The 2016 AAAI Fall Symposium Series:
Cross-Disciplinary Challenges for Autonomous Systems
Technical Report FS-16-03

# ActorSim, A Toolkit for Studying Cross-Disciplinary Challenges in Autonomy

**Mark Roberts,**[1] **Laura M. Hiatt,**[2] **Alexandra Coman**[3]
**Dongkyu Choi,**[4] **Benjamin Johnson,**[5] **David W. Aha**[6]

[1]NRC Postdoctoral Fellow; Naval Research Laboratory, Code 5514;
Washington, DC | mark.roberts.ctr@nrl.navy.mil

[2]Naval Research Laboratory, Code 5515; Washington, DC | laura.hiatt@nrl.navy.mil

[3]NRC Postdoctoral Fellow; Naval Research Laboratory, Code 5514;
Washington, DC | alexandra.coman.ctr.ro@nrl.navy.mil

[4]Assistant Professor, Aerospace Engineering, University of Kansas,
Lawrence, KS | dongkyuc@ku.edu

[5]NRC Postdoctoral Fellow; Naval Research Laboratory, Code 5514;
Washington, DC | benjamin.johnson.ctr@nrl.navy.mil

[6]Naval Research Laboratory, Code 5514; Washington, DC | david.aha@nrl.navy.mil

**Abstract**

We introduce ActorSim, the Actor Simulator, a toolkit for studying situated autonomy. As background, we review three goal-reasoning projects implemented in ActorSim: one project that uses information metrics in foreign disaster relief and two projects that learn subgoal selection for sequential decision making in Minecraft. We then discuss how ActorSim can be used to address cross-disciplinary gaps in several ongoing projects. To varying degrees, the projects integrate concerns within distinct specializations of AI and between AI and other more human-focused disciplines. These areas include automated planning, learning, cognitive architectures, robotics, cognitive modeling, sociology, and psychology.

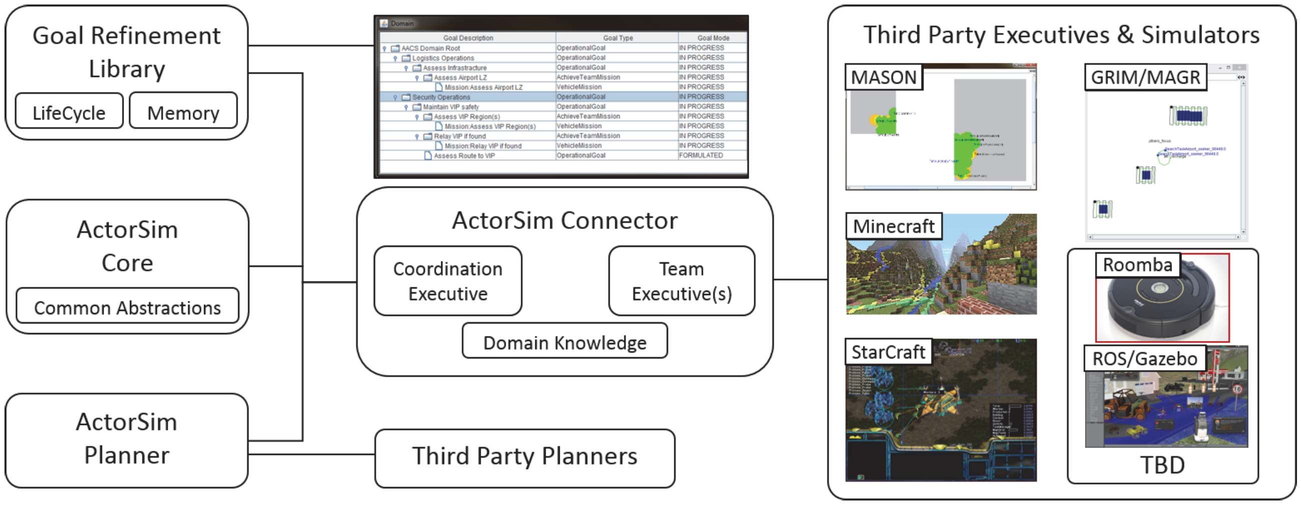
## 1 Introduction

Autonomous systems are increasingly called upon to augment human capabilities; i.e., to work with a human in an assistive way or to perform tasks that humans would not otherwise be able to do. A clear example of such integration is that of robotic systems, which have become a common addition to many industrial and residential applications and are widely used within close proximity to humans.

