

Soar Technology, Inc. Autonomous Platforms Overview

Point of Contact

Andrew Dallas

Vice President Federal Systems

(734) 327-8000

adallas@soartech.com



Since 1998, we've studied and modeled many kinds of human behavior in order to create systems that "think the way people think:" constantly learning, getting smarter, and adapting to new times and situations. Our work is rooted in cognitive science, a deep understanding of human perception, memory, performance, learning, and emotion. To this we add a careful analysis of domain knowledge, built into our systems so they work the way people work.

These technologies are being used to build intelligent control architectures in support of autonomous and collaborative robotic systems. Robotic intelligence frees humans from being "tethered" to a robot – and makes the robot a true "teammate" instead of a piece of equipment. SoarTech's control architectures are capable of working with single or multiple platforms and incorporate autonomous behaviors for task planning and mission planning; moving in formation; approaching a detected object; forming, joining, or leaving a team; and learning behaviors from demonstration. Through this work, SoarTech has also become a leader in the development of smart and intuitive human/robot interface devices. Our Smart Interaction Device (SID) architecture is the core of this system and has been used for both ground and air robot interaction.

SoarTech has developed its robotic technical base by executing programs in robotics, teaming with known entities and programs in the field of robotics, and bringing in outside expertise in robotics as employees or consultants.

The information below describes our expertise in ground robotics. Note, our particular strength is in robot perception, robot planning/control, intelligent robot architectures, robot behaviors, human robotic interfaces, and cognitive reasoning that is critical for the continued advancement of autonomous and semi-autonomous robotics.

Mission Level Control for Autonomous Platforms

Intelligent agents for robotics control minimize the percentage of time devoted to interaction with a robot or UV, and maximize the percentage of time that the operator can neglect the robot while maintaining oversight of the robot's actions. SoarTech's control architectures are capable of working with single or multiple platforms and incorporate autonomous behaviors for task planning and mission planning; moving in formation; approaching a detected object; forming, joining, or leaving a team; relinquishing/accepting control of an asset; conducting a roadblock; and learning behaviors from demonstration.

Current Approaches and their Limitations

When it comes to ground vehicles, state-of-the-art unmanned vehicle autonomy currently depends heavily on a combination of deployment in a controlled environment and detailed prior-knowledge of that environment (e.g., high-fidelity road maps or charts). Without either of these constraints, robotic vehicles must be tele-operated; all currently deployed autonomous vehicles (i.e., non-research systems used in practice) are tele-operated most or all of the time. The fundamental technical challenge that prevents more pervasive autonomy is the problem of making good decisions under the vast number of situations that an autonomous vehicle can encounter in the real world. In the limited cases where existing systems have succeeded in overcoming this complexity, it is because they have applied algorithms that can adapt during execution.

There are limitations of autonomous air control systems as well. The current state of the art in autonomous aircraft control is in Flight Management Systems (FMSs), which handle many critical aspects of dynamic aircraft control such as maneuvering, path planning (including waypoint constraints), and some types of fault/failure recognition and handling. However, current FMS systems do not reason about mission-level decisions, making them insufficient to execute autonomous Close Air Support (CAS) missions. For example, FMSs do not develop an understanding of their mission, they do not choose or execute complex maneuvers or tactics based on weapons delivery or targets, they do not choose routes based on mission threats, they do not consider weapon damage or visibility when making weapons approach (if they can make weapons approaches at all), and, perhaps most importantly, do not interact with users to accept CAS missions or to share their situational awareness with others. Similarly, ground control stations (GCSs) are designed primarily for remote control of aircraft, often providing sensor feeds, basic aircraft information, and ways to maneuver the aircraft by specifying routes or waypoints. The GCS relies almost entirely on the user to process information and share situational awareness with other participants. Furthermore, lags and the potential for dropped connections make it difficult to impossible for the GCS operator to execute the complex, time-critical maneuvers that CAS aircraft often perform. With this minimal mission-level support provided by existing state of the art and the potential for increased complexity due to the addition of UAVs, a JTAC and safety pilot will be challenged to control even a single remotely piloted aircraft for single target prosecution.

To improve the level of autonomy in ground and air systems, these systems need to become smarter. They need to be able to learn their environments, be able to identify their surroundings, and be able to interact and remain in contact with users.

