Hierarchical Approach to Transfer of Control in Semi-Autonomous Systems

(Extended Abstract)

Kyle Hollins Wray Luis Pineda Shlomo Zilberstein
College of Information and Computer Sciences
University of Massachusetts Amherst
{wray,lpineda,shlomo}@cs.umass.edu

ABSTRACT

Semi-Autonomous Systems (SAS) describe a class of systems that are explicitly controlled by both human and agent actors, incorporating the distinct capabilities of each. They are designed for domains that require quick and safe transfer of control between the agent and human, and embed control transfer decisions in the overall high-level plan. We formally define SAS as a hierarchical model and its properties. We discuss how micro-level transfer of control and macro-level path planning can be solved together. Finally, we explore how SAS can be applied to semi-autonomous vehicles.

1. INTRODUCTION

Purely autonomous systems have been applied to a wide range of domains ranging from water reservoir control [3] to energy-conserving smart environments [7]. Almost all applications, however, require human intervention as part of their standard operating procedure. Few models actively incorporate this collaboration, and instead use hard-coded default behaviors [2]. Semi-Autonomous Systems (SAS) are models that explicitly model the collaboration of actors, in order to proactively utilize their respective capabilities [11].

New challenges arise in semi-autonomous systems due to the inherent uncertainty and complexity of human behavior. We consider semi-autonomous driving [10] as our target application. Within this domain, vehicles can only operate autonomous on a subset of the roads (e.g., only well-mapped roads). For longer routes, the vehicle requires the human to occasionally take control during execution. This transfer of control process requires second to second monitoring as the vehicle messages the driver to change the vehicle's controlling entity. It may be unsuccessful due to the state of the human (e.g., distracted) or simply time limitations.

The proposed collaborative multiagent framework is quite distinct from other models such as Shared Plans [4], Teamwork [9], and Dec-POMDPs [1]. First, SAS requires exactly one actor in control at a time. Second, transfer of control must be explicitly modeled. Finally, SAS must proactively leverage each actor's capabilities (or lack thereof) as it efficiently moves through the sate space.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*
Copyright © 2016, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2. SEMI-AUTONOMOUS SYSTEMS

Semi-Autonomous Systems (SAS) rely on collaboration between a human and an agent in order to achieve some goals while maintaining a measure of safety [11]. We consider semi-autonomy within the context of automated planning, extending a Markov Decision Process (MDP) to support semi-autonomy, as formally defined below.

DEFINITION 1. *A **semi-autonomous system** is represented by a tuple $\langle \mathcal{A}, S_+, A_+, T_+, C_+, G, L \rangle$.*

- *$\mathcal{A}$ is a set of actors (controlling entities).*
- *$S_+ = S \times \mathcal{A}$ is a set of factored states: a standard state set S and the current controlling actor $\mathcal{A}$.*
- *$A_+ = A \times \mathcal{A}$ is a set of factored actions: a standard action set A and the next desired actor $\mathcal{A}$.*
- *$T_+ : S_+ \times A_+ \to \triangle^{|S_+|}$ is a transition function, comprised of a state transition $T_\mathbf{a} : S \times A \to \triangle^{|S|}$ for each actor $\mathbf{a} \in \mathcal{A}$, and control transfer function $\rho : S_+ \times \mathcal{A} \to \triangle^{|\mathcal{A}|}$.*
- *$C_+ : S_+ \times A_+ \to \mathbb{R}^+$ is a cost function.*
- *$G \subseteq S_+$ is a set of goal states.*
- *$L \subseteq S_+$ is a set of live states, such that for actor capability function $\psi : S \to 2^{\mathcal{A}}$, $L = \{\langle s, \mathbf{a} \rangle | \mathbf{a} \in \psi(s)\}$.*

The actors $\mathcal{A}$ of the system describe controlling entities, which include at a minimum a human λ and an autonomous agent ν; we focus in this paper on situations involving these specific two actors. The states must record who is in control at any given time, and the actions must record intentions to switch control to new actors. In SAS, we cannot always assume that transfer of the control has a flawless execution. Hence our T_+ is factored into two components: $T_\mathbf{a}$ and ρ.

The first component, an **actor state transition function** denoted as $T_\mathbf{a} : S \times A \to \triangle^{|S|}$, describes how an actor $\mathbf{a} \in \mathcal{A}$ can operate in the world when in control (n-simplex $\triangle^n$). The second component, a **control transfer function** denoted as $\rho : S_+ \times \mathcal{A} \to \triangle^{|\mathcal{A}|}$, describes the result of attempting to transfer control from the current actor in a given state. Combining these two functions, we define the **SAS state transition function** T_+ for any state $s_+ = \langle s, \mathbf{a} \rangle$, action $a_+ = \langle a, \hat{\mathbf{a}} \rangle$, and successor $s'_+ = \langle s', \mathbf{a}' \rangle$:

$$T_+(s_+, a_+, s'_+) = \begin{cases} T_\mathbf{a}(s, a, s'), & \text{if } \mathbf{a}{=}\hat{\mathbf{a}}{=}\mathbf{a}' \\ T_\mathbf{a}(s, a, s')\rho(s_+, \hat{\mathbf{a}}, \mathbf{a}'), & \text{if } \mathbf{a}{\neq}\hat{\mathbf{a}} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In Equation 1, the first component corresponds to keeping the current actor, which simply follows the actor's state transition. The second component describes the actor still in control but seeking to switch to a different actor at the next state. Finally, the third component indicates that it is impossible to take away control from an actor without the desire to transfer control. The separation of macro-level planning and micro-level transfer of control denotes is hierarchical approach, as a type of *option* [8].