The challenges of autonomous systems most often arise from the integration of distinct disciplines of research. In this paper, we discuss two kinds of cross-disciplinary challenges. First, we focus on cross-disciplinary challenges within the field of Artificial Intelligence that pertain to the integration of two or more sub-disciplines. Specializations such as machine learning, vision, automated planning, or cognitive systems (to name some that we focus on in this paper) have substantially advanced our understanding of learning, perception, and reasoning. Yet the way these specializations represent and solve problems is distinct enough to create challenges in how to integrate them effectively and in how to generalize a technique perfected within one specialization. Second, we focus on the continued integration of insights from human-focused sciences such as sociology, psychology, and cognitive modeling. Challenges in this focus relate to how to integrate insights from those sciences or how to provide those sciences with tools that advance knowledge about human behavior and cognition. Addressing both kinds of challenges will be important because an embodied autonomous system will usually integrate insights from a variety of these fields.

We highlight a toolkit called ActorSim, which we believe provides a strong starting point for studies of autonomous systems. We begin with a description of ActorSim (Section 2) followed by a brief review of its use in several existing projects (Section 3) that have already taken some steps toward addressing within-AI challenges. We then highlight three projects that both extend this focus and broaden ActorSim's cross-disciplinary areas to include human-focused sciences. The first project (Section 4) will extend ActorSim with a cognitive architecture in support of long-term autonomy and perpetual learning in open environments. The second project (Section 5) will build on this cognitive architecture to improve robot safety by leveraging work from cognitive science and robotic safety. The final project (Section 6) seeks to integrate findings from psychology, sociology, and narratology.

Throughout the paper we will focus on our research agenda, which is guided by several questions concerning an autonomous system's interactions with either its environs or other entities; these questions include: How to improve its decision making with experience and context? How to determine the extent to which it should modify or disregard instructions? How to justify and clarify its choices to a human? How to embed information metrics for making priority decisions among goals? Another set of questions originate from our assumption of modeling decision making as hierarchical decomposition via Goal-Task Networks (Alford et al. 2016) and include: How to design or learn heuristics for choosing the best decomposition for a particular context?



| Goal Description | Goal Type | Goal Mode |
|---|---|---|
| AACS Domain Root | OperationalGoal | IN PROGRESS |
| Logistics Operations | OperationalGoal | IN PROGRESS |
| Assess Infrastructure | OperationalGoal | IN PROGRESS |
| Assess Airport LZ | AchieveTeamMission | IN PROGRESS |
| Mission:Assess Airport LZ | VehicleMission | IN PROGRESS |
| Security Operations | OperationalGoal | IN PROGRESS |
| Maintain VIP safety | OperationalGoal | IN PROGRESS |
| Assess VIP Region(s) | AchieveTeamMission | IN PROGRESS |
| Mission:Assess VIP Region(s) | VehicleMission | IN PROGRESS |
| Relay VIP if found | AchieveTeamMission | IN PROGRESS |
| Mission:Relay VIP if found | VehicleMission | IN PROGRESS |
| Assess Route to VIP | OperationalGoal | FORMULATED |

Figure 1: The Component Architecture of ACTORSIM

How to determine when to enlist a reactive approach rather than decompose? How to safely interleave execution for potentially hundreds of simultaneous goals?

## 2 The Actor Simulator, ACTORSIM

ACTORSIM[1] (Figure 1) is a general platform for conducting studies of autonomy in simulated environments; it was originally conceived of during projects centered on the study of goal reasoning. Although goal reasoning has strong ties to planning, acting, and robotics, it is an understudied area of research partly because there exists no publicly available language, definition, and generic implementation. We view goal reasoning as leveraging the work of Ghallab, Nau, and Traverso (2014, 2016), wherein deliberation takes place on (1) descriptive models of *what* to accomplish and (2) operational models of *how* to perform a task. A descriptive model for a goal to be `(in room`$_a$`)` might be decomposed into the subgoals `moveTo(door`$_a$`)`, `open(door`$_{toA}$`)` if applicable, and `enter(room`$_a$`)`. An operational model of opening the door is a more detailed sequence of actions such as: determine the type of door, door configuration, and handle, grasp the handle, unlatch the handle, and push (or pull or slide) the door. Thus, the deliberation in such systems resembles a hybrid goal-task network with goals interspersed. To this we add goal reasoning, which supports Planning and Acting by allowing an actor to determine and prioritize its goals dynamically.

