Joel Greenyer, Malte Lochau, and Thomas Vogel (Hrsg.): Explainable Software for Cyber-Physical Systems (ES4CPS), GI-Dagstuhl Seminar 19023, January 2019,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 11

Strategy Patterns for Multi-Agent CPS

Claudio Menghi[1]

1 Introduction

CPS are software systems that are tightly integrated with their environments. These systems usually rely on a set of agents that collaborate or compete for achieving certain goals. Defining how these agents should behave, i.e., their missions, is complex and error-prone [Me18b] and plays a critical role in the definition of explainable CPS. Explainability refers to the capability of documenting and defining the mission the agents must achieve in a human-comprehensible and machine-processable format; two requirements that are apparently in conflict. On the one hand, the mission specification should be *machine-processable*, i.e., it must be precise and rigorous. This requirement intrinsically calls for the use of mathematical instruments, such as logical languages, that precisely define the semantics of missions. However, practitioners are often unfamiliar with the intricate syntax and semantics of logical languages which make these languages inappropriate for practical adoption [DAC99]. On the other hand, the mission specification should be *human-comprehensible*, i.e., it should be easily managed and defined by people with low technical expertise. However, comprehensible mission specifications, such as the one defined by using natural language, are often not machine-processable. Even if specification patterns tailored for the CSP domain have been recently proposed as a solution for this problem [Me18b], these patterns are defined for single-agent CPS applications, and thus they do not explicitly address the multi-agent case.

In multi-agents CPS applications, *strategies* play a pivotal role. Strategies are plans of actions designed to achieve the desired goal and specify when and how the different agents should perform their actions. While machine-processable strategy aware logics have been proposed in the literature, such as Strategy Logic [CHP07], they are not used in practice since they are not easily comprehensible by non-experts. Indeed, the use of logically based specifications is hard and error-prone [DAC99].

This talk discusses the problem of defining mission specifications for multi-agents CPS applications that are both machine-processable and human-comprehensible. It discusses *strategy patterns*, a proposed solution which extends the specification patterns for the CSP domain, recently proposed [Me18b], to consider multi-agents CPS.

[1] University of Luxembourg, Luxembourg, claudio.menghi@uni.lu

2 Strategy Patterns

Specification patterns are a well-known software engineering solution for defining requirements that are both human-comprehensible and machine-processable [DAC99]. Patterns map recurrent human-comprehensible specification problems into machine-processable solutions. They include a usage intent, known uses, and relationships to other patterns, which facilitate humans in specifications, and a template specification in a target logic. Extensions of the original patterns of Dwyer et al. [DAC99] have been proposed in the literature. For example, an extension of specification patterns tailored for the CSP domain has been recently proposed [Me18b]. However, none of the proposed extensions explicitly consider strategies, which, as discussed in the previous section, are primary concerns in multi-agents CPS.

Strategy patterns extend the property specification patterns for robotic missions [Me18b] by considering multi-agents CPS applications. This talk envisions two main classes of strategy patterns that are tailored for cases in which agents are *collaborating* and *competing* for achieving their goals. Within each of these classes, patterns capture recurrent mission specification problems. For example, the *divide and conquer collaborative patrolling* pattern extends the patrolling pattern recently proposed by Menghi et. al. [Me18b]. Given a set of locations to be patrolled by a set of agents, it assigns disjoint subsets of locations to the agents that must independently patrol them. Viceversa, in the *unite and conquer collaborative patrolling* pattern, all the robots must patrol all the locations within a given set. LTL is proposed as a target logic for the template mission specifications. Indeed, LTL is strongly used in the CSP domain [Lu18], it has been used in the specification patterns for robotic missions [Me18b], and it is also supported by several planners (e.g., [Me18a]).

References

[CHP07] Chatterjee, Krishnendu; Henzinger, Thomas A; Piterman, Nir: Strategy logic. In: International Conference on Concurrency Theory. Springer, pp. 59–73, 2007.

[DAC99] Dwyer, Matthew B; Avrunin, George S; Corbett, James C: Patterns in property specifications for finite-state verification. In: International Conference on Software Engineering. ACM, pp. 411–420, 1999.

[Lu18] Luckcuck, Matt; Farrell, Marie; Dennis, Louise; Dixon, Clare; Fisher, Michael: Formal Specification and Verification of Autonomous Robotic Systems: A Survey. arXiv preprint arXiv:1807.00048, 2018.

[Me18a] Menghi, Claudio; Garcia, Sergio; Pelliccione, Patrizio; Tumova, Jana: Multi-robot LTL Planning Under Uncertainty. In: Formal Methods. Springer, pp. 399–417, 2018.

[Me18b] Menghi, Claudio; Tsigkanos, Christos; Berger, Thorsten; Pelliccione, Patrizio; Ghezzi, Carlo: Property Specification Patterns for Robotic Missions. In: International Conference on Software Engineering: Companion Proceeedings. ICSE. ACM, pp. 434–435, 2018.