

Interview met Peter Hruschka

‘Software is een deel

Als je een hondenhok wilt bouwen heb je geen planning nodig. Als je een huis voor je gezin wilt bouwen kun je beter eerst een ontwerp maken. Datzelfde geldt dat voor software. Peter Hruschka gaat nog een stap verder en spreekt niet over software-ontwikkeling, maar over systeemontwikkeling. Hruschka verbindt *requirements* met *architecture*, op elk niveau van het ontwerpproces, dus ook voor alle subsystemen.

HANS VAN THIEL

stelsysteem'

Wanneer deze editie van PT Embedded Systems op de deurmat valt moet het boek *Process for System Architecture and Requirements Engineering* [1] verschenen zijn. Het is een nieuwe uitgave van Derek Hatley, Peter Hruschka en Imtiaz Pirbhai. De auteurs hebben hun methodologie, die niet specifiek is voor software maar gericht is op algemene systeemontwikkeling, PSARE genoemd. PT Embedded Systems sprak met auteur Peter Hruschka in zijn huis annex kantoor in het Duitse Aken over zijn werk in het algemeen en deze geïntegreerde benadering in het bijzonder.

PT Embedded Systems:

'U bent een van de eersten die aan *computer aided software engineering* (Case) heeft gewerkt kan ik mij herinneren?'

Peter Hruschka:

'Indertijd, achttien jaar geleden, was

ik bezig met software-engineering bij bedrijven als Daimler Benz en Deutsche Telecom en raakte ik overtuigd van de voordelen van gestructureerd ontwerpen. Daar kwam de eerste tool uit voort, Promod, en die bleek heel goed aan te sluiten bij de ideeën van Hatley en Pirbhai die enige jaren later, overigens onafhankelijk van mijn werk, hun bekende boek [2] publiceerden.'

PT Embedded Systems:

'In die tijd waren ontwikkeltools heel sterk afhankelijk van programmeertalen?'

Peter Hruschka:

'De eerste versies van Promod waren gericht op Pascal. Halverwege de jaren tachtig stapten we over naar C om de *portability* te verbeteren. We hadden toen ook veel klanten in de Verenigde Staten. Promod heeft grafische editors, analysers, report generators en de methode is op zich taalafhankelijk. Het is natuurlijk steeds verder ontwikkeld en maakt nu ook gebruik van UML (unified modeling language). Promod wordt op dit ogenblik, na een levensduur van vijftien jaar, overigens niet langer ondersteund.'

PT Embedded Systems:

'Verschilt de ontwikkeling van embedded software veel van die van algemene softwaresystemen?'

Peter Hruschka:

'Ingebedde software wordt meestal gestuurd door een externe gebeurtenis en voor deze *event driven* software, denk aan aerospace, toepassingen in auto's, is *state modeling* heel belang-

rijk. Hatley en Pirbhai introduceerden toestandsgestuurde modellertechnieken in het ontwerpproces.'

PT Embedded Systems:

'Hoe ziet U de toekomst van gestructureerd ontwerpen?'

Peter Hruschka:

'Het zal ooit worden vervangen, maar door wat? Objectoriëntatie (OO), zou ik zeggen. Op het ogenblik past echter minder dan 5% van alle ontwikkelaars

OO toe en slechts 30% gebruikt een ontwikkelmethode. De overgrote meerderheid van alle ontwerp, 70%, wordt dus zonder enige methode uitgevoerd. Het belangrijkste is de overtuiging op dit gebied te bestrijden.'

PT Embedded Systems:

'Is er een fundamenteel verschil tussen OO en gestructureerd ontwerpen?'

Peter Hruschka:

'Het ene ontwerp is gebaseerd op dingen en het andere op processen. Naar mijn mening blijven zaken als probleemdefinitie en *business processes* minder geschikt om als objecten te beschrijven. Voor het daadwerkelijk bouwen van systemen zal OO meer en meer toegepast worden.'

PT Embedded Systems:

'Wat vindt u in dit verband van Java?'

Peter Hruschka:

'Java is heel belangrijk. Het is een goede taal die de valkuilen van C++ en Smalltalk vermijdt. Het is echter een jonge taal die zeker nog niet is uitontwikkeld. De programmeertaal beïnvloedt overigens wel het ontwerp, maar dat is minder het geval voor grotere projecten dan voor kleinere. Hoe groter het project, hoe onbelangrijker de programmeertaal. In het algemeen schat ik dat 80% van het ontwikkelproces taalafhankelijk is. In het stadium van *detailed design* wordt dit 50%.