ACTORSIM partially implements the goal lifecycle of Roberts et al. (2016), shown in Figure 2. ACTORSIM complements existing open source planning systems with a standardized implementation of goal reasoning that is run by a goal reasoner, or GRPROCESS. The main components consist of the following.

ACTORSIM **Core** provides the interfaces and minimal implementations of the platform. It contains the essential abstractions that apply across all simulators. This component contains information about Areas, Locations, Actors, Vehicles, Symbols, Maps, Sensors, and configuration details.

ACTORSIM **Planner** contains the interfaces and minimal implementations for linking to existing open source planning systems. This component unifies Mission Planning, Task Planning, Path Planning, and Motion Planning. It currently includes simple, hand-coded implementations of these plan-

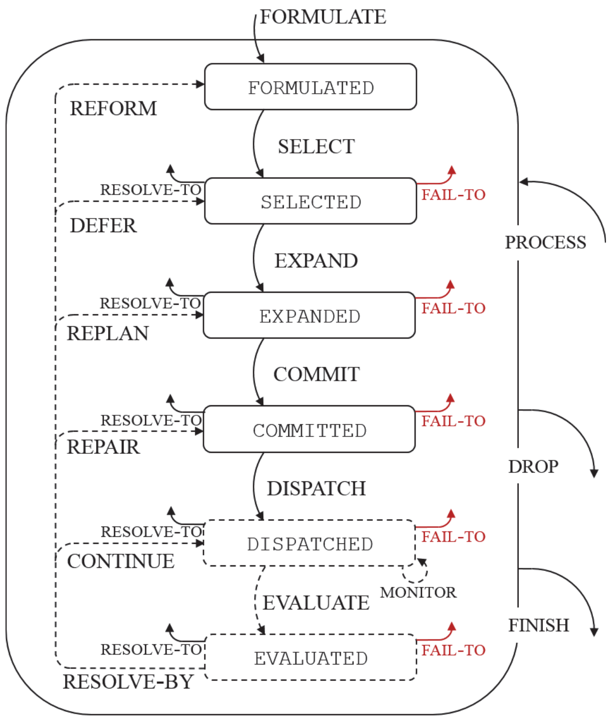


Figure 2: The goal lifecycle. Refinement strategies (arcs) denote possible decision points of an actor, while modes (rounded boxes) denote the status of a goal in memory.

[1]Available at http://makro.ink/actorsim/

ners, although we envision linking this component to many open source planning systems.

ACTORSIM **Connector** links to existing simulators directly or through a network protocol. Currently supported simulators include George Mason University's MASON (Luke et al. 2005) and two computer game simulators: StarCraft and Minecraft. We envision links to common robotics simulators (e.g., Gazebo, ROS, OpenAMASE), additional game engines (e.g., Mario Bros., Atari arcade, Angry Birds), and existing competition simulators (e.g., RDDLSim, Robocup Logistics League).

ACTORSIM **Coordinator** (not shown in the figure) houses the interfaces that unify all the other components. This component contains abstractions for Tasks, Events, Human interface Interaction, Executives (i.e., Controllers), and Event Notifications. It uses Google's protocol buffers[2] for messaging between distributed components.

The **Goal Refinement Library** is a standalone library that is integral to ACTORSIM, but could be used on its own. It provides goal management and the data structures for transitioning goals throughout the system. It contains the default implementations for goals, goal types, goal refinement strategies, the goal memory, domain loading, and domain design.

As ACTORSIM is applied to the projects we detail below, it should become more accessible for other researchers to use. A focus of the next year of development will be generating better documentation and tutorials to accompany the software with the aim of lowering the threshold for adoption of the tool by other researchers.

