Soar is a theory, implemented as a software architecture, that seeks to describe and realize the fundamental, functional components of intelligence. Soar has proven very successful at integrating disparate agent capabilities in performance systems. These capabilities have included planning, plan execution, natural language understanding, natural interaction with humans, interagent communication, and teamwork, among many others.

The Soar research community has concentrated more on realizing individual capabilities and creating complex agent systems than on understanding how to manage the knowledge that these systems require. At present, much research in the Soar community is directed towards addressing this oversight by developing more efficient methods of capturing, encoding, and reusing agent knowledge. Examples of this work include learning from observation (automatic knowledge synthesis via mimicking a human expert), applying the lessons of modern software engineering (e.g., strong typing of agent representations through both development tools and changes to the Soar rule syntax), developing algorithms for knowledge verification, and creating design patterns for frequently used capabilities like planning.

An obvious potential research direction is to develop knowledge with which agents can produce additional knowledge. We assume that Soar agents can plan by giving them planning knowledge; however, can we develop a Soar agent that learns to plan? In the past few years, researchers in both artificial intelligence and control systems have begun to address this question. One thing that sets Soar apart from other efforts is that Soar’s primitive mechanisms have already been used to develop complex agent systems. What we have yet to show is that competent agents can be achieved by starting with an agent that has very little general knowledge and must autonomously learn from its environment.

Soar researchers apply Soar to increasingly complex application problems in order to better understand the capabilities of the architecture. These systems have demonstrated state-of-the-art capability while also feeding back into basic research, driving refinements to Soar. Most recently, these Soar applications have been developed in simulation environments. While these domains are complex and require real-time behavior, a logical next challenge would be to apply Soar to complex, embedded systems, such as unmanned vehicles. Such an application would likely raise many research issues. Some obvious research issues in unmanned vehicles include:

- Reasoning under uncertainty: While simulations also have elements of uncertainty, perception and action are both much less certain in real environments.
- Satisficing: In some situations, an agent may not have time to consider all options and possibilities. An agent should be able to take reasonable action with the information at hand, even if incomplete.
- Performance monitoring and fault localization: As errors occur, the agent should be able to recognize problems and determine the source of errors.
- Robustness: An agent needs not only knowledge of the task at hand, it also needs more general knowledge that will allow it to reason about situations not anticipated by its developers.
- Symbol grounding: In simulation, an agent can be given symbolic interpretations of its environment. In real environments, we will need to understand how to convert sensor signals to symbolic representations.
- System validation: Systems that behave in the real world will need to be trustworthy. Trust implies provably correct, predictable behavior.

Undoubtedly, applications in unmanned vehicles would expose assumptions and weaknesses in Soar, driving further basic research on Soar itself. However, given the success of many years of building practical applications to test the comprehensiveness of the architecture, Soar is an ideal candidate for high-level control of unmanned vehicles.