het D vraagteken C en het geheel mag ook leeg zijn. En dan [onverstaanbaar]. De D hoeft er niet te zijn en dan is het C of de D is er wel met een D C. (.) [onverstaanbaar]. Ja.

H: Ja, hij is volgens mij ook goed. Ja dat ging best, hoe vond je het gaan?

---

## PRETEST PROEFPERSOON 2

---

W: Goed, nou het is dus neem ik aan hier de bedoeling dat ik er dus een mooie DTD van ga maken waar dit, deze XML aan moet voldoen. Ik zie dus hier een, dat ik hier eigenlijk eh: vier verschillende soorten: XML heb waarbij er eentje alleen een president heeft, eentje alleen een koning en eentje alleen een koningin en dat ook nog de combinatie koning koningin mogelijk is. Ik moet dus in feite een hele mooie DTD maken waarbij dus: een: element eh: voorkomt. Dus een element land. Wat dus op verschillende manieren onderverdeeld kan zijn. Dus als ik dan ga beginnen, ik weet niet of het nou de bedoeling is dat ik de hele DTD van het begin af aan ga zitten schrijven.

H: [praten door elkaar] content model van het element land waarvan je hier dus vier instanties hebt.

W: Ok dus dat eh, dan schrijf je dus een element wat dus land heet. ((typt)) En dat is dus opgebouwd of uit een elementtype president of een elementtype koning, maar dat moet 'of' zijn dus ik moet geen komma zetten maar ik moet dat dus als een 'of' schrijven. Of een-

H: Ja je tikt het-

W: Ja ik ben het aan het intikken.

H: Ok dan zie ik het vanzelf verschijnen.

W: Of een koning, of dus een koning gevolgd door een koningin of een koningin alleen. Ik heb eerlijk gezegd het vermoeden dat het ook nog wel makkelijker kan (.) maar om te beginnen is dit denk ik niet zo gek dus dan geef ik aan dat het element zo opgebouwd is. Dan moet ik daarna uiteraard ook de deelelementen, die moeten ook apart gedefinieerd zijn. Waarbij dus het element president, dat is gewoon een heel simpel element van het type PC data.

H: Dat hoeft je niet te doen.

W: Die hoeft ik allemaal niet aan te geven, ok. Dus dan zou ik het dus op deze manier zou ik het aangeven.

H: Dit is het model zoals het ook juist is.

W: Uhuh. Ja ik heb hier dus mijn element land wat dus vier verschillende mogelijkheden heeft. Of een president of een koning of een koning en koningin of alleen een koningin waarbij ik nou dus de onderdelen apart moet specificeren.

H: Dit is de oefenopgave nog [onverstaanbaar]

W: Je (.) Je geluid valt even weg, ik heb je niet gehoord.

H: Dit is de oefenopgave nog [onverstaanbaar]. Is dit model deterministisch?

W: Eh, dit model is niet deterministisch want op het moment dat ik een koning tegen kom dan weet ik nog niet of ik het tweede of het derde geval hier te pakken heb (.) Maar ik kan daarna dus of een koningin hebben of niet.

H: Nee, en dus?

W: En dus zou ik het deterministisch moeten maken, inderdaad. En hoe zou ik dat dan doen? (.)

H: Wat denk je nu?

W: Nou ik zit op het ogenblik, ik zit gewoon te kijken en ik denk eerlijk gezegd vrij weinig. (.)

H: Is het nu nog stil in je hoofd, of?

W: Het is nog stil in mijn hoofd inderdaad. Dat, ik zie zo gauw geen mogelijkheid om dit mooi deterministisch te maken. (.) Ja want als ik een eh: Ik zou hooguit kunnen zeggen van een eh: zowel de koning als, dat ik er een element van maak waarbij ik maar drie mogelijkheden bij heb. Eh: waarbij ik dus zeg ik heb dus of alleen een president, nee wacht even ik heb maar twee mogelijkheden dan. Ik heb of een president, dus dan zou ik dit krijgen, nee wacht even (.) En dan heb ik dus of een president, of een: eh: of (.) een koning die nul of één keer kan voorkomen. Dit geef je aan met een vraagtekentje erachter dacht ik. Gevolgd door een koningin die nul of één keer kan voorkomen en dan zou ik dus dit krijgen. En dan is het dus volgens mij wel deterministisch geworden.

H: Zijn er nog andere dingen waar je aan denkt?

W: Nee, dus eh: verder niks anders waar ik op het ogenblik aan denk. (.)

H: [onverstaanbaar]

W: Sorry?