## 3 Existing Projects Using ACTORSIM

ACTORSIM has been used to study autonomy in Foreign Disaster Relief, a paradigmatic and Navy-relevant domain, and Minecraft, a challenging computer game studied by researchers (e.g., (Kope, Rose, and Katchabaw 2013), (Abel et al. 2015)) and for which a research platform was recently released by Microsoft Research (c.f., http://bit.ly/232bxhV).

### 3.1 Foreign Disaster Relief

*Cross-disciplinary focus: robotics, LTL synthesis, planning and execution.*

An early prototype of ACTORSIM was embedded in a larger system called the Situated Decision Process (SDP), which links hierarchical task planning (i.e., a goal network) with reactive vehicle controllers (Roberts et al. 2015). The controllers were automatically built by synthesizing correct-by-construction Finite State Automatas (FSAs) from a restricted variant of Linear Temporal Logic. This work introduced Coordination Variables that allowed task planning to control and receive feedback from a reactive layer. It also saved considerable computation during FSA synthesis. In a small demonstration, we showed that the SDP adjusts to notable events (e.g., finding a survivor) or retasks vehicles to continue stalled missions when such events occur.

This system was extended to a more recent study on Goal Reasoning with Information Measures (GRIM) (Johnson et al. 2016). In this system ACTORSIM was integrated with a Multi-Agent Goal Reasoner, which acted as the simulation engine (cf., top right of Figure 1). GRIM presented initial efforts towards grounding a goal reasoning system in the metrics used by the controlled vehicles during execution. We showed that the goal lifecycle provided a formal structure upon which to refine and resolve goals and, when paired with information measures, GRIM satisfied more goals than the baseline system without resolve strategies.

Future extensions will investigate additional goal types as well as more complex algorithms and information measures. The initial study used a simple approximation for the measure of uncertainty in the survey goal, and a more accurate measure of the uncertainty (e.g., tracking the total area covered by the vehicles sensors) and corresponding expectations would allow GRIM to improve its performance estimates and react accordingly. Additionally, a planner or scheduler could be used to SELECT goals while accounting for the likelihood of discovering an official in each region, thus enabling GRIM to more intelligently choose which goals to pursue. Likewise, adapting plan expectations (e.g., recognizing that the vehicles are not completing the survey at the expected rate, and changing the expectations accordingly) would enable GRIM to more quickly identify and evaluate problems, and thus improve the likelihood that it could RESOLVE-BY any discrepancies. These extensions will enable a more thorough evaluation of GRIM's benefits via an experiment with randomly generated scenarios.

### 3.2 Overcoming Obstacles in Minecraft

*Cross-disciplinary focus: planning and learning in episodic autonomy.*

ACTORSIM has also been used in two studies to learn subgoal selection in the game of Minecraft. Both studies task the GRPROCESS, acting as a virtual player, with controlling Steve to achieve the goal of navigating to a gold block through an obstacle course. We represent this as a top goal of moving to the gold block and with five subgoals (walk forward, walk around, build a staircase, mine, and build a bridge) that help the character lead to that objective. The order of subgoal choice impacts performance. These subgoals do not contain operational knowledge, so preconditions on actions ensure that the character will not violate safety by falling too far or walking into a pool of lava or water. The character's movement is axis aligned and discrete. We coded an expert decision maker that chose the best subgoal based on observations of the state of blocks directly around the player. We collected traces from the expert procedures, capturing the state, distance to the goal, sub-goal chosen, and whether the chosen subgoal succeeded.

In the first study, the character made decisions using these observations (Roberts et al. 2016). In addition to the expert choice, we implemented randomized choice and an ordered choice to study the impact of the training set on learning an effectiveness. We trained a decision using traces from Random, Ordered, and Expert procedures and showed that, for this limited domain, learning from structured exploration (i.e., the Ordered traces) is as effective as Expert exploration

[2]https://developers.google.com/protocol-buffers/

and costly knowledge gathering is ineffectual. In the second study (Bonanno et al. 2016), we extended subgoal selection to learn from raw pixels using deep learning. In a limited pilot study where a virtual character must overcome obstacles, we showed that a deep learning architecture built upon AlexNet learned an effective policy 93% of the time with very little training.

