

Cobot Protocol Customisation Manager – PLUME

by Antonello Calabrò and Eda Marchetti (ISTI-CNR)

Making cobot safety protocols closer to a cookbook than to methodologies: Let users implement protocols without overhead in knowledge and understanding of the procedures.

The current intensive research activities around data-driven AI technologies, computer science and cognitive science still need to completely solve the stringent need for compliance and conformity with safety and security regulations. They provide essential documentation, requirements, and directives that must be satisfied, validated, and verified during the cobot's development. Indeed, to achieve a reliable human-machine symbiotic collaboration, regulations must be considered when applying data-driven AI technologies into cognitive robotics environments. However, the rules' complexity could impact the application of cognitive AI & cobots unless the required directives are not expressed as structured, easy-to-apply operational guidelines.

In this context, protocols are documents containing the methods, the collection of the procedures, and the prerequisites necessary to carry out safety experiments. These aim to verify and validate a cobot's quality (like safety and security) properties and the risk assessment [1].

Conceived inside the COVR project [L1], PLUME (Protocol Customization Manager) wants to provide an easy-to-use desktop application for writing/updating domain-specific protocols. PLUME is based on a model-driven approach that supports the protocol during its lifecycle. It guides users in creating, managing, and updating protocols to leverage the quality of the services they can offer to society.

As shown in Figure 1, through the PLUME user-friendly interface, a classification of the specific application domains and

the relative device types is provided. Consequently, PLUME allows to:

- Gather the required directives representing the legislative acts and goals that all EU countries must achieve. In the example, the Machinery Directive can be selected.
- Gather the domain-specific standards and requirements directly connected to the selected directive. In the example, the ISO/TS 15066 standard is automatically associated, and requirements like “Limit energy transfer during collision” are shown to the users.
- Select or specify the abilities (skills) of the cobot. These will be verified and validated for risk reduction. The “Limit physical interaction energy” safety skill is selected in the example.
- Specify the working environment conditions in which the cobot device will operate. In the example, the “indoor - factory” can be identified.
- Select or specify the metrics, measures, and boundaries used during the validation. In particular, the metrics can be associated with values or Boolean when a threshold-based assessment is necessary. In the example, the “holding” with the “power on” is the condition in which the device will be tested.
- Identify possible test procedures or cases useful for the cobot's validation and assessment.

As shown in Figure 1, PLUME creates an interactive protocol skeleton reporting the collected information. It provides the users with a manageable document for cobot testing and validation. It allows them to:

- finalise the content with additional information,
- modify or refine the different fields according to specific needs or constraints,
- include existing protocols (or part of them) into a new one.

From a process point of view, the PLUME life-cycle includes two kinds of users: the generic Users (for instance, developers or testers) who are responsible for the protocol creation and the Experts (like a member of insurance, legislation government/organisations, or professionals) who are responsible for the approval of the protocols.

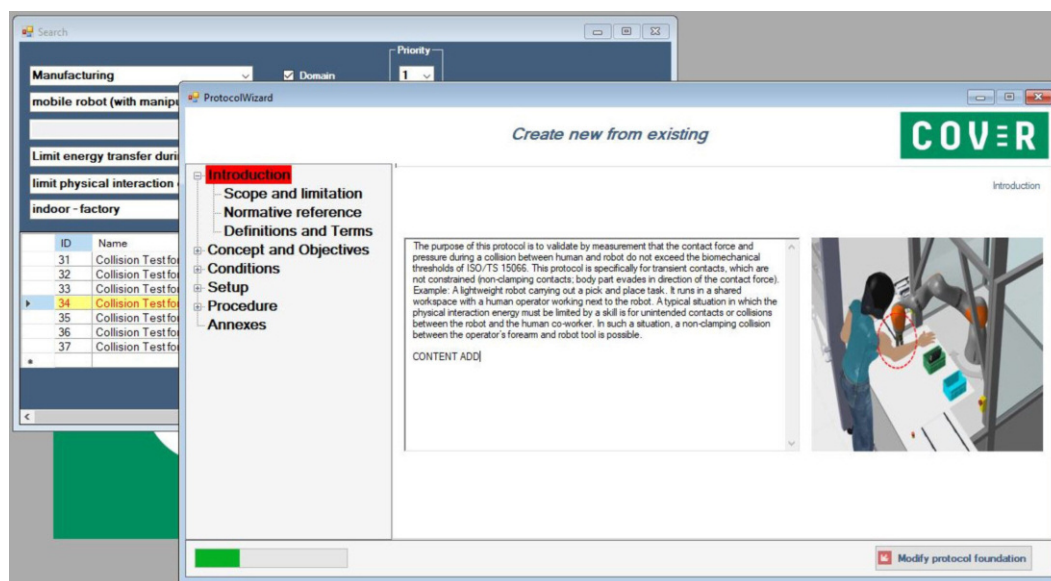


Figure 1: PLUME interactive protocol skeleton interface.