PT Embedded Systems:

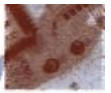
'Uw nieuwe boek *Process for System Architecture and Requirements Engi-*

'Op het ogenblik past echter minder dan 5% van alle ontwikkelaars OO toe en slechts 30% gebruikt een ontwikkelmethode.'

neering, heeft u geschreven samen met Derek Hatley en de helaas overleden Imtiaz Pirbhai. Het gaat niet speciaal over software-ontwikkeling, maar over systeemontwikkeling.'

Peter Hruschka:

'In dit boek verbinden we *requirements* met *architecture*. Dat is op zich niet helemaal nieuw, want dat werd al gedaan voor eisen en architectuur op >



hoger niveau. Het nieuwe is dat we dit nu doen op elk niveau van het ontwerpproces, dus ook voor alle subsystemen. We beschrijven een benadering in meerdere lagen waarin steeds requirements en architectuur naast elkaar optreden. De methode is zowel geschikt voor top-down- als bottom-up-ontwerp.

Het waterval-model werkt niet. Je kunt nu eenmaal niet in één slag de requirements en de architectuur opstellen. De praktijk is bovendien dat je vaak met bestaande systemen te maken hebt die slecht gedocumenteerd zijn en waarvan het overzicht ontbreekt.

Onze nieuwe benadering is dan dat je eerst uit de bestaande onderdelen de requirements destilleert. De tweede stap is dat je die echte requirements combineert en op een hoger niveau brengt. In de derde stap herstructureer je dan de nieuwe architectuur.

Migratie en reengineering gaan veel beter met dit model, dat ook beter geschikt is om de andere delen van het systeem te integreren.

Software op zichzelf doet nog helemaal niets. Daar is ook hardware voor nodig en misschien zijn er mechanische, hydraulische en elektrische deelsystemen. Er zijn meestal mensen betrokken bij het uiteindelijke functioneren van een systeem. We willen met het ontwerpproces al deze deelsystemen kunnen ondersteunen en alle processen modelleren.'

PT Embedded Systems:

'Maar het soort deelsysteem bepaalt toch de mogelijkheden?'

Peter Hruschka:

'Overgingen met betrekking tot het type deelsysteem moet je zo lang mogelijk uitstellen. Het gaat om de informatie die deelsystemen uitwisselen en de kanalen waarmee zij communiceren. Op een bepaald punt gaat het ontwerp naar de experts toe, bijvoorbeeld deskundigen in mechanisch ontwerp of software. Het requirements-model is echter onafhankelijk daarvan. Technologisch neutraal ontwerpen is echt wel mogelijk. Denk aan het ontwerp van een hotel. Inchecken kan door een computer gebeuren via een magneetkaart, of door een baliemedewerker. De requirements zijn voor beide gevallen dezelfde.

Vanuit de high level requirements



'Het motto van *Extreme Programming* luidt: *hack it now, fix it later*. Het werkt tot een zekere programmagrootte, daarna niet meer.'

van een systeem kom je bij de architectuur. Je hebt dan misschien vijf componenten die de eisen kunnen vervullen. Dan ga je keuzes maken over de typen onderdelen en nadenken over hun architecturen. De praktijk is dat bij grote ondernemingen misschien 150 applicaties draaien die elk 5 miljoen gulden kosten. Hun onderlinge samenhang is echter volstrekt onduidelijk. Als je in dergelijke situaties een overzicht van het totaal kunt krijgen zou dat bijzonder waardevol zijn.'

PT Embedded Systems:

'Er zijn verschillende, ook grote, bedrijven die zeer succesvol zijn met hun software maar die geen enkele ontwerpmethode hebben gebruikt. Weliswaar ontstaan er op een gegeven ogenblik problemen, vooral bij het onderhoud, maar toch...'

Peter Hruschka:

'Grady Booch heeft het wel eens zo gezegd: als je een hondenkooi wilt bouwen heb je geen planning nodig. Als je een huis voor je gezin wilt bouwen kun je beter eerst een ontwerp maken. Als je een heel gebouwencomplex wilt opzetten kun je beter nog weer wat langer nadenken. In de praktijk worden er heel wat hondenkooien gebouwd. Vroeger of later ontkom je dan niet aan een re-engineering van je systemen. Bent U bekend met *Extreme Programming*?'

PT Embedded Systems:

'Alleen als buzz-word.'

Peter Hruschka:

'Het motto van *Extreme Programming* luidt: *hack it now, fix it later*. Het is een collectie van een stuk of tien technieken om te coderen. Het werkt tot een zekere programmagrootte, daarna niet meer.