Future extensions will extend planning and learning. An important part of better learning is to apply deep learning (and other learning approaches) for object classification within Minecraft. Another learning task is to better recognize the context of the character. For better planning, we are extending the planning models to include richer variants of automated planning such as PDDL+ (Fox and Long 2006), and extending the tasks to include mining resources and surviving.

## 4 Perpetual Learning

*Cross-disciplinary focus areas: long-duration autonomy, planning and learning, and cognitive architectures.*

Robotic and autonomous systems increasingly operate for longer durations and often for longer than their engineered lifespan (e.g., the Mars Rovers). Yet many systems are often evaluated within a closed episodic paradigm with a minimal set of tasks. A long-running process will likely encounter a variety of tasks and contexts and could ideally progress its own agenda for over hundreds or thousands of tasks, some of which may be assigned externally. We call such a system a *perpetual learner* because it directs its own curricula to continually create and master new tasks, incrementally revises previously mastered tasks, or halts learning for tasks it has sufficiently mastered.

We aim to study perpetual learning in dynamic, open, multi-agent, simulated environments where a perpetual learner continuously performs hundreds of tasks appropriate to the context while responding to requests from teammates. Researchers have already investigated some aspects of perpetual learning, namely how to move beyond episodic evaluation, how to partition computational effort through decomposition, and how to maintain and update memory. The literature in these three areas provides a foundation from which to build, and we discuss the challenges of integrating them into a perpetual learner in the remainder of this section.

For moving beyond episodic learning, perpetual learning can leverage successes in never-ending learning (Mitchell et al. 2015), continual learning (Ring 1997), lifelong learning (Thrun and Mitchell 1995), and transfer learning (Isele, Rostami, and Eaton 2016), which have each significantly advanced our understanding of learning across tasks and time. One of the challenges is determining where and when to integrate these approaches into the perpetual learner.

At a minimum, a perpetual learner must maintain a retrievable memory of its current tasks and its own performance. Many disciplines contribute to the study of memory in intelligence. Here we plan to focus on the literature in cognitive systems, which has refined a crisp computational definition of memory structures. In particular, we will supplement ACTORSIM with key concepts from the Icarus architecture (Langley and Choi 2006) as shown in Figure 3. These will include a long-term memory to store knowledge, a short-term (i.e., working) memory to focus computation, plus a retrieval and learning processes to transfer knowledge between these. Space limitations prevent a detailed comparison between this proposed architecture and other cognitive architectures (e.g., SOAR (Laird 2012) and ACT-R (Anderson and Lebiere 1998)), but two distinctions worth noting here are ACTORSIM's the use of the goal lifecycle and the separation of planning and goal reasoning into distinct processes. The challenge in integrating memory stems from determining what to store in a perpetual learning system and when to enlist memory versus when to rely on more reactive or reflexive approaches. Another challenge is how to retrieve the best goal templates for instantiating goals with respect to the long-term and working memories; we plan to leverage *priming* defined by Hiatt & Trafton (2015) for retrieval and ranking.

ACTORSIM provides a starting point for implementing a perpetual learner in the two domains mentioned above. We plan to implement the cognitive architecture shown in Figure 3 and augment a baseline agent with a self-directed agenda for curricula learning. Tasks will vary for each environment and will be encoded as Goal-Task Networks. We will hand-code the initial networks while exploring ways to learn them from scratch and ablate provided or learned networks to assess their impact on performance. For curricula learning, we will extend insights from NELL, the Never-Ending Language Learner, which is a state-of-the-art, semi-supervised, continuously learning agent that built a word ontology of thousands of concepts by passively reading web pages over 10 years (Mitchell et al. 2015). The insights from NELL include: (1) coupling the training of inter-related learning tasks; (2) allowing the agent to learn new inter-task couplings; (3) allowing the agent's representation to grow; and (4) graduating the agent through progressively harder tasks as its competency increases. We will focus on three research questions of understanding under conditions does perpetual learning improve an agent's (1) adaptation to a new task; (2) responsiveness to commanding from or cooperation with simulated teammates; and (3) long-term performance over a system that lacks perpetual learning.

