charles river analytics

625 Mount Auburn Street · Cambridge, MA 02138 · 617.491.3474 · www.cra.com

Prescient, Socially Intelligent Coach (PSI-Coach, ASIST)

Dr. Bryan Loyall, Dr. Avi Pfeffer, Dr. James Niehaus, Rob Hyland, Curt Wu, Kenny Lu, Dr. Peter Weyhrauch

Charles River Analytics, 625 Mount Auburn Street, Cambridge MA 02138

End of Phase Report (also inclusive of 2022 R&D Status Reports) 28 April 2023

Prepared for: Joshua Elliot, (703) 526-2873, Joshua.elliot@darpa.mil
Award number: HR0011-19-C-0126
Period of performance: 01 November 2019 – 28 April 2023
CLIN: 0001
Deliverable ID: R1911120

REPORT DOCUMENTATION PAGE	*Form Approved* OMB No. 0704-0188

1. REPORT DATE *(MMMM YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
April 2023	BAA Final Report	01 November 2019 – 28 April 2023

4. TITLE AND SUBTITLE	
Prescient, Socially Intelligent Coach (PSI-Coach)	**5a. CONTRACT NUMBER** HR0011-19-C-0126
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Bryan Loyall, Dr. Avi Pfeffer, Dr. James Niehaus, Rob Hyland, Curt Wu, Kenny Lu, Dr. Peter Weyrauch, Dr. Alex Gee	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Charles River Analytics Inc. 625 Mount Auburn Street Cambridge, MA 02138	R1911122
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Army Research Office Nicole Fox 800 Park Office Drive, Suite 4229 Research Triangle Park, NC 27709	**10. SPONSOR/MONITOR'S ACRONYM(S)** DARPA-I2O **11. SPONSORING/MONITORING AGENCY REPORT NUMBER**

12. DISTRIBUTION AVAILABILITY STATEMENT
Distribution Statement A: Approved for public release. Distribution is unlimited.

13. SUPPLEMENTARY NOTES
This is an BAA report. Report contains color.

14. ABSTRACT
Improving team performance requires a deep understanding of what a team is doing—whether those actions are rational, irrational, or idiosyncratic—and administering effective, non-annoying interventions so the team can reach high levels of achievement. Recognizing detailed user goals, mental states, and behaviors (in all of their human complexity) from low-level actions in dynamic open worlds is a challenging task requiring real time inference of complex, human cognitive processes. Predicting and planning well-timed, effective interventions takes a human coach years to learn and has repeatedly failed (in sometimes spectacular ways) in research and commercial AI systems. To meet these challenges, we created a Prescient, Socially Intelligent Coach (PSI-Coach). PSI-Coach is designed to unobtrusively monitor each team member to: (1) recognize their goals, mental states, and behaviors—without the assumption of rationality—by combining probabilistic programming inference with a cognitive architecture optimized to capture human variation (this architecture has modeled the minds of 4th-gen fighter pilots, missile defense operators, medical team members, and many others over the last 30 years); (2) infer detailed plans and goals (even when people do multiple things at the same time or change and adapt their behavior) using algorithms that reverse-engineer goals and mental states from dynamic streams of actions; (3) recognize shared mental models and whether they are in alignment or skewed using joint-behavior inference and analysis; (4) analyze these goals, mental states, behaviors, and shared mental models to compute practical, real-time team performance indicators based on empirical results of effective team performance collected over 10 years on more than 1,200 teams; and (5) use all of this information to predict team performance and plan effective, strategically timed interventions that maximize team performance. PSI-Coach makes these predictions and interventions using a planning technology that reasons about effectiveness, timing, and disruption of potential interventions. This technology has delivered preliminary results, unobtrusively guiding interactive stories with unpredictable participants.