Intelligent Agents. TacAir-Soar is a model of the tactical decision making of a pilot flying missions in the tactical air domain. It provides high-fidelity, entity-level behaviors for a wide range of simulated aircraft and missions:

- Realistic computer-generated forces (CGFs) behave at human level, in real-time
- Many entities can be managed by few operators; complex training interactions require minimal operator oversight
- Natural voice or electronic communications
- HLA radio interface
- Simple, efficient interface for configuring missions
- Integrates with existing simulations

The key to simulation is believability, and TacAir-Soar behaviors are more realistic than those of any other currently available system not controlled by a human. Because TacAir-Soar entities are based on a model of human cognition, they reason, decide, and act similar to the way a human would. Since TacAir-Soar is fully autonomous, only a small number of human operators are required to run a simulation -- even an operation with hundreds or thousands of entities. The Soar architecture places no limit on the total number of agents, and TacAir-Soar provides multiple entities per machine. This makes very large-scale simulations cost effective, since it is much less expensive to add another machine than another operator.

TacAir-Soar entities are not limited to performing their pre-briefed missions or following a script. They coordinate their actions through shared goals, planning, and explicit communication. Even though each entity is autonomous, it does not act in isolation. Entities maintain high-level goals while reacting to

changes in their environment, and coordinate their actions using existing doctrine and C⁴I systems. As the mission develops, entities may change roles dynamically. Entities may communicate with other entities, other simulators, or humans. Human controllers can redirect and retask aircraft through doctrinally correct radio messages in plain English, just as they would control real aircraft. Current product innovations are providing instructors with the ability to direct TacAir-Soar entities to behave in ways that enhance pedagogical goals in addition to their standard doctrinal behaviors.

In addition to TacAir-Soar, SoarTech has developed and evaluated a cognitive reasoner that is capable of tactical mission planning for a ground vehicle. The technology behind our approach is the Soar cognitive architecture, which provides fundamental processes that support complex inference and real-time learning. Our solution consists of behavior models built on this architecture that ingest both vehicle state and inorganic report data and computes the best way, given mission constraints and goals, to get from the current location to the destination. This behavior model also learns from its prior experience, and is able to plan faster and safer routes in cases similar to those it has observed before. It is also able to explore the environment in cases where its experience indicates that prior paths have been unsatisfactory.

Development Approach. Our intelligent agents consist of three elements.

1. Cognitive Reasoner:

A cognitive reasoner provides a unified place to do heuristic and special case reasoning and to expand this reasoning to new cases using learning. Unlike the *ad hoc* special case reasoning that is very common in robotics, cognitive reasoners can apply consistent decision processes to all contexts and can even resolve conflicts and ambiguities using a centralized decision process. We have shown in prior simulation work that it is possible to scale this type of reasoner to a very large number of missions and situations without losing responsiveness and decision consistency.

2. Learning Mechanisms:

Episodic reasoning has potential to make autonomous behavior more robust. Our efforts to date have shown the following reasoning and behavior improvements:

- 1 Vehicles were able to complete missions faster in many cases.
- 2 Vehicles were able to remember and avoid issues from similar missions, even when it was not told about these issues directly.
- 3 The vehicle's behavior was naturally exploratory with exploration gradually increasing the more issues the system experienced.
- 4 The vehicle was able to learn very quickly (sometimes in as little as one example).

3. Integration of cognitive reasoning with perception and planning.

Our approach to integrating cognitive reasoning with perception is to use a spatial reasoner to mediate between the cognitive reasoning and perception system. For example, when the cognitive reasoner recognizes that a route is sub-optimal, it may decide to explicitly use the alternate route. The way this intention is communicated to the rest of the system is by lowering the cost of the alternate route in an attempt to nudge the base planner into selecting that route. Many times it does, but other times it does not and, as is often done with cost-based solutions, the developer must play around with costs and weights (i.e., "tuning") to get just the right balance. But the balance is fragile, and this balancing act is fundamentally unnecessary. The CR "knows" exactly what it wants to do (in this case, try the alternate

route) but has no way to directly express it using the cost map as an interface. Other integration and planning architectures could be built to better exploit the CR's knowledge, and we are exploring these other options.

Intelligent User Interfaces

Our method to producing human level behavior for Autonomous Platforms is exemplified through our approach to supervisory control of unmanned systems. Though traditionally thought of as a robotics issue, Supervisory Control is mentioned here as the basis for our approach to Human-Robotic Interaction (HRI). Supervisory Control provides the constructs from which to build more human-centered HRI.