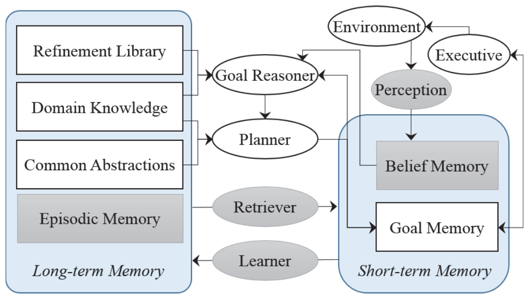


Figure 3: Integrating a cognitive architecture into ACTORSIM. Boxes indicate data, ovals indicate processes, and gray indicates new components.

## 5 Robotic Safety During Concurrent Execution

*Cross-disciplinary focus areas: Robotic systems, safety, planning and execution*

One area of research that has been identified as a challenge area by the Office of the Secretary of Defense (OSD)'s Autonomy Community of Interest is combining autonomy skills developed by different researchers, and possibly in different disciplines. We envision a paradigm for combining autonomy skills in which multiple, different behaviors can be active at the same time and their actions can be combined at various levels of abstraction in the ACTORSIM architecture; e.g., combining similar trajectories into a composite path, or interleaving atomic actions reach its goal faster.

There are at least two questions to answer in order to accomplish this, which traditionally lie in different disciplines. The first is how to support goal reasoning and planning for a domain where multiple actions can be executed simultaneously? This research question is distinct from literature on single- and multi-agent planning as well as literature on goal reasoning because of the interdependence of the physical robot resources that the behaviors utilize. A second question is how to ensure that concurrently executed skills are being executed safely? The way in which these active behaviors interact can be constrained both by the behaviors themselves, as well as by the overall architecture. For example a humanoid robot could combine a grasping behavior which moves the robot's left hand with speech processing behavior but could prohibit the combination of grasping with the robot moving to a new location.

These two questions are coupled, so solving them gracefully requires a cross-disciplinary approach. Researchers investigating goal reasoning and planning for this paradigm must model the behavioral constraints as well as include any abstract programming interfaces that are put in place to ensure safety. Likewise, work on the concurrent execution must be aware of the goals and plans that may arise when planning for this domain and account for them. In addition to ACTORSIM being well-suited to each of these challenges, it is also an appropriate platform for researching their combination because of its support for modeling activities at all levels of abstraction, from abstract goals to simulated (or real) execution.

Our technical approach will extend the cognitive architecture from Figure 3 with constraint-based planning (e.g., (Frank and Jonsson 2003; Jonsson et al. 2000)) that represents durative actions as tokens on a timeline with constraints existing between the start and end of those tokens. A plan is a partially ordered sequence of such tokens, and is consistent when it has no conflicting constraints. ACTORSIM will be modified to incorporate such constraints into goal refinement and leverage constraints from other components in the system. Robotic safety is thus ensured by allowing only consistent plans.

## 6 Rebel Agents

*Cross-disciplinary focus areas: Robotic systems, narratology, psychology, sociology, cognitive science.*

Rebel Agents (Coman, Gillespie, and Muñoz-Avila, 2015) are goal-reasoning agents capable of refusing an externally-assigned goal (or a course of action associated with that goal) that conflicts with their own internal motivation (which can be based on various factors, such as models of memory, emotion, social relationships, etc.). Our research objectives and intended outcomes pertaining to Rebel Agents include establishing a framework for AI agent rebellion informed by social and personality psychology, investigating potential benefits and challenges pertaining to Rebel Agents in various application domains (including human-machine interaction and interactive storytelling), and implementing Rebel Agents driven by various motivation models. Rebel Agents were initially proposed in an interactive storytelling context, for the purpose of enhancing character believability as well as providing a source of conflict, a key aspect of narrative in any medium. Decentralized autonomy consists of allowing agents playing characters in an interactive narrative to largely self-determine their behavior (Evans and Short 2014). Goal reasoning holds promise for narrative intelligence. Samsonovich & Aha (2015) propose a model of character reasoning in terms of actors and characters, but this model has not yet been translated into a digital-narrative implementation.