H: Hiermee is deze opgave voor jou afgerond?

W: Eh: (.) Ja. Tenminste, ja wat ik in feite zou moeten doen is het inderdaad als een volledige DTD formuleren. Dit element XML ook als XML neerzetten en kijken of het goedgekeurd wordt, of dat door de parsing heen komt.

H: Zullen we deze afsluiten?

W: Prima.

H: Hoe gaat het? Lastiger allemaal hardop?

W: Nou ik blijf die brom op de achtergrond heel erg storend vinden.

H: Dat heeft met mijn microfoon te maken, maar dat [onverstaanbaar]. Ben je toe aan de volgende opgave?

W: Ja hoor.

H: Ok dan gaan we

W: Ik zal hem even opzoeken.

H: Lees even goed, en het gaat hier ook steeds weer om alleen het content model van dit geval het element letters.

W: ((zucht))

H: Dus als je, dit kunnen we nu [onverstaanbaar]. Dan zet ik mijn microfoon uit ja?

W: Ja ok.

H: Dus jij kan nu beginnen met de opdracht.

W: Ok goed ik zie hier dus drie instanties van het element letters waarbij de eerste (.) eh: gewoon helemaal leeg is. De tweede bevat alleen een, bevat een A en de derde bevat een B of een A (.) waarbij A en B dus ja ook weer elementen zijn. Eh: Dat wil dus zeggen dat eh, in dit geval kan ik hieruit afleiden dat als er een B is dan is er ook een A, maar eh: en dan staat er een A na. Maar de (.) maar de A kan alleen voorkomen, de B kan niet alleen voorkomen. Wat zal ik hier zeggen? Ik zou zeggen van, Ik krijg hier dus weer een, een element (.) wat letters heet. En dat kan dus zijn of empty, en als het niet empty is dan zit er in elk geval een A in en dan zit er dus al dan niet een B voor dus dat de B alleen kan niet voorkomen dus dat betekent dat ik aan moet geven dat ik een B heb met een vraagtekentje, gevolgd door een A. A is 0 of één keer, nee is precies één keer en dat vraagtekentje is nul of één **dus dan zou ik hiermee, zou ik dit gedefinieerd hebben. Ik hier dus mee óf een leeg element samenstellen, of een element wat alleen een A bevat of een element wat een B bevat gevolgd door een A. (.)** Waarmee ik denk dat dit verder opgelost is.

H: Je bent ermee klaar?

W: Ja.

H: [onverstaanbaar] de derde opgave ook nog [onverstaanbaar]. Ik zal de microfoon weer uitzetten en een nieuwe pagina maken.

W: Dan zal ik even naar de derde opgave gaan kijken. (.)

H: Hierbij hetzelfde, je kunt voor straat een s en postcode een p zetten.

W: Ok, er wordt gevraagd wat is er mis met het contentmodel van het element postadres. Hier het element postadres, ja wat mij hier (.) opvalt is dat er dubbele haakjes staan. Wacht even, nee er staat hier, ik heb hier twee mogelijkheden. Dat is 1 (.) eh ik heb twee mogelijkheden. Of ik heb een straat die nul of één keer voorkomt, een huisnummer en een postcode, of ik heb een huisnummer een postcode en een woonplaats. Ehm: Het probleem hierbij is dat de straat omdat hij niet voor hoeft te komen kan ik dus beginnen met een huisnummer waardoor het model hier even niet deterministisch is. Hoe zou ik dat hier kunnen oplossen? Nou ik zou dit heel makkelijk (.) nou, niet zo makkelijk. Je zou kunnen zeggen ik maak er maar één mogelijkheid van (.) Maar hiermee is wel duidelijk uitgesloten dat ik een adres krijg wat zowel een straat als een woonplaats bevat. Dus die dingen zou je wel moeten uitsluiten. Ehm: ik heb in elk geval een huisnummer en een postcode (.) je zou dus aan moeten geven op de een of andere manier dat het voorkomen van de straat zou moeten uitsluiten er een woonplaats is. En woonplaats dat het uitsluit dat er een straat is. Dus als de ene nul of één keer voorkomt dat de ander dan één of nul keer voorkomt. Hoe je dat nou aan kan geven in een DTD zou ik zo gauw niet weten. (.)

