

The CPSwarm Technology for Designing Swarms of Cyber-Physical Systems

Midhat Jdeed¹, Melanie Schranz², Alessandra Bagnato³, Sarah Suleri⁴,
Gianluca Prato⁵, Davide Conzon⁵, Micha Sende², Etienne Brosse³,
Claudio Pastrone⁵, and Wilfried Elmenreich¹

¹ Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
Email: *firstname.lastname@aaau.at*

² Lakeside Labs GmbH, Klagenfurt, Austria
Email: *lastname@lakeside-labs.com*

³ Softeam RD Department, Paris, France
Email: *firstname.lastname@softeam.fr*

⁴ Fraunhofer Institute for Applied Information Technologies, Sankt Augustin,
Germany

Email: *firstname.lastname@fit.fraunhofer.de*

⁵ LINKS Foundation, Torino, Italy
Email: *firstname.lastname@linksfoundation.com*

Abstract. The increasing interactions among Cyber-Physical Systems (CPSs) lead to systems with emerging and unpredictable behaviors. Such an example is the domain of swarm robotics where the interactions among swarm members remain a complex topic especially in environments that are dynamically changing. However, CPSs find applications in a number of large-scale, safety-critical domains, e.g., transportation, smart cities. Considering this fact, CPSwarm project positions itself in the domain of swarm of CPS design and engineering, and to provide tools and methodologies that pave the way toward well-established, model-based and predictive engineering design methodologies and toolchains for a swarm of CPS. This paper showcases the CPSwarm results during the first two years, 2017 and 2018, and introduces the technological concept behind CPSwarm. The CPSwarm approach aims to establish a science of system integration in the domain of swarms of CPS, i.e., of complex herds of heterogeneous CPSs that interact and collaborate based on local policies in order to solve complex industrial-driven and real-world problems. Moreover, the paper presents a search and rescue (SAR) case study using the CPSwarm approach.

Keywords: Swarm robotics · Swarm engineering · Software technologies.

1 Project data

- **Acronym:** CPSwarm
- **Title:** Cyber-Physical Systems (CPS) Swarms
- **Start date:** January 1, 2017
- **Duration:** 36 months
- **Partners:** LINKS Foundation, Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V., Lakeside Labs, Alpen-Adria-Universität Klagenfurt, TTTech Computertechnik, SEARCH Laboratory, DigiSky, SOFT-EAM, and Robotnik Automation.
- **Web site:** <https://www.cpswarm.eu/>

2 Introduction

While the increased CPS adoption has resulted in the maturation of solutions for CPS development, a single consistent science of system integration for CPS has not yet been consolidated. Therefore, CPS development remains a complex and error-prone task, often requiring a collection of separate tools, to follow a CPS design cycle including modelling, simulation, optimization, deployment. Moreover, interactions amongst CPSs might lead to new behaviors and emerging properties, often with unpredictable results. Rather than being an unwanted byproduct, these interactions can become an advantage if explicitly managed since early design stages. CPSwarm tackles this challenge by proposing a new science of system integration and tools to support engineering of CPS swarms. CPSwarm tools will ease development and integration of complex herds of heterogeneous CPSs that collaborate based on local policies and that exhibit a collective behavior capable of solving complex, industrial-driven, real-world problems. The project defines a complete toolchain that enables the designer to (a) set-up collaborative autonomous CPSs; (b) test the swarm performance with respect to the design goal; and (c) massively deploy solutions towards reconfigurable CPS devices. Model-centric design and predictive engineering are the pillars of the project, enabling definition, composition, verification and simulation of collaborative, autonomous CPSs while accounting for various dynamics, constraints and for safety, performance and cost efficiency issues. Project results will be tested in real-world use cases in 3 different domains: swarms of Unmanned Aerial Vehicles (UAVs) and rovers for SAR applications; platooning of autonomous freight vehicles; and swarm logistics [1].

The paper is organized as follows: The next section provides a review of related projects. Section 4 presents the software technology for swarm of CPSs the CPSwarm is developing. Section 5 describes the case study and the implementation process using the initial CPSwarm technology. Conclusions are drawn in Section 6, together with an outline of possible future work.

