

## Samenvatting

Dit proefschrift behandelt het systematisch ontwerpen van applicatieprotocollen. Het laagste niveau van abstractie waarop applicatieprotocollen worden beschouwd in dit proefschrift is dat van protocolarchitectuur. Een protocolarchitectuur is een voorschrift voor het implementeren van een protocol en omvat de condities die noodzakelijk maar voldoende zijn om protocolentiteiten onafhankelijk van elkaar te kunnen implementeren. Protocolarchitecturen worden gestandaardiseerd door internationale organisaties, zoals ISO en ITU-TSS, teneinde eindgebruikers onafhankelijk te maken van fabrikant- en leverancierspecifieke protocolproducten.

Een belangrijke referentie-architectuur voor het ontwikkelen van protocolstandaarden, inclusief applicatieprotocolstandaarden, is het 'Reference Model for Open Systems Interconnection' (OSI-RM). Het OSI-RM beschrijft een gelaagde applicatieprotocolarchitectuur, bestaande uit drie lagen: de Applicatielaag, de Presentatielaag en de Sessielag. De Presentatielaag en de Sessielag zijn 'vaste' lagen: alle gedistribueerde computertoepassingen moeten van deze lagen gebruik maken. De Applicatielaag daarentegen omvat 'Application service elements' (ASEs) die elk een bepaalde samenwerking voorschrijven en die geselecteerd worden afhankelijk van de behoefte aan een dergelijke samenwerking. Het is dus mogelijk dat gedurende bepaalde fases in een gedistribueerde samenwerking geen enkel ASE betrokken is.

De applicatieprotocolarchitectuur beschreven door het OSI-RM heeft een aantal belangrijke nadelen: (1) de 'vaste' lagen in de applicatieprotocolarchitectuur maken dat deze architectuur niet geschikt is voor het ondersteunen van alle mogelijke gedistribueerde computertoepassingen. Bovendien leidt het gebruik van 'vaste' applicatieprotocol-lagen tot omvangrijke, onderling afhankelijke en ingewikkelde standaarden, die daardoor moeilijk te gebruiken en te onderhouden zijn; (2) de concepten waarop de 'flexibele' Applicatielaag berust zijn niet duidelijk beschreven. De potentiële voordelen van deze laag, zoals een betere scheiding van ontwerpzorgen, kunnen hierdoor niet worden benut; en (3) de ontwikkeling van applicatieprotocolstandaarden kost in het algemeen erg veel tijd.

Deze problemen zijn voor een belangrijk deel te wijten aan het ontbreken van ontwerpmethoden in het OSI-RM. Concepten die een belangrijke rol kunnen spelen in ontwerpmethoden, zoals het concept 'service', zijn niet duidelijk gedefinieerd en worden (hierdoor) niet optimaal benut. Methoden voor service-decompositie komen daardoor als vanzelfsprekend niet aan de orde. Een ander probleem is de onduidelijke relatie met andere ontwerpdisciplines voor gedistribueerde systemen, met name het ontwerpen van

systemen voor gedistribueerde verwerking. Als gevolg hiervan is het moeilijker om de behoefte aan applicatieprotocollen te identificeren tijdens het ontwerpen van complexe gedistribueerde systemen waarbij meerdere ontwerpdisciplines betrokken zijn. De integratie van gedistribueerde verwerking en applicatieprotocollen wordt beschouwd in het 'Reference Model of Open Distributed Processing' (ODP-RM), maar de genoemde relatie is ook hier niet duidelijk beschreven.

Bij het ontwerpen van applicatieprotocollen is het nuttig om een onderscheid te maken tussen gedrag en functionele entiteiten waaraan gedrag wordt toegewezen. Het structureren van gedrag wordt hierdoor mogelijk zonder de samenstelling van een systeem uit deelsystemen te impliceren. Het onderscheiden van een gedragsdomein geeft aanleiding tot het identificeren van de elementaire concepten 'actie', 'interactie', en 'causale relatie'. Het onderscheiden van een entiteitendomein leidt tot de identificatie van de elementaire concepten 'actiepunt' en 'interactiepunt'.

Het onderscheiden van gerelateerde abstractieniveaus vormt een goed uitgangspunt voor het definiëren van ontwerpmethoden. In dit proefschrift worden abstractieniveaus gedefinieerd waarbij: (1) een systeem beschouwd wordt als een samenstelling van deelsystemen, (2) het systeem beschouwd wordt onafhankelijk van zijn mogelijke samenstellingen, en (3) de interacties tussen het systeem en zijn omgeving beschouwd worden onafhankelijk van de rol van het systeem (en de omgeving) in deze interacties. Deze abstractieniveaus kunnen voor de beschrijving van willekeurig systemen worden gebruikt, dus ook voor deelsystemen binnen systemen. Op deze manier is het mogelijk om een ontwerptraject voor applicatieprotocollen te definiëren, waarin relevante ontwerpresultaten in het ontwerpproces worden geordend op basis van de corresponderende abstractieniveaus. Deze ontwerpresultaten worden 'ontwerpmijlpalen' genoemd. Opeenvolgende mijlpalen bakenen ontwerpstappen af waarin specifieke ontwerpdoelen worden nagestreefd.