H: Ik kan je verklappen dat het ook niet [onverstaanbaar] is [onverstaanbaar]

W: Ok, daar zal ik een andere oplossing voor moeten verzinnen. (.) Nja het enige, je kan het gewoon altijd heel makkelijk uitschrijven waarbij je dus vier mogelijkheden krijgt. (.) En dan ben ik er nog niet want dan is het nog steeds niet deterministisch. (.) Dan zou ik dus, moet ik, ik heb, ik kan zeggen van nou ik heb twee mogelijkheden in feite. Ik begin met mijn straat of ik begin met een huisnummer. En dan, als ik met een straat begin dan heb ik daarna een huisnummer en een postcode en als ik met een huisnummer begin dan heb ik in elk geval een postcode en al dan niet een woonplaats. En ik denk dat ik daarmee wel deterministisch ben. Dus dan ((typt)) zal ik eens gaan kijken als ik het ga opschrijven of ik er dan wat duidelijker uit kan komen. ((typt)) Nou als ik een straat heb, dan heb ik daarna in elk geval een huisnummer en een postcode. Maar als ik dat niet heb, wat heb ik dan? Dan heb ik een huisnummer ((typt)) Dan heb ik het huisnummer en in elk geval een postcode ((typt)) en na de postcode dan heb ik dus al dan niet een woonplaats. Wacht even, het valt van mijn scherm af heb ik het idee. (.)

H: Ja je kunt het afkorten hè, met 's' 'h' en 'p'.

W: Het valt er wel af, dus je krijgt dan eh, ik tik het even opnieuw, dan wordt het wat beter leesbaar. (.) En dan zetten we op de volgende regel, dan staat daar dus of ((typt)) een huisnummer, [onverstaanbaar] in elk geval een postcode en dan krijg ik dus al dan niet een woonplaats en dan ((typt)) denk ik dat ik hiermee, nee wacht even ik ben er nog niet, ik heb, nee dan ben ik er wel (.) het is semantisch hetzelfde. Ik heb hier de mogelijkheden dat ik een straat huisnummer en postcode heb, ik heb de mogelijkheid dat ik alleen huisnummer en postcode heb en ik heb de mogelijkheid dat ik een huisnummer postcode en woonplaats heb dus ik denk dat ik hiermee alle mogelijkheden die het gegeven postadres geeft dat ik die toch gedekt heb.

H: Eh, ben je klaar met deze?

W: Ja, ik ben hier wel mee klaar denk ik.

H: Ok, dankjewel. Ik vind dat je het nog aardig gedaan hebt, vooral die laatste was erg moeilijk.

---

## POSTTEST PROEFPERSOON 2

---

W: Even kijken als ik hier begin en dus als we de mogelijkheden allemaal uit gaan schrijven. A B A B of A B of een lege daar gebruik ik een euro voor, want die stelt toch weinig voor, en als laatste A B A B. Dus hier beginnen we mee. Jij ziet wat ik schrijf?

H: Ja.

W: Dus wat we hiermee kunnen doen ehm, is die, wacht even hoor, als ik hieronder ga schrijven, ok dan als ik die lege, ik doe even twee stappen tegelijk anders moet ik zo veel schrijven. Wat ik doe ik haal die lege er gewoon af en dan kan ik dit schrijven op deze manier. En dan zetten we hier een vraagtekentje achter van die is er of één keer of niet. Dan ben ik dat lege ding dus kwijt. Wat ik dan hiermee kan doen is een hele mooie A B buiten haakjes halen en dan krijg ik binnen de haakjes hier een A B A B, dan krijg ik of een lege of een A B en dit haakje en dit vraagtekentje blijft gewoon staan. ((typt)) Dan kan ik op diezelfde manier dus weer die lege erbij halen en dan kan ik nog een keer (.) de (.) A B erbuiten gaan halen en als ik dan doorga als het goed is dan hou ik dit over. ((typt))

H: Ja je, wat denk je nu?

W: Ja dit was het volgens mij?

H: Dit was het? Je bent nu helemaal klaar met de opgave?

W: Ja ik denk dat dit het is. Ja ik ben klaar met het oplossen, het enige wat we nog moeten doen is kijken of inderdaad alles dekt.

H: Ja is prima, ga maar door, dat is een essentieel onderdeel

W: Ja nou, ja uiteraard. Maar dus de eerste, A B A B A B zit er duidelijk dus in (.) Ja, de A B van die tweede die zit er ook in. Als ik even, als ik de voorlaatste en het daarvoorstaande vraagtekentje weglaat zeg maar dan hou ik dus A B over. De lege die zit er ook in voor het laatste vraagtekentje [onverstaanbaar] en de A B A B zit er ook in. (.) Als ik dit tweede vraagtekentje: ja, ze zijn er allemaal.

H: Ja ok, dat is correct.