While decentralized autonomy can alleviate the content generation burden by not requiring fully scripted or closely guided character behavior, it also makes it more difcult to maintain the narrative coherence of the story. A radically different alternative is the strong-story-type system, in which all story decisions are made by a drama manager. There are approaches covering the middle ground between the two, and this is where Rebel Agents would fit in: a drama manager oversees the story, but the agents can rebel against the drama manager's decisions when reasoning that they could handle their own character arcs in more engaging ways.

This application of the Rebel Agent model to enhancing character arcs in narrative-intelligence contexts would allow us to explore the use of ACTORSIM for interactive-storytelling tasks. We are also interested in trying to connect ACTORSIM with story-centric interactive fiction environments which are fully or partially text-based, similar to work by Sharma et al. (2010).

In addition to interactive narrative, autonomous agents could express protest regarding assigned tasks for reasons including ethics, safety, concern for their self-development, or teaming considerations. Our planned cross-disciplinary approach, which draws on psychology and sociology, will take into account the fact that, when it comes to human behavior, refusal, reluctance, protest, rebellion, and similar attitudes occur in varied situations and manifest in varied ways. A task can be refused immediately after it is assigned; a general process for this in human-robot interaction is proposed by Briggs and Scheutz (2015). However, a person might also start questioning the assigned task during execution because its implications become clear, because something changes in the environment, thus affecting the task's implications, or because the motivation of the person conducting the task has changed in ways affecting how they view the task. Humans are often proactive in expressing their protest regarding specific tasks they are expected to conduct as well as the general context of their activities.

Here is one example of rebellion based on a self-development motivation model. In personality psychology, the strengths-based leadership approach (Rath and Conchie 2009) argues that every person, whether a leader or not, should be offered the opportunity to routinely conduct activities in line with their strengths. We envision a Rebel Agent that can assess whether its currently assigned task plays up to its strengths, and express protest if that is not the case. Self-monitoring/assessment and, possibly, outside feedback, would be used by the agent not only to improve its performance, but also to maintain awareness of how well it performs certain tasks relative to others. Outside feedback and self-monitoring can be used to improve the agents ability to conduct various tasks over time. However, for reasons either endogenous or exogenous to the agent, certain abilities can plateau at an average level, while others are developed into exceptional skills (the agents strengths). The agent would be aware of these strengths and proactive in ”selling them”. Alternatively, the agent could proactively request tasks that provide it with better learning opportunities.

We will explore ways in which the goal reasoning within ACTORSIM can accommodate such processes pertaining to rebellion. To accomplish this we intend to draw on the psychology and sociology literature to identify a wide range of situations reflecting human rebellion and related attitudes and search for ways in which these could be applicable to AI agent autonomy in useful ways.

## 7 Summary

Cross-disciplinary approaches are critical to solving some of the challenges faced within autonomous systems research and development. We discussed specific challenges with respect to a toolkit called ACTORSIM. To date, much of the development effort on ACTORSIM has focused within sub-disciplines of AI. We identified new directions that will broaden ACTORSIM to include disciplines outside of AI.

## Acknowledgments

This research was funded by OSD and NRL. We thank Brandon Enochs at NRL for insightful discussions on the robotic safety project. We also thank anonymous reviewers whose comments improved the paper.

## References

Abel, D.; Hershkowitz, D. E.; Barth-Maron, G.; Brawner, S.; OFarrell, K.; MacGlashan, J.; and Tellex, S. 2015. Goal-based action priors. In *Proc. ICAPS*.

Alford, R.; Shivashankar, V.; Roberts, M.; Frank, J.; and Aha, D. W. 2016. Hierarchical planning: Relating task and goal decomposition with task sharing. In *Proc. IJCAI*. AAAI Press.

Anderson, J. R., and Lebiere, C. 1998. *The atomic components of thought*. Erlbaum.

Bonanno, D.; Roberts, M.; Smith, L.; and Aha, D. 2016. Selecting subgoals using deep learning in minecraft: A preliminary report. In *Deep Learning for Artificial Intelligence: Papers from the IJCAI Workshops*.

Briggs, G., and Scheutz, M. 2015. ”Sorry, I cant do that”: Developing mechanisms to appropriately reject directives in human-robot interactions. In *AAAI Fall Symposium Series*.