The **actor capability function** ψ defines which actors can act in each state. In the semi-autonomous driving domain, $\psi(s)$ always contains the human λ, but only includes the vehicle ν on a subset of the roads on which it can operate. This function induces the set of **live states** L (Definition 1). Live states describe states in which the system is *alive* or *safe*. We impose **live state constraints** that guarantee non-live states are unable to reach any goal state, also called *dead ends* [6]. We characterize both *policies* and *systems* based on their ability to maintain live state. Specifically, if a policy guarantees live state, then we call it **strong**; otherwise, it is **weak**. A SAS is called **strong** if its optimal policy is strong, and **weak** otherwise. In practice, this requires that we prove the transfer of control ρ is well-behaved.

3. DESIGN AND APPLICATION

Designing semi-autonomous systems first requires creating the transfer of control process, then defining the SAS itself.

3.1 Transfer of Control

Transfer of Control (TOC) is the critical method that enables effective and safe transference of the controlling entity in the system. TOC considers the sequences of messages to the human (or other actors) that preserves the human's amiable perception of the agent (e.g., aggregated annoyance) while still effectively conveying the desired intentions. We assume this process can always be aborted safely. For example, in semi-autonomous driving, the vehicle will pull over to the side of the road.

This process can be modeled as a POMDP [5]. States encode time remaining, information regarding the last message sent, and a notion of the human state. Actions are the TOC messages and an abort action. Observations encode noisy sensors within the SAS that monitor the human. The state transition and observation functions capture the uncertainty regarding the human's response to messages.

3.2 Semi-Autonomous Vehicles

The application of our TOC model produces a ρ, based on its performance in transferring control in various scenarios. A SAS for Semi-Autonomous VEhicles (SAVE) can be created using a weighted directed graph of roads with start and end vertexes. Each vertex corresponds to an intersection. Edge weights denote the time spent on each road following the speed limit. This SAVE graph is represented as the standard states S in the SAS. The standard actions A are directions to take at intersections. The state transitions $T_{\mathbf{a}}$ follow from the graph, only entering dead ends when the vehicle ν enters a road in which it cannot drive.

The transfer of control process can be used to effectively notify the user with ever-increasing persistence (Figure 1). This transfer process is captured in the SAVE transition, using pre-solved instances of the TOC POMDP model as a kind of option, thus behaving as a hierarchical model.

Figure 1: Semi-autonomous vehicle simulator: System requests to transfer control to a busy driver.

4. CONCLUSION

We present a new multiagent model for semi-autonomous systems. It enables explicit modeling of agents (actor), including humans, each with their own capabilities. We show how a POMDP can be used to solve the transfer of control problem, enabling a safe change of the controlling entity. Finally, we state how semi-autonomous vehicles may be modeled as a SAS.

Acknowledgments. This work was supported in part by the National Science Foundation grant number IIS-1405550.

REFERENCES

[1] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[2] J. Biswas and M. M. Veloso. Localization and navigation of the CoBots over long-term deployments. *International Journal of Robotics Research (IJRR)*, 32(14):1679–1694, 2013.

[3] A. Castelletti, F. Pianosi, and R. Soncini-Sessa. Water reservoir control under economic, social and environmental constraints. *Automatica*, 44(6):1595–1607, 2008.

[4] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[5] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998.

[6] A. Kolobov, Mausam, and D. Weld. A theory of goal-oriented MDPs with dead ends. In *Proceedings of the 28th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 438–447, Corvallis, Oregon, August 2012.

[7] J.-Y. Kwak, P. Varakantham, R. Maheswaran, M. Tambe, F. Jazizadeh, G. Kavulya, L. Klein, B. Becerik-Gerber, T. Hayes, and W. Wood. SAVES: A sustainable multiagent application to conserve building energy considering occupants. In *Proccedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 21–28, Valencia, Spain, 2012.

[8] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[9] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 2:83–124, 1997.

[10] K. H. Wray and S. Zilberstein. Multi-objective POMDPs with lexicographic reward preferences. In *Proceedings of the 24th International Joint Conference of Artificial Intelligence (IJCAI)*, pages 1719–1725, July 2015.

[11] S. Zilberstein. Building strong semi-autonomous systems. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, pages 4088–4092, Austin, Texas, 2015.