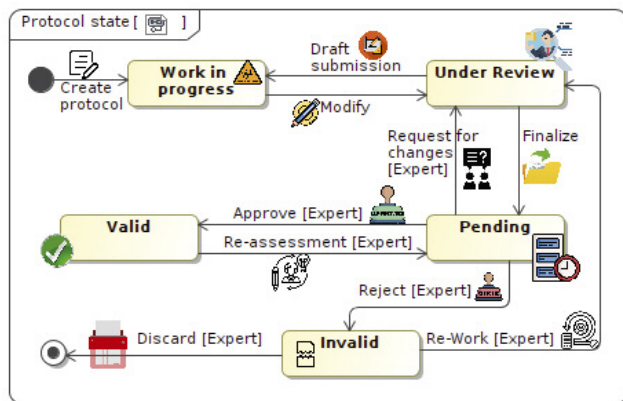


Figure 2: PLUME protocol life cycle.

As reported in Figure 2, during its lifecycle, the protocol passes through different stages:

- Work in progress is when the protocol skeleton fields need to be completed by the Users using the protocol wizard provided by PLUME.
- Under review is when the User provides a Draft submission or the protocol. In this stage, if the Users modify the protocol (Modify arrow in Figure 2), it returns to the Work in progress stage. Otherwise, it is classified as finalised and passed to the next Pending step (Finalize arrow in Figure 2).
- Pending is when the Experts evaluate the protocol to assess its correctness and completeness. In this case, three different situations can happen: Request for changes, i.e., the Experts describe needed modifications, and protocol returns to the Under review stage. Reject, i.e., the Experts do not approve the protocol, and it is moved to an “Invalid” stage. Approve, i.e., the Experts consider the protocol valid and move it into the Valid stage.
- Invalid is when a protocol needs to be discarded or integrated with Experts’ suitable instruction to leverage it again to the Under review stage.
- Valid is when a protocol is available and ready to be used. If Experts decide to modify the protocol, it returns to the Pending stage for the second round of evaluation.

The current implementation of PLUME’s standalone application relies on the MONO-Project framework [L2] and can be integrated with different datasets and repositories. Inside the COVR project, PLUME can interact with the Toolkit management system.

Links:

[L1] <https://www.safearoundrobots.com/home>

[L2] <https://kwz.me/hvQ>

[L3] <https://www.safearoundrobots.com/toolkit/home>

Reference:

- [1] M. Askarpour, et al., “Formally-based Model-Driven Development of Collaborative Robotic Applications”, in *J. Intell, Robotic Syst.* 102(3): 59 (2021). <https://doi.org/10.1007/s10846-021-01386-2>

Please contact:

Antonello Calabrò, ISTI-CNR, Italy
antonello.calabro@isti.cnr.it

Behaviour Trees for Representing Human-Robot Collaboration Processes in the World-Wide Lab

Mohamed Behery, Philipp Brauner, Martina Ziefle and Gerhard Lakemeyer (RWTH Aachen University)

Shorter product lifecycles, more product variants, individualised production, and the desire for sustainable production call for agile control frameworks that enable smarter robotic control and collaborating human-robot teams. We propose generalising and standardising “Behaviour Trees” that use human action nodes as a process model and task-execution-monitoring approach for human-robot collaborative assembly processes to increase the agility of human-robot teams while ensuring a safe and trusted human-robot interaction. Within the DFG (Deutsche Forschungsgemeinschaft) funded Cluster of Excellence “Internet of Production”, we take a cross-disciplinary approach to conceptualisation and validation to ensure algorithmic soundness, technical viability, and social acceptance by the workers of increasingly agile human-robot teams.

Production processes that involve Human-Robot Collaboration (HRC) are required to meet the demands for more product variant requests, initialised production, and short product lifecycles, which calls for agile control frameworks that ensure a safe, trusted, and socially accepted integration of humans in the robots’ workspace.

Moreover, the exchange and replication of production setups and pipelines is desirable in the World-Wide Lab (WWL) [1] and requires an abstract modular transferable process representation. The Cluster of Excellence Internet of Production (IoP) [L1] tackles these problems, among others, using process Digital Shadows (DSs) [2]. DSs are context- and task-specific process models that can be stored in online servers to be used in many ways including process control, verification, and decision support systems. Additionally, they can be used to efficiently couple the different entities of the WWL.

HRC processes have so far not been integrated into the WWL. The integration is challenging due to the lack of DSs for these processes. This can be attributed to the ad-hoc robot-specific programming involved in these use cases, which can be laborious and time consuming. They are inherently challenging because they cover a wide range of setups, all of which have a proximity to human workers.

HRC processes need DS to enable real-time introspection, code reuse, and integration into the WWL. A DS of an HRC process must conserve the safety requirements (and guarantees) of the process. For example, an HRC assembly must represent the requirement of a fully assembled product while maintaining the safety of the human co-worker.