

Users	<ul style="list-style-type: none"> • Requirement engineers without any knowledge related to formal methods. • Software developers without any knowledge of formal methods and logical languages. • System engineers interested to formally verify a model w.r.t. some properties.
Key Features	<ul style="list-style-type: none"> • Automated consistency checking of requirements expressed in natural language. • Automated synthesis of high-level task-oriented optimal “correct-by-construction” policies (no verification needed).
Benefits for the User	<ul style="list-style-type: none"> • Automated consistency checking of a set of requirements written in controlled natural language. • No prior knowledge related to specification languages is required to input the requirements (GUI support). • Human-readable feedback in the case of inconsistent requirements. • Domain and application independent.
Inputs	Set of requirements in natural (controlled English) language, formulated as PSPs for Linear Temporal Logic (LTL) extended to constrained numerical signals
Outputs	Consistency result (yes/no). In the case of inconsistency, the tool returns the minimal set of requirements that causes the inconsistency.
Block Design	
Example of Usage	Let consider a set of requirements from the design of an embedded controller for a robotic manipulator. The controller should direct the robotic arm for a given task, e.g, to look for an object placed in a given position and move to such position in order to grab the object. In addition, the robot must stop also in the case of an unintended collision with other objects or with the robot itself. Finally, if a general alarm is detected, e.g., by the interaction with a human supervisor, the robot must stop as soon as possible. In this case, constrained numerical signals can be used to represent requirements related to various parameters, e.g., angle, speed, acceleration, and torque of the joints, size of the object picked, force exerted by the end-effector, etc. Aim of the tool is to check the consistency of the controller requirements (expressed in controlled natural language) and, in the case of inconsistency, return valuable feedback to the designer.
Role in the Toolchain	Requirements verification at the early stage of the design process