Mijlpalen en ontwerpstappen vereisen dat gedrag kan worden gerepresenteerd en gemanipuleerd. Gedrag kan worden gedefinieerd in termen van acties, interacties en hun causale relaties. Om de rol van een functionele entiteit in interacties met andere entiteiten te kunnen definiëren is een structureringstechniek nodig die we 'constraint-oriented' gedragscompositie hebben genoemd. Deze techniek kan ook gebruikt worden om gedrag te structureren, onafhankelijk van een toewijzing aan functionele entiteiten. Daarnaast is een structureringstechniek nodig die het mogelijk maakt deelgedragingen te relateren, zodanig dat (inter)acties van een deelgedrag afhankelijk gemaakt worden van (inter)acties van andere deelgedragingen. Deze techniek hebben we de 'causality-oriented' gedragscompositie genoemd. Het manipuleren van gedrag in ontwerpstappen is gericht op het decomponeren of het verfijnen van een gegeven gedrag. Gedragsmanipulaties die een rol spelen in het applicatieprotocol ontwerptraject zijn geïdentificeerd en correctheidseisen aan deze manipulaties zijn geformuleerd.

Met behulp van de geïdentificeerde abstractieniveaus, gedragsmanipulaties en structureringstechnieken is het mogelijk specifieke applicatieprotocol ontwerpmethoden te definiëren. Twee methoden worden gepresenteerd die naast elkaar gebruikt kunnen worden. Beide methoden hebben als uitgangspunt een vereiste applicatieservice. Een van de ontwerpmethoden leidt tot een gelaagde structuur van applicatieprotocollen. Een applicatieprotocol samengesteld uit protocollagen wordt beschreven met ‘constraint-oriented’ gedragscompositie.

Zoals eerder opgemerkt hebben ‘vaste’ lagen belangrijke nadelen. Het is beter de ontwerper van een applicatieprotocol het aantal en de keuze van lagen te laten bepalen, en geen ‘vaste’ lagen voor te schrijven. Omdat hergebruik van complete lagen beperkt is, is nog een andere, horizontale structurering nodig die hergebruik mogelijk maakt van delen van een laag. De tweede ontwerpmethode leidt tot een structurering van applicatieprotocollen in deelgedragingen. Een deelgedrag van een protocol hebben we ‘protocolsectie’ genoemd. Het concept ‘protocolsectie’ is vergelijkbaar met dat van ASE, behalve dat protocolsecties gedefinieerd kunnen worden in willekeurige lagen. Een applicatieprotocol samengesteld uit protocolsecties wordt beschreven met ‘causality-oriented’ gedragscompositie. De compositie van applicatieprotocollen met behulp van protocollagen en protocolsecties vormt de basis voor een flexibele referentie-architectuur voor applicatieprotocollen.

Het proefschrift is als volgt gestructureerd:

- *Hoofdstuk 1* presenteert een globale probleembeschrijving voor het onderzoek, en de afbakening, de doelstellingen en de methode van aanpak van het onderzoek. Verder worden een aantal concepten geïntroduceerd die van algemeen belang zijn bij het ontwerpen van gedistribueerde systemen.
- *Hoofdstuk 2* bespreekt kwaliteitscriteria die gebruikt kunnen worden om het ontwerpen van applicatieprotocollen te sturen en om reeds ontworpen applicatieprotocollen te evalueren.
- *Hoofdstuk 3* bespreekt de OSI applicatieprotocolarchitectuur en de hiermee gerelateerde concepten. Het bevat ook een korte beschrijving van de belangrijkste applicatieprotocolstandaarden die ontwikkeld zijn in het kader van het OSI-RM.
- *Hoofdstuk 4* evalueert de OSI applicatieprotocolarchitectuur en de hiermee gerelateerde concepten. Het bespreekt tevens de relatie tussen de kwaliteit van applicatieprotocolstandaarden en de aard van standaardisatie, en de implementatievrijheid die geboden wordt door protocolstandaarden.
- *Hoofdstuk 5* presenteert een raamwerk voor het ontwerpen van applicatieprotocollen. Het identificeert elementaire concepten voor gedrag en voor functionele entiteiten, en verschillende abstractieniveaus waarop willekeurige systemen kunnen worden gerepresenteerd. Op basis hiervan wordt een applicatieprotocol ontwerptraject beschreven

dat bestaat uit een aantal achtereenvolgende ontwerpstappen tussen mijlpalen in het applicatieprotocol ontwerpproces.

- *Hoofdstuk 6* bespreekt een model dat gebruikt kan worden in het applicatieprotocol ontwerptraject voor het representeren en het manipuleren van gedrag. Het model bevat, naast de eerder genoemde elementaire ontwerpconcepten, concepten en regels voor het samenstellen van gedrag. Twee technieken voor het structureren van gedrag worden beschreven en een aantal vormen van gedragsdecompositie en -verfijning worden geïdentificeerd.
- *Hoofdstuk 7* presenteert een flexibele referentie-architectuur voor applicatieprotocollen. Deze referentie-architectuur is gebaseerd op de compositie van protocollagen en van protocolsecties, waarbij gebruik wordt gemaakt van respectievelijk de 'constraint-oriented' en de 'causality-oriented' structureringstechniek. De architectuur omvat ook de karakterisering van een aantal applicatieprotocolsecties die als bouwstenen in veel verschillende toepassingen gebruikt kunnen worden.
- *Hoofdstuk 8* beschrijft enige suggesties voor verder werk.
- *Hoofdstuk 9* presenteert een samenvatting van de conclusies uit de vorige hoofdstukken.