3 Related Work

The CPSwarm project focuses on system integration to provide a comprehensive but self-contained tool chain to model, optimize, simulate and deploy a CPS swarm. The project builds up on several other approaches that followed the idea of representing the CPS design process. To name but a few, the idea of the ENOSYS project [2] was to specify and develop a tool supported design flow for designing and implementing embedded systems by seamless integration of high-level system specifications, software code generation, hardware synthesis and design space exploration. The aim of the INTO-CPS [3] project was to create an integrated tool chain for comprehensive model-based design of CPSs. The project CERBERO [4] has a cross-layer model-based approach to describe, optimize, and analyze the system and all its different view, while supporting adaptivity. A similar approach is followed by OpenMETA [5], a model- and component-based design tool chain for CPSs that deals with a new horizontal integration layer to support model, tool and design process integration. Also,

FitOptiVis [6] project proposes the introduction of model-driven engineering technologies to design and develop dependable autonomous systems, driven by industrial use cases. CyFuzz [7], on the other, is a differential testing framework to find bugs in CPS development environments. The main focus of each project was on a single CPS and a selected set of tools, instead of aiming to represent the entire design process for CPS swarms.

4 CPSwarm Technology and Relevance of Project

To our knowledge, swarm algorithms have not yet been applied to real-world applications so far, thus, the CPSwarm aim is to address the software technologies design and their integration for such systems in order to solve complex industrial-driven and problems. The CPSwarm approach is to integrate various tools into the so called CPSwarm Workbench [8]. The Workbench is a toolset comprised of *Modelling Library*, *Modelling Tool*, *Simulation and Optimization Environment*, *Code Generator*, and *Deployment Tool*. Consequently, the CPSwarm *Launcher* is developed to be the glue for the complete toolset in the Workbench, Figure 1. The Launcher provides a user-friendly Graphical User Interface (GUI), to allow an easy and structured usage of the toolset. In addition, the Launcher acts as a wizard to guide the users through the CPSwarm workflow. It provides support for managing asset files for each respective phase within a project. In the following, a brief introduction about each tool is provided:

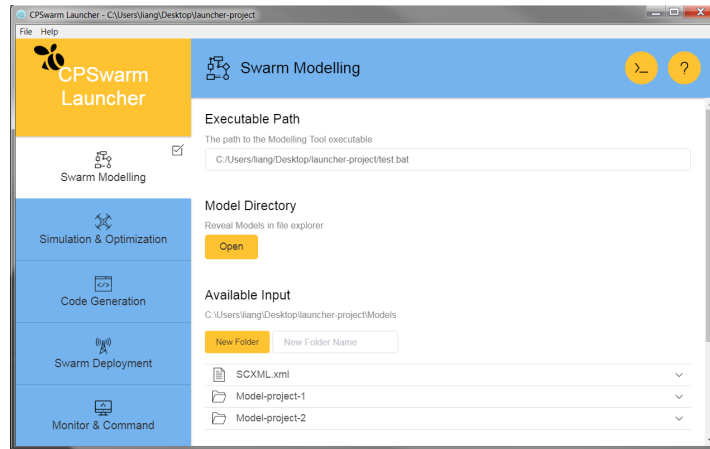


Fig. 1. The CPSwarm Launcher.

1. **Swarm Modelling** allows the definition of two main aspects of a swarm [9]: Its architecture as a set of CPSs composed of hardware components, and its behaviour as a set of individual state machine. Based on top of the SysML standard, many reusable models (CPS, sensor, actuator, controller and behaviour) are provided inside dedicated libraries to the user.
2. **Simulation and Optimization** allows evaluating the performance of a swarm solution. This environment includes mainly three components [10]:

the Simulation and Optimization Orchestrator (SOO) that coordinates all the simulation and optimization tasks; a set of Simulation Managers (SMs) that allow controlling heterogeneous Simulation Tools (STs) with common Application Programming Interfaces (API); and an Optimization Tool (OT), like FFramework for EVolutionary design (FREVO) [11], used to perform the optimization processes.