When implementing supervisory control, the user's job changes to one of managing the robot rather than controlling it in a tele-operation mode. As adapted from Sheridan's approach, we think of the three requirements for supervisory control as autonomy, high-level commands, and situational awareness. This occurs routinely in human teams that include a leader and a subordinate: the leader gives a high-level task, and the subordinate is given some autonomy to perform the task while at the same time keeping the leader informed of progress or problems. For supervisory control in human-robot teams, the same conditions must hold: the robot must have autonomy to perform at least some tasks, must accept high-level tasking (far above tele-operation), and must provide feedback to the user to help maintain the user's awareness. Autonomy in unmanned platforms has been slowly increasing over time, but OCUs have tended to lag behind in terms of providing high-level tasking, providing effective situational awareness to the user, and in usability generally.

Current Approaches and their Limitations

There are currently two main methods for interacting with most unmanned vehicles (UxVs): through tele-operation or through point-and-click map-based interfaces. Both of these require almost complete attention by the UxV operator. These are the primary means of controlling UxVs for two reasons: UxVs' *lack of autonomy* and *lack of smart user interfaces*. UxVs that lack autonomy altogether are essentially remote controlled vehicles. Planned UxVs are beginning to show elements of autonomy (e.g., DARPA's Urban Challenge, ONR's SUMET program), but autonomy is largely applied to navigation: getting from point A to point B without getting stuck or hitting obstacles. This kind of autonomous navigation is necessary but not sufficient, especially when these platforms must interact with users to accomplish their missions.

User interaction with the AUGV through an operator control unit (OCU), whether through tele-operation or map-based interfaces, also remains a burden. The user must first determine tasking for the AUGV in mission terms. Once users know what they want the UxV to do, they must translate their intent into the language and forms provided by the OCU, often at a very low level of tasking. Forcing the user to adapt to the system (rather than the other way around) puts much more burden on the operator, and makes the UxV less useful. This is in contrast to human teams, in which a team lead can essentially express tasking directly through a variety of modalities – speech and gesture, for instance. Tasking the UxV is only part of the problem: feedback to the user is also impoverished in current systems. Status information about the platform is often shown as indicators on the OCU display or with video feeds, requiring a great deal of training to use and constant eyes on the OCU. In some situations these are warranted, but other modalities such as generated speech could reduce the burden on the operator.

To be more useful on the battlefield, AUGVs must incorporate *smart, interactive user interfaces* that allow natural modes of communication between the user and the UxV, including natural modes of feedback to the user.

Smart Interaction Device. Control of unmanned systems (UxS) today is dominated by the use of bulky, laptop-sized operator control units (for ground vehicles) or ground control stations (for air systems). Besides being bulky, they are also hard to use, requiring months of training to become certified, and force the operator to think at a low level of detail to command the vehicle.

Our approach is introducing small, ultra-portable devices that include an *intelligent user interface* to help raise the level of interaction between the human and machine to the level of supervisory control. Over the last few years, we have been developing the **Smart Interaction Device (SID)** whose purpose is to enable natural HRI for interaction between a human and machine, see Figure 1. SID facilitates high-level, multi-modal interaction between a user and associated machines. SID leverages the automation provided on the platform, but also adds intelligence in the user interface itself that helps to translate high-level user commands into machine-readable commands. SID also helps maintain user situational awareness by providing feedback to the user, either by request or based on expected protocols. The result is that the user can spend less time minding the machines they are working with, and more time paying attention to his or her surroundings, working on other tasks, or managing multiple vehicles.



Figure 1: SID is an intuitive, multi-modal interface for robotic control.

General Emphasis on handheld/tablet approaches

In current operations, there is a wide array of gear that might be carried in the field to be used for communication. Different communication modes might be used at different times or under different conditions, and may not necessarily be used in concert. For example, for autonomous rotary wing aircraft operation, the VHF/HF radio may be used to request resupply or CASEVAC, with enough information to get the aircraft within a few kilometers of a landing zone. Light signals on the ground might indicate the landing zone with minimal need for further radio calls. Once near the landing zone, a Soldier on the ground might use hand/arm signals to guide the aircraft to the ground, though they are not trained in formal signaling. Of course, this is the ideal case where nothing goes wrong and no adjustments need to be made. In the event of trouble, people will use whatever signals they can to

communicate, which may include mixing modes or switching modes unexpectedly.

Going beyond the current technologies in the field, we are looking at tablet computers as an interesting alternative (especially for UAS systems). Tablet computers open up the possibility for different input and output modalities. Most tablet computers today allow for touch displays with the ability to sketch on the screen. Conversations about terrain and physical positions benefit from having shared artifacts such as maps which could be displayed on the tablet and sketched on or otherwise referred to. Conversations about landmarks or obstacles in the environment benefit from having eyes on those things, so video or imagery on the tablet could be extremely beneficial. A digital display also allows for more permanence in the information that is conveyed: whereas spoken language must be remembered or deliberately recorded, text or imagery on a digital display can remain for future reference.