Coman, A.; Gillespie, K.; and Muñoz Avila, H. 2015. Case-based local and global percept processing for rebel agents. In Kendall-Morwick, J., ed., *ICCBR (Workshops)*, volume 1520 of *Proc. of the CEUR Workshop*, 23–32. CEUR-WS.org.

Evans, R., and Short, E. 2014. Versu - a simulationist storytelling system. *IEEE Trans. Comput. Intellig. and AI in Games* 6(2):113–130.

Fox, M., and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *JAIR*. 28:236–297.

Frank, J., and Jonsson, A. 2003. Constraint-based attribute and interval planning. *Journal of Constraints* 8(4):339–364.

Ghallab, M.; Nau, D.; and Traverso, P. 2014. The actor's view of automated planning and acting: a position paper. *Artificial Intelligence* 208:1–17.

Ghallab, M.; Nau, D.; and Traverso, P. 2016. *Automated Planning and Acting*. Cambridge University Press.

Hiatt, L. M., and Trafton, J. G. 2015. An activation-based model of routine sequence errors. In *Proc. of the International Conference on Cognitive Modeling*.

Isele, D.; Rostami, M.; and Eaton, E. 2016. Using task features for zero-shot knowledge transfer in lifelong learning. In *Proc. of IJCAI*, 1620–1626. AAAI Press.

Johnson, B.; Roberts, M.; Apker, T.; and Aha, D. 2016. Goal reasoning with information measures. In *Proc. ACS*.

Jonsson, A.; Morris, P.; Muscettola, N.; Rajan, K.; and Smith., B. 2000. Planning in interplanetary space: Theory and practice. In *Proceedings of AIPS 2000*, 177–186.

Kope, A.; Rose, C.; and Katchabaw, M. 2013. Modeling autobiographical memory for believable agents. In Sukthankar, G., and Horswill, I., eds., *Proc. AIIDE*. AAAI.

Laird, J. 2012. *The Soar Cognitive Architecture*. MIT Press.

Langley, P., and Choi, D. 2006. A unified cognitive architecture for physical agents. In *Proc. AAAI*. AAAI Press.

Luke, S.; Cioffi-Revilla, C.; Panait, L.; Sullivan, K.; and Balan, G. 2005. MASON: A multi-agent simulation environment. In *Simulation: Trans. of the soc. for Modeling and Simulation International*, volume 82(7), 517–527.

Menager, D., and Choi, D. 2016. A robust implementation of episodic memory for a cognitive architecture. In *Proc. of Annual Meeting of the Cognitive Science Society*.

Mitchell, Tom *et al.* 2015. Never-ending learning. In *Proc. AAAI*.

Rath, T., and Conchie, B. 2009. Strengths-based leadership: Great leaders, teams, and why people follow. *Gallup Press*.

Ring, M. B. 1997. Child: A first step towards continual learning. *Machine Learning* 28(1):77–104.

Roberts, M.; Apker, T.; Johnston, B.; Auslander, B.; Wellman, B.; and Aha, D. W. 2015. Coordinating robot teams for disaster relief. In *Proc. FLAIRS*.

Roberts, M.; Alford, R.; Shivashankar, V.; Leece, M.; Gupta, S.; and Aha, D. 2016. Goal reasoning, planning, and acting with ActorSim, the Actor Simulator. In *Poster Proc. ACS*.

Samsonovich, A. V., and Aha, D. W. 2013. Character-oriented narrative goal reasoning in autonomous actors. In *Goal Reasoning: Papers from the ACS Workshop (Technical Report CS-TR-5029). College Park, MD: University of Maryland, Department of Computer Science*, 166–181.

Sharma, M.; Ontan, S.; Mehta, M.; and Ram, A. 2010. Drama management and player modeling for interactive fiction games. *Computational Intelligence* 26(2):183–211.

Thrun, S., and Mitchell, T. M. 1995. Lifelong robot learning. *Robotics and Autonomous Systems* 15:15–46.

Trafton, G.; Hiatt, L.; Harrison, A.; Tamborello, F.; Khemlani, S.; and Schultz, A. 2013. ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction* 2(1):30–54.