

nodigt U graag uit op de openbare verdediging van het proefschrift van

Humberto Rodríguez Avila

ter behaling van de graad van Doctor in de Wetenschappen

Titel van het proefschrift:

**Orchestratie van Actorgebaseerde Talen voor Cyber-physical
Systemen**

Curriculum vitae

Humberto behaalde zijn masterdiploma in Toegepaste Wiskunde en Informatica in 2014 aan de Universiteit van Holguin, Cuba.

Sinds 2015 is Humberto doctoraatsstudent aan het Software Languages Lab (SOFT) van de Vrije Universiteit Brussel. Het onderzoek van Humberto heeft zich voornamelijk toegespitst op nieuwe programmeertaalconstructies voor actorgebaseerde systemen. Zijn werk resulteerde in vijf wetenschappelijke publicaties in peergereviewde tijdschriften, conferenties en workshops.

Promotor:

Prof. dr. Wolfgang De Meuter

Co-promotor:

Prof. dr. Joeri De Koster

De verdediging heeft plaats op

Dinsdag 24 augustus 2021 om 15u00

De verdediging kan via een livestream gevolgd worden. Contacteer Rodriguez.Avila.Humberto@vub.be voor meer informatie

Samenstelling van de jury

Prof. dr. Ann Nowé (VUB, voorzitter)

Prof. dr. Kris Steenhaut (VUB, secretaris)

Prof. dr. Theo D'Hondt (VUB)

Prof. dr. Shigeru Chiba (UTokyo, Japan)

Dr. Tim Felgentreff (Oracle Labs, Germany)

Abstract van het doctoraatsonderzoek

Actorgebaseerde programmeertalen bieden reeds verschillende essentiële constructies aan voor het ontwikkelen van cyber-physical systemen. Zulke systemen benutten het isolatieprincipe van actoren ten volle teneinde hun performantie- en schaalbaarheidseisen te realiseren. Jammer genoeg betekent de afhankelijkheid van 'isolatie als meest fundamentele bouwsteen' van het actormodel meteen ook dat programmeurs gedoemd zijn om complexe interactiepatronen tussen actoren uit te drukken m.b.v. ingewikkelde combinaties van asynchroon gestuurde boodschappen tussen de actoren. In de voorbije drie decennia werden dan ook verschillende taalontwerpen voorgesteld die als doel hebben de complexiteit te reduceren die voortvloeit uit het beschrijven van de coördinatie tussen de actoren met zulke interacties. Wij verdedigen de stelling dat geen van deze voorstellen voldoet om het ingewikkeld soort interactiepatronen uit te drukken die men typisch terugvindt in moderne cyber-physical systemen.

Dit proefschrift formuleert zeven 'smart home automation' softwareconstructiescenario's (waarin ieder 'smart home apparaatje' voorgesteld wordt door zijn eigen actor) die duidelijk de noodzaak aantonen voor meer geavanceerde synchronisatiepatronen van boodschappen uitgewisseld tussen actoren; patronen die dus afwezig zijn in moderne gedistribueerde actorgebaseerde programmeertalen. We hebben bewijs verzameld voor de relevantie van deze scenario's in de praktijk d.v.m. een bevraging in verschillende online 'home automation' gemeenschappen op het internet. Het resultaat van deze bevraging toont overduidelijk de nood aan van meer geavanceerde synchronisatiemechanismen in moderne actorsystemen.

Een gedegen analyse van deze zeven scenario's op programmeertaalniveau heeft niet minder dan vijf fundamentele categorieën van synchronisatiepatronen bloot gelegd. Deze omvatten 1) filteren van boodschappen gebaseerd op hun inhoud en op hun tijdseigenschappen. 2) selecteren van één of meerdere boodschappen gebaseerd op de volgorde waarin deze werden ontvangen. 3) correlaties van boodschappen m.b.v. logische operatoren. 4) accumulaties van boodschappen gebaseerd op tijdstippen en/of tellingen. 5) aggregaties van geaccumuleerde boodschappen.

In dit proefschrift presenteren we Sparrow, een domeinspecifieke programmeertaal (DSL) die gebouwd werd bovenop Elixir. Sparrow breidt het enkelvoudige bericht-matchingsysteem van een moderne actorgebaseerde taal (in casu dus Elixir) uit met ondersteuning voor meervoudige bericht-matching. Dit wordt verwezenlijkt door het abstraheren en samenstellen van elementaire bericht-patronen. Sparrow biedt taalondersteuning aan voor alle vijf de categorieën van synchronisatiepatronen die hierboven werden beschreven. Sparrow werd voorzien van een precieze specificatie m.b.v. een formele uitvoerbare calculus, genaamd NEST. We evalueren Sparrow met een kwantitatieve vergelijking tussen een gangbare implementatie van de zeven hogervermelde scenario's met een implementatie in Sparrow. Onze vergelijking toont aan dat Sparrow tot een substantiële reductie leidt van overtollige code die in de gangbare implementaties verstrengeld is met de feitelijke synchronisatiecode.