SID Development approach. SID consists of a domain-independent set of core modules (“SID Core”) that are designed to accommodate the kinds of high-level interactions that occur in these domains. SID additionally includes input-device-specific, domain-specific and platform-specific layers, each of which is customized to a particular application. Different ways of interacting with the robot and different robot capabilities necessitate different user-facing and robot-facing software/hardware interfaces.

Domain-*independent* knowledge in SID consists of rules for how to manage dialogues, how to break a task down into finer tasks, and how to maintain situational awareness about a platform. Domain-*dependent* knowledge consists of rules for how translate a particular user command to a particular platform-level command, or how to interpret particular input modes in a given domain.

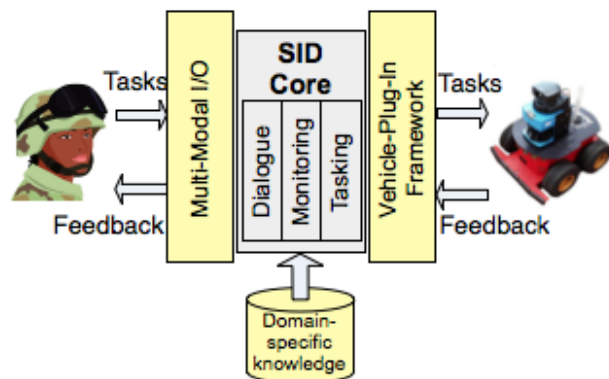


Figure 2: A description of the SID technical architecture.

As illustrated in Figure 2, the main behavior components in SID Core are the Dialogue Manager, and Monitor, and the Tasker:

- The **Dialogue Manager** is responsible for all interactions with the user, including interpreting user intent, asking clarification, and providing feedback to the user about the status of the robotic platform
- The **Monitor** is responsible for maintaining awareness of the state of the robotic platform, especially as it pertains to fulfilling the user’s intent
- The **Tasker** is responsible for translating user’s intent into commands for the robotic platform.

These modules work over a common shared memory representation that maintains the state of the vehicle, the state of the conversation, and the current plan assigned to the vehicle. This allows all modules in SID Core to have access to the task definition and progress as task execution proceeds. This shared representation also allows SID to engage in a dialogue with the user before, during, and after task execution.

Situations, Actions, Goals, Environments Interface (Sapient). While existing OCUs are still fairly rudimentary, significant work has been done at the theoretical level that focuses on more effective ways to provide situational awareness to the operator through the OCU. In particular, Endsley and others have identified several factors that should play a role in OCU design. At a high level these include the three levels of SA: Perception of elements in the environment (level 1), comprehension of the current situation (level 2), and projection of future status (level 3). Our second line of work in HRI specifically targets situation awareness aiding the operator in maintaining these levels of awareness for a mission that includes many ground robots.

Current Approaches and their Limitations

As previously discussed, the current operational standard in UV situation awareness and control is tele-operation via an operator control unit (OCU). Research has shown that these types of OCUs are difficult for operators to use, in large part because the operator has a difficult time maintaining situation awareness even though their attention is completely focused on the UV. For example, in some experiments, operators spent an average of 30% of each run “acquiring SA while no other task was being done. Despite this time spent trying to acquire SA, users often expressed confusion about where their UVs were located ...” There are many reasons for this difficulty, including the independent movement of camera and vehicle, difficulty perceiving depth and distances via a camera, and poor overhead map design. Furthermore, when OCU displays are cluttered, it is difficult for the operator to pick out the specific information that is needed. Thus one thread of research has focused on providing information via multiple modalities such as voice or force feedback.

OCUs developed for autonomous systems present some advantages. For example, a small UAV interface provides an overhead map as a primary display, thus allowing the user to maintain better mission SA. This is possible because the operator does not need to directly control the vehicle and thus doesn’t require a first person view or high fidelity controls.

The multi-display OCU (Figure 3) developed by The University of Michigan and SoarTech in the MAGIC International Robot Competition (MAGIC) took this concept a step further to enable two operators to control a team of 14 unmanned ground vehicles (UGVs). The MAGIC system included semi-autonomous UVs that did not require constant tele-operation, allowing the displays to be focused on providing mission-level awareness. However, the MAGIC OCU (including SoarTech’s SAGE interface) went even further and provided (1) simplified display of vehicle tasks and pose, (2) simplified projection of future actions, (3) automatic detection and highlighting of events of interest, and (4) user correction of certain aspects of situation awareness (primarily map structure) via a few simple commands.