Voor multi-project, multi-system ontwerpprojecten kun je niet zonder methode. Denk ook aan *total quality management* (TQM), het vaststellen van kwaliteit door overeenstemming met requirements.'

PT Embedded Systems:

'Dat laatste geldt met name ook voor embedded systems?'

Peter Hruschka:

'Voor ingebedde systemen gelden vaak heel harde eisen met betrekking tot veiligheid, beveiliging, betrouwbaarheid en timing. Hier kom je inderdaad bijzondere beperkingen tegen. Als iets binnen tien milliseconden moet worden afgehandeld kun je die taak niet in vijf componenten gaan splitsen.'

PT Embedded Systems:

'U heeft over dit onderwerp, samen met anderen, ook een boek [3] geschreven: *Real Time Systems - Investigating Industrial Practice*.

Peter Hruschka:

'Ja, we hebben daarin onder meerdere een classificatie gemaakt van embedded systemen in verschillende categorieën, gebaseerd op verschillen in requirements. De ontwerpmethode hangt vervolgens af van de categorie van het te ontwikkelen systeem.'

PT Embedded Systems:

'Ziet U een rol weggelegd voor UML?'

Peter Hruschka:

'PSARE kan met UML worden uitgevoerd. Als een dataflow-diagram niet voldoet dan kan men eenvoudigweg een *activity diagram* gebruiken. In plaats van *state transition diagrams* kan men de UML *state chart diagrams* toepassen. UML is geen ontwerpmethode maar een standaardisatie van grafische hulpmiddelen. Wij geven in PSARE niet alleen diagrammen maar ook onderliggende specificaties in tekst. Een verschil met het vorige Hatley/Pirbhai is verder dat eindige automaten nu ook genest kunnen zijn.'

PT Embedded Systems:

'Wat vindt U van *capsules* en *signals* en dergelijke die in het zogenoemde Real-Time UML zijn toegevoegd?'

Peter Hruschka:

'UML is een standaard met een ingebouwd extensiemechanisme. Daar-

mee kun je bijna alles veranderen. Je kunt ook concepten zoals *signals* en *ports* die van andere methoden afkomstig zijn toevoegen aan UML en dan claimen dat ze *UML compliant* zijn. Dat is eigenlijk niet goed. Men zou een standaard moeten navolgen in plaats van de eigen begrippen langs een omweg in te voeren.'

PT Embedded Systems:

'U bent consulent, trainer en auteur en lid van twee verschillende samenwerkingsverbanden: *The Atlantic Systems Guild* en *System Bauhaus*. Wat moet ik me daarbij voorstellen?'

Peter Hruschka:

'Het zijn losse samenwerkingsverbanden tussen een aantal vrijgevestigde specialisten in systeemontwikkeling. *The Atlantic Systems Guild* is vooral gericht op de Verenigde Staten en *System Bauhaus* op Europa en dan vooral Duitsland. *System Bauhaus* bestaat uit een groep systeem- en softwareontwikkelaars die zich geïnspireerd voelen door de Bauhaus-beweging uit de architectuur. Een van de slagzinnen



'Grady Booch heeft wel eens gezegd: als je een hondenhok wilt bouwen heb je geen planning nodig. Als je een huis voor je gezin wilt bouwen kun je beter eerst een ontwerp maken.'

van Bauhaus luidde bijvoorbeeld: *Vorm volgt functie*. Dit motto hebben wij overgenomen omdat wij dat ook voor systeemontwikkeling toepasbaar vinden.'

PT Embedded Systems:

'Tot slot: heeft PSARE een specifieke *Software development life cycle* (SDLC) of hangt die van het ontwerp af?'

Peter Hruschka:

'PSARE is een development life cycle voor requirements en architectuur waarin opeenvolging (sequencing, red.) nauwelijks een rol speelt. Nogmaals: het is ook zeer geschikt voor toepassing op bestaande systemen.' □

Achtergrond

[1] *Process for System Architecture and Requirements Engineering*, D.Hatley, P.Hruschka, I.Pirbha, Dorset House, New York, 2000. ISBN 0-932633-41-2

[2] *Strategies for Real Time Systems Specification*, D. Hatley, I. Pirbhai, Dorset House, New York, 1987, ISBN 0 932 633 11 0.

[3] *Real Time Systems - Investigating Industrial Practice*, John Wiley, Chichester, I.Pyle, P. Hruschka, M. Lissandre, K. Jackson, 1993, ISBN 0 471 93553 0.

Peter Hruschka verzorgt dit najaar een cursus bij Bergson Software Tools in Eindhoven. Inlichtingen: info@bergson-tools.com en www.bergson-tools.com