

AN AUTOMATIC DRIVING CHALLENGE USING MODEL-BASED DESIGN AND SYNCHRONOUS PROGRAMMING



CREDITS



Eugène Asarin





Mihaela Sighireanu



Margarette Lim









Delphine Merkel, Chafic Jaber



Thierry Lorioux @ EIDD Benjamin Blanc @ CEA Selma Saïdi @ TUH





FMF 2017, Toulouse

CONTEXT AND GOALS

SYNC CONTEST champions to Master students

- model-based design
- synchronous languages

for coding transport systems with various objectives

- UPD-EIDD: apply project-based skills-oriented pedagogy to teach embedded systems, model-based design, formal modeling and spec.
- ANSYS: promote SCADE Suite among students
- EXPECTRA: discover brilliant students

... an original approach compared to other robot-based competitions

• e.g., PersyCup @ Grenoble, Game of Robots @ Angers



2017 EDITION: AUTOMATIC DRIVER

"Design a completely autonomous driver in a simple urban environment including **roads, crossings, traffic lights, speed limits, obstacles**, **one vehicle** (1 frontal + 1 abdominal color sensors, 1 sonar, min 2 wheels), and one itinerary to follow."





Virtual city environment for monitoring driver's performance and scoring

Toy city environment for Lego Mindstorms NXT



AGENDA

- February 2016: organizers start the project
- March-June 2016: organizers fix the specification
- July-August 2016: academia code virtual city & driver solution
- September 2016: organizers open registration and send specs
- Oct.-Dec. 2016: students learn SCADE and code virtual driver
- January 2017: teams deploy driver on Lego Mindstorms
 - based on SCADE manuals and academia tips
- February 2nd, 2017: Final
 - Morning: 3 new programming problems
 - Follow a slow vehicle (*)
 - Turn temporary on a parallel road (**)
 - Find the road (***)
 - Afternoon: race on robotic platform, music&food, awards





CHALLENGES

For students:

- Iearn SCADE and understand the given code
- develop automatic driver (automata, PID)
- calibrate to get max score for all maps and itineraries
- deploy the SCADE code on Lego Mindstorms
- add code for the new tasks

For organizers:

- find a simple and still interesting problem
- define an optimal city environment
 - road configuration, maps and itineraries
 - simulator with monitoring, scoring, sound effects
- provide useful libraries and modular code
- make easier deployment Lego Mindstorms





3 months
3 months
} 1 month
3 hours

IMPACT

For students:

- Iearn more on SCADE, C, control theory
- prizes and publicity
- jobs

For organizers:

- Iearn more on SCADE Suite and Lego platforms
 - SCADE system, Rapid Prototyper, code generation & deployment
- publicity
- new teaching experience due to new subject
 - \checkmark Good CS & Mechatronics students need 2 months to be comfortable in SCADE.
 - ✓ CS students are uneasy with control theory.
 - \checkmark Modeling style improves by specs and examples.







... AND THE FORMAL METHODS?

Formal methods in the foreground

- principle of model-based design
 - with hierarchical state machines, synchronous and data flow semantics
- run-time monitoring for simulation and scoring, no verification

and in the background

- Students have theoretical background in
 - formal modeling (LTS, PN, automata) and specification (LTL, CTL, Hoare's logic),
 - verification methods (deductive verification, model-checking, static analysis),
 - contract-based programming (Haskell, Ada)
- Teachers are researchers in foundations of algorithmic verification
 - model-checking of (counter, timed, hybrid)-extended automata,
 - model-checking and static analysis of imperative, concurrent, distributed code,
 - logics and automata theory

for a **simple case study of automatic driver** with typical challenges for formal verification



OVERVIEW: SYSTEM SPEC





OVERVIEW: SCADE NODE





OVERVIEW: ROBOT SPEC





OVERVIEW: DRIVER (PART OF)





OVERVIEW: CITY OBSERVERS

ModelBasedDesignOfRobotChallenge





OVERVIEW: OBSERVER LIGHTRUN





CASE STUDY FOR FM

Qualitative analysis challenges

- deal with/abstract floating point (float64) data and functions
 - sin, cos for turning
 - distance for colors comparison
 - error integrator in PID
- for a given class of maps
 - e.g., straight roads, small crossings, more colors

Quantitative analysis challenges

- measure quality of driving
 - 6 criteria monitored, get costs instead bool
 - compare with a reference driver
- measure memory space required
 - unexpected crashes at deployment





FOLLOW US

- Web: <u>www.synccontest.com</u>
- YouTube: channel "Sync Contest2017"
- Flickr: group "5535 Paris-Diderot" album "SynC Contest 2017"
- Facebook

... and JOIN US for the next edition in 2019!

Thank you!