These OCUs, and other similar products, share a common limitation – they all operate on homogeneous UV teams. We anticipate that *heterogeneity* will add significant complexity over the existing state of the art. This added complexity must be addressed in order to maintain or reduce current state of the art operator: UV ratios. Examples of such a complexity include the operator’s need to maintain SA at multiple scales (e.g., ground v. air assets), know diverse UV control mechanics, and handle a wider variety of threats and situations. To date research and development of C2 solutions for heterogeneous teams is limited.

The Genesis of Sapient—MAGIC 2012. Sapient solves a fundamental problem in current ground robot operations – the large amount of attention and interaction required to control a team of unmanned vehicles. This work was accomplished as part of our winning effort MAGIC 2010. The competition presented each team with the challenge of directing a team of semi-autonomous ground robots to map a large space (indoor and out) while looking for and disabling “dangerous” objects. This task was designed to mimic the type of intelligence, surveillance, and reconnaissance (ISR) missions typically presented to small military units. Each team had to deploy at least three robots (our team deployed fourteen) and was allowed only two human operators. Operators were penalized for interacting with any part of their system, thus a premium was placed on autonomy – not just of the robots, but of the operator control units as well. Therefore, the competition provided an excellent framework within which to research and test ideas for minimizing user-robot interaction while maintaining or increasing operator situation awareness of high-level tasks.



Figure 3: The University of Michigan/SoarTech MAGIC OCU. Soartech’s interface is shown on the right most screen.

Development approach. Whereas, SID is developing the intelligent communication infrastructure for HRI, Sapient is developing the intelligent display of information on the OCU so that the human and machine can share situation awareness necessary for team decision making. Sapient:

- monitors the mission and behavior of a team of heterogeneous UVs,
- presents the user with filtered, fused, and focused information about the mission,
- detects and projects important events and alert the user to them,
- detects and reconciles with the user potential situation ambiguities, and
- automates the presentation of critical contextual information to the user.

Sapient enables the management-by-exception control scheme – allowing operators to spend less time interacting with the system and more time doing other important tasks.

To achieve these results Sapient is implementing computational situation awareness (CSA), and using it as a basis for intelligent control of UI displays that visually fuse information from disparate sources. Situation Awareness can be thought of as consisting of three processes:

- “perception, the awareness of cues and information in the environment;”
- “comprehension, the retention, interpretation, and combination of perceptual information to provide meaning;” and
- “projection, the ability to anticipate future events.”

In Sapien we are encoding these SA processes into the OCU itself. In this way the OCU can share information in terms that aid, rather than distract, the operator. We call this solution computational SA (CSA) because the system is generating its own SA. Computational SA will act as a surrogate user maintaining attention on the mission and sub-tasks so that the user can attend to other tasks. When opportunities and issues are encountered, it communicates SA to the operator via visualizations that fuse the current organic situation (e.g., UxV position, sensor views) with relevant historic and inorganic data (e.g., blue force tracker and hostile activity data).

The centerpiece of our computational SA concept is the Soar architecture. Soar is a computational theory of human intelligence implemented as a virtual machine. Soar contains computational elements designed to mimic human memory, reasoning, and decision-making. These elements, such as working memory, procedural memory, semantic memory, and episodic memory match well to the core human elements of SA as described by Endsley. Using the Soar architecture together with supporting processes such as spatial reasoning components, we will research and design algorithms for maintaining and sharing SA with the user.

While our primary focus is on the problem of situation awareness, Sapien is also integrating with control systems and allowing users to issue high-level commands that Sapien will map to appropriate UxV commands. In another recently awarded effort, we are adding advanced decision support and control schemes to Sapien. To ensure that Sapien is flexible enough to meet deployment requirements, we are designing Sapien to be as independent as possible from specific computer, network, and display systems. Where appropriate we are making the use of standards, e.g., the variable message format (VMF) in FBCB2.

Conclusion

Our approach to supervisory control increases the operator’s ability to neglect entities while they are handling their own tasks, and decreases the amount of time needed to interact with said entities during those times they do need supervision. The future of unmanned systems builds on this, and reaches towards the integration of unmanned machines in society with humans. One hurdle to robotic integration is societal trust of these machines and systems. Increased trust will come from helping users feel more comfortable with the interaction, by making human-robot interaction more natural, and ensuring better situational awareness of the robot via improved communication between the two. As a leader in the world of cognitive systems, SoarTech has been working for the past 15 years on behavior models that help improve autonomous systems, and bring together the control of systems with the user-interface necessary for humans to use them as seamlessly as possible, and makes the robot a true “teammate” instead of a piece of equipment.