3. **Code Generation** represents the linking level between the modeling phase and the deployment of the code on an actual CPS. Using the Modelling Tool the user can easily model new CPS behaviours using Finite State Machines (FSM). The Code Generator accepts as input a standard description of the state machine using the State Chart XML (SCXML) format and return as output a python implementation of that state machine based on the SMACH library. The code generation process is realized using a template-based approach: a template is a text file and is usually composed by a static part that appears in the output as it is and a dynamic part written with a template language. This language can be processed by a template engine and replaced with the proper data given at runtime.
4. **Swarm Deployment** is a lightweight software update and monitoring system for resource-constrained Internet of Things (IoT) devices. It aims to provide secure, practical and easy to use utilities for over-the-air (OTA) provisioning of software on small computers (e.g., Raspberry Pi or other device with ARM or x86 architecture). Deployment manager is a centralized web service exposing APIs and a GUI for various deployment-related operations, ranging from compilation to installation and runtime. Deployment agents run on individual devices, performing deployment-related tasks.
5. **Monitor and Command Tool** that addresses the challenges to the phase after deployment, i.e., as the swarm of CPSs executes its mission. This tool runs in the runtime environment and acts as a central commander able to give specific commands to the CPSs swarm and monitor its behavior during mission execution. It can discover the events provided by the swarm members via the communication library. It sends configuration commands for modification of the swarm behavior.

5 Case Study: Search and Rescue

In one of the CPSwarm case studies, we consider heterogeneous swarms of unmanned ground vehicles (UGVs) and UAVs that in a collaborative and completely autonomous way can conduct certain missions in the surveillance of critical infrastructure like industrial or power plants as well as in SAR tasks such as finding human casualties or people trapped at the site of a major catastrophe. To validate the model-based design offered by the CPSwarm Workbench a simplified and miniaturized version of a SAR mission has been modeled, simulated and deployed in an area of 30m x 25m. The mission has a group of drones cover a selected area using a swarm strategy looking for possible casualties while the ground robots are on standby. When a casualty (represented by markers) is discovered, the drone communicates the position of the casualty to the rovers and

each rover computes its distance from the casualty and sends this info back to the drone. Then, the drone selects the closest rover to rescue the casualty and starts hovering above the casualty communicating possible changes of position. Finally, the selected rover reaches the casualty and virtually brings it back home.

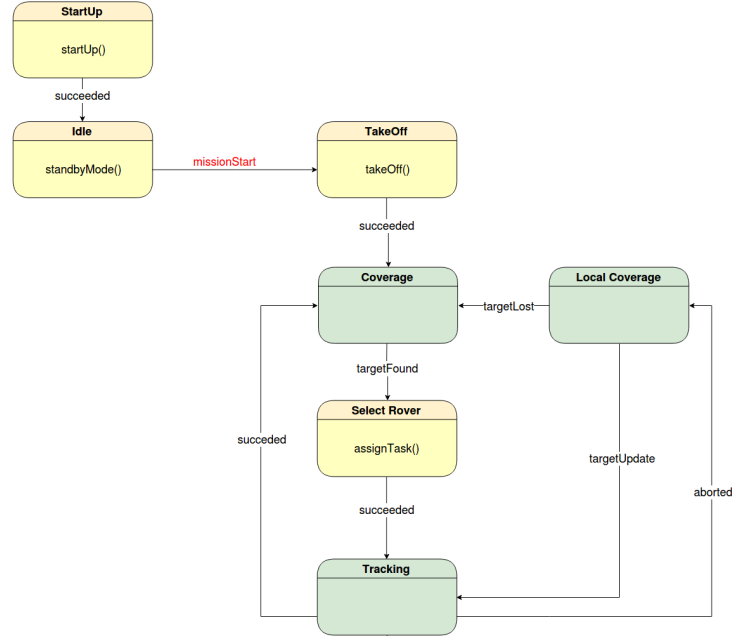


Fig. 2. SAR state machine for UAVs.

The implementation of this use case had been realized using all parts of the CPSwarm Workbench. For instance, as a first step, behaviours of both UGVs and UAVs has been designed in the Modelling Tool adopting an approach based on Finite State Machines (FSM). Each state has been linked or with a swarm algorithm (green boxes) or with an individual function provided by the CPSwarm Abstraction Library (yellow boxes), an example is shown in Figure 2. Then, the state machines have been exported in the SCXML format and provided as input for the Code Generator to produce a python implementation of the modelled behaviours to be deployed on the actual CPSs. More details can be found in this video⁶ on the CPSwarm YouTube channel.