15. SUBJECT TERMS
BAA report, AI coach, mental models, probabilistic programming, cognitive architecture, team performance, interventions, interactive stories

16. SECURITY CLASSIFICATION OF: Unclassified			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	113113	Dr. Bryan Loyall
U	UU	U			**19b. TELEPHONE NUMBER** (*Include area code*) (617) 234-1513

Acknowledgments

This work was performed under DARPA-I2O contract number HR0011-19-C-0126. The authors thank Joshua Elliot, Brian Sandberg, Tanya Tanner, and Jennifer Mack for their support and direction on this project.

We would also like to thank our extended project team, who were essential to making PSI-Coach (ASIST) a successful effort. Dr. Bryan Loyall served as the Principal Investigator of the effort, contributing to the overall research efforts and design and development of the solution as well as managing many day-to-day operations. Dr. Avi Pfeffer, Chief Scientist at Charles River Analytics, served as Cognitive Inverter Lead, supported by lead developer Kenny Lu. Dr. James Niehaus, Strategic Coach Lead, contributed his expertise in interactive narrative, cognitive modeling, and planning systems. Mr. Curt Wu led software development, adapting to changing needs as the project evolved. Mr. Rob Hyland served as Program Manager, driving the team successfully to fulfil experimental and program milestones.

Table of Contents

List of Figures

List of Tables

List of Symbols, Abbreviations, and Acronyms

Acronym	Definition
AGLOW	ASI for Group Learning and Optimization of Collaborative Workflows
AI	Artificial intelligence
ASI	Artificial social intelligence
AU	Action units
CADE	Cyber Adversary Discovery Engine
CMU	Carnegie Mellon University
COS	Center for Open Science
CV	Critical victims
DBN	Dynamic Bayesian Network
DCO	Defensive cyberspace operations
DSL	Domain specific language
FACS	Facial action coding system
FOV	Field of view
GLOW	Group Learning and Optimization of Collaborative Workflows
HSR	Human Subject Research
ICV&E	Interactive Cyber Visualization & Exploration
MSS	Mariner Skills Suite
MToT	Machine Theory of Teams
NLP	Natural language processing
NSST	Navigation Seamanship Shiphandling Trainer
PP	Probabilistic programming
PPAML	Probabilistic Programming for Advanced Machine Learning
PPL	Probabilistic Programming Language
PSI-Coach	Prescient, Socially Intelligent Coach
SAR	Search and rescue
SBIR	Small Business Innovative Research
SWO	Surface Warfare Officers
TCI	Test of Collective Intelligence
ToM	Theory of Mind
TToM	Team Theory of Mind
TTP	Tactics, techniques, and procedures
UNR	University of Nevada Reno
USAR	Urban Search and Rescue

1. Introduction

Improving team performance requires deeply understanding what a team is doing—whether rational, irrational, or idiosyncratic—and administering effective, non-annoying, interventions so the team can reach high levels of achievement. Recognizing detailed user goals, mental states, and behaviors (in all of their human complexity) from low-level actions in dynamic open worlds is a challenging task requiring real time inference of complex, human cognitive processes. Predicting and planning well-timed, effective interventions requires a depth of understanding that takes human coaches years to learn, and has repeatedly failed (in sometimes spectacular ways) in research and commercial AI systems.

To meet these challenges, we created a Prescient, Socially Intelligent Coach (PSI-Coach). PSI-Coach is designed to unobtrusively monitor each team member to: (1) recognize their goals, mental states, and behaviors—without the assumption of rationality—by combining probabilistic programming inference with a cognitive architecture optimized to capture human variation; (2) infer detailed plans and goals—even when people do multiple things at the same time, or change and adapt their behavior—using algorithms that reverse-engineer goals and mental states from dynamic streams of actions; (3) recognize shared mental models and whether they are in alignment or skewed using joint-behavior inference and analysis; (4) analyze these goals, mental states, behaviors, and shared mental models to compute practical, real-time team performance indicators; and (5) use all of this information to predict team performance and plan effective, strategically timed interventions that maximize team performance. PSI-Coach makes these predictions and interventions using a planning technology that reasons about effectiveness, timing, