

## Samenvatting

Vooruitgang in draadloze netwerktechnologie heeft de realisatie van een nieuw soort applicaties mogelijk gemaakt. Zogenaamde ambiente applicaties bestaan uit mobiele apparaten die spontaan met elkaar communiceren zonder de noodzaak van een centrale server. De netwerktechnologie die dergelijke spontane interacties mogelijk maakt noemt men *mobiele ad hoc netwerken*. De hardware kenmerken van deze nieuwe netwerken — vooral het feit dat de communicatie erg instabiel is en dat de apparaten niet kunnen beroepen op een centrale server — maken dat vorige software-oplossingen voor gedistribueerde systemen onbruikbaar zijn voor de ontwikkeling van ambiente applicaties. Software-technologie heeft zich tot nu toe gericht op het omgaan met de hardware kenmerken van ambiente applicaties, maar geven geen antwoorden op hoe grote en robuuste ambiente software kan gebouwd worden.

Software contracten hebben een belangrijke rol gespeeld bij de ontwikkeling van grote en robuuste software in het algemeen. Software contracten worden op brede schaal toegepast in statisch getypeerde talen en maken momenteel hun intrede in dynamisch getypeerde programmeertalen. Traditionele contract systemen kunnen echter niet worden toegepast op de constructie van ambiente applicaties omdat de wijze waarop deze gevalideerd worden er voor kan zorgen dat er fouten geïntroduceerd worden die niet aanwezig zijn in de oorspronkelijke ambiente applicatie. Daarnaast berusten veel van de huidige contract systemen voor gedistribueerde systemen op een *centrale* server, dewelke niet verenigbaar is met het ad-hoc karakter van ambiente toepassingen. Het werk in dit proefschrift wordt gemotiveerd door het ontbreken van een passend contract systeem voor ambiente toepassingen. Dit belemmert de exploratie naar complexere en meer grootschalige ambiente toepassingen.

In dit proefschrift stellen we een nieuw contract systeem voor dat speciaal ontworpen is voor de ontwikkeling van ambiente toepassingen. Met deze ambiente contracten kan de programmeur uitdrukken welke verplichtingen moeten nageleefd worden tussen de gedistribueerde modules van een ambiente applicatie. Het belangrijkste onderzoeksthema dat we hebben aangekaart is de ontwikkeling van een contract systeem dat rekening houdt met de specifieke hardware-eisen van ambiente toepassingen. We tonen verschillende toepassingen van ambiente contracten en leggen uit hoe de schuld moet toegekend worden in geval van een contractbreuk.

Ambiente contracten zijn geformuleerd als een uitbreiding van AmbientTalk. Naast een praktische implementatie is er ook een formele basis ontwikkeld in PLT-Redex. We hebben het ambient contract systeem gevalideerd door het toe te passen in het kader van een groot applicatie raamwerk genaamd UrbiFlock. Door onze contracten toe te passen in praktische applicaties hebben we geconstateerd dat vele informele specificaties makkelijk gecodeerd kunnen worden met behulp van het ontwikkelde ambient contract systeem.

## Abstract

Advances in wireless technology have enabled the realization of a new brand of distributed applications, called ambient applications. Such ambient applications consists of mobile devices that spontaneously interact with each other without the need for a centralized server. The network technology that enables such spontaneous interactions is called a *mobile ad hoc network*. The hardware characteristics of these new kinds of networks — mainly the fact that the communication is highly volatile and that devices cannot rely on a centralized server — have rendered previous software solutions for distributed systems obsolete. To date, software technology has focused on dealing with these hardware characteristics of ambient applications but has failed to give answers on how to build big and robust ambient software.

A proven software methodology called software contracts has played an important role in the development of big and robust software in general. Software contracts are widely adopted in statically typed languages and are currently finding their entrance in dynamically-typed programming languages. Traditional contract systems however, cannot be applied to the construction of ambient applications because the way in which they are validated can introduce bugs that are not present in the original ambient application. In addition, many of the current contract systems for distributed systems rely on a *centralized* server which is not reconcilable with the ad-hoc nature of the ambient applications. Our work is motivated by the lack of an appropriate contract system for ambient applications which hampers the exploration of more complex and more large scale ambient applications.

In this dissertation, we propose a new contract system, dubbed *ambient contracts* which is specifically designed for the development of ambient applications. With ambient contracts the programmer can express the obligations and promises between the distributed modules of an ambient system. The main research challenge that we address is the adaptation of current contract systems to take into account the specific hardware requirements of ambient applications. We show various applications of ambient contracts, and explain how to assign blame in case of a violation.

Ambient contracts are formulated as an extension of AmbientTalk, a dynamically-typed object-oriented language built upon the principles of the actor model. Next to a practical implementation ambient contracts also have a formal specification which is validated by making use of the PLT-Redex tool. We have validated the ambient contract system, called AmbientTalk/C, by employing it in the context of a large application framework called UrbiFlock. From this use case, we found that many of its previous informal specifications could be easily encoded with the developed ambient contracts.