6 Conclusion

This paper presented the CPSwarm approach for designing swarms of CPSs. The proposed CPSwarm Workbench represents the entire life cycle of a CPSs swarm, from modelling, simulation, optimization, to deployment and finally monitoring. In the CPSwarm SAR case study, we were able to map the CPSwarm Workbench vision onto a real-world application using heterogeneous CPSs swarms. As the

⁶ <https://www.youtube.com/watch?v=F4cFbDkZrWA>

CPSwarm project substantially addresses key challenges facing European CPS industry, most of the developed technologies, with the final achievements, will be available as open source products using the well-established exploitation and dissemination channels of the project partners to encourage their broad take-up by industry. Moreover, the CPSwarm project is considering educational activities in its dissemination strategy by planning for workshops at universities or summer schools to introduce the CPSwarm Workbench. Also, a low-cost robot, Spiderino [12], is developed for education and swarm research and it is used in workshops at schools and the Alpen-Adria-Universität in Klagenfurt, Austria.

7 Acknowledgments

This work has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 731946, CPSwarm project.

References

1. A. Bagnato, R. K. Bíró, D. Bonino, C. Pastrone, W. Elmenreich, R. Reiners, M. Schranz, and E. Arnautovic, “Designing swarms of cyber-physical systems: The h2020 cpswarm project,” in *Proceedings of the Computing Frontiers Conference*. ACM, 2017, pp. 305–312.
2. E. Brosse, I. R. Quadri, A. Sadovykh, F. Ieromnimon, D. Kritharidis, R. Catrou, and M. Sarlotte, “Enosys fp7 eu project: An integrated modeling and synthesis flow for embedded systems design,” in *7th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip*. IEEE, 2012, pp. 1–5.
3. P. G. Larsen, J. Fitzgerald, J. Woodcock, P. Fritzon, J. Brauer, C. Kleijn, T. Lecomte, M. Pfeil, O. Green, S. Basagiannis *et al.*, “Integrated tool chain for model-based design of cyber-physical systems: the into-cps project,” in *2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*. IEEE, 2016, pp. 1–6.
4. M. Masin, F. Palumbo, H. Myrhaug, J. de Oliveira Filho, M. Pastena, M. Pelcat, L. Raffo, F. Regazzoni, A. Sanchez, A. Toffetti *et al.*, “Cross-layer design of reconfigurable cyber-physical systems,” in *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2017, pp. 740–745.
5. J. Sztipanovits, T. Bapty, S. Neema, L. Howard, and E. Jackson, “Openmeta: A model- and component-based design tool chain for cyber-physical systems,” in *From Programs to Systems. The Systems perspective in Computing*, S. Bensalem, Y. Lakhneck, and A. Legay, Eds. Springer, 2014, pp. 235–248.
6. Z. Al-Ars, T. Basten, A. de Beer, M. Geilen, D. Goswami, P. Jääskeläinen, J. Kadlec, M. M. de Alejandro, F. Palumbo, G. Peeren *et al.*, “The fitoptivis ecseel project: highly efficient distributed embedded image/video processing in cyber-physical systems,” in *Proceedings of the 16th ACM International Conference on Computing Frontiers*. ACM, 2019, pp. 333–338.
7. S. A. Chowdhury, T. T. Johnson, and C. Csallner, “Cyfuzz: A differential testing framework for cyber-physical systems development environments,” in *International Workshop on Design, Modeling, and Evaluation of Cyber Physical Systems*. Springer, 2016, pp. 46–60.
8. L. Junhong, F. Tavakolizadeh, and S. Adugna Chala, “Initial cpswarm workbench and associated tools,” CPSwarm, Public Deliverable D3.4, 2017.
9. M. Schranz, G. Prato, and E. Brosse, “Updated cpswarm modelling tool,” CP-Swarm, Public Deliverable D5.3, 2018.
10. D. Conzon, A. Pitman, M. Cantero, A. Soriano, and O. Morando, “Initial integration of external simulators,” CPSwarm, Public Deliverable D6.5, 2018.
11. A. Sobe, I. Fehérvári, and W. Elmenreich, “Frevo: A tool for evolving and evaluating self-organizing systems,” in *2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems Workshops*. IEEE, 2012, pp. 105–110.
12. M. Jdeed, S. Zhevzhyk, F. Steinkellner, and W. Elmenreich, “Spiderino-a low-cost robot for swarm research and educational purposes,” in *2017 13th Workshop on intelligent solutions in embedded systems (WISES)*. IEEE, 2017, pp. 35–39.