**SCIENCES &**
**BIOENGINEERING SCIENCES**

The Research Group

**Software Languages Lab**

has the honor to invite you to the public defense of the PhD thesis of

# Humberto Rodríguez Avila

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:

**Orchestration of Actor-based Languages for Cyber-physical Systems**

Promotor:
**Prof. dr. Wolfgang De Meuter**
Co-promotor:
**Prof. dr. Joeri De Koster**

The defense will take place on

**Tuesday, August 24, 2021 at 15h00**

The defense can be followed through a live stream. Contact Rodriguez.Avila.Humberto@vub.be for more information

## Members of the jury

Prof. dr. Ann Nowé (VUB, chair)
Prof. dr. Kris Steenhaut (VUB, secretary)
Prof. dr. Theo D'Hondt (VUB)
Prof. dr. Shigeru Chiba (UTokyo, Japan)
Dr. Tim Felgentreff (Oracle Labs, Germany)

## Curriculum vitae

Humberto obtained his master's degree in applied mathematics and informatics for the administration from the University of Holguín, Cuba, in 2014.

Since 2015, Humberto has been a PhD student at the Software Languages Lab (SOFT) at the Vrije Universiteit Brussel. Humberto's research has been mainly focused on programming language abstractions for actor-based systems. His work resulted in five scientific publications in international peer-reviewed journals, conferences, and workshops.

## Abstract of the PhD research

Actor-based programming languages already offer many essential features for developing modern cyber-physical systems. These systems exploit the actor model's isolation property to fulfill their performance and scalability demands. Unfortunately, the reliance of the model on isolation as its most fundamental property requires programmers to express complex interaction patterns between actors as complex combinations of asynchronous messages. In the last three decades, several language design proposals have been introduced to reduce the complexity that emerges from describing said interaction and actors' coordination. We argue that none of these proposals is satisfactory to express the many complex interaction patterns between actors found in modern cyber-physical systems.

This dissertation formulates seven smart home automation software construction scenarios (in which every smart home appliance is represented by its own actor) which motivate the need for advanced types of message synchronization patterns between actors in practice; patterns that are lacking in modern distributed actor-based languages. We have collected evidence for the practical relevance of these scenarios by means of an online poll conducted in various online home automation communities. The results of this poll clearly cement the need for advanced synchronization mechanisms in modern actor systems.

A careful analysis of these seven scenarios at the programming language level uncovers five fundamental categories of synchronization patterns. These include 1) the filtering of messages, both based on their content as well as on their timestamps. 2) The selection of one or more messages based on the order in which they arrive. 3) Correlation of messages using logical operators. 4) Accumulation of messages based on windows in time as well as the number of messages. And lastly, 5) Aggregation of accumulated messages.

In this thesis, we present Sparrow, a domain-specific-language (DSL) built on top of Elixir. Sparrow extends the single-message matching paradigm of contemporary actor-based languages to support multiple-message matching. This is enabled by supporting the abstraction and composition of elementary message patterns. Sparrow includes novel language abstractions to support all five categories of synchronization patterns. We also implemented an executable formal calculus for Sparrow -- called NEST -- that serves as a precise specification of its defining language features. We evaluate our DSL using a quantitative comparison of a state-of-the-art implementation of all seven scenarios with an implementation in Sparrow. Our evaluation shows that Sparrow effectively reduces the amount of extraneous code that is interleaved with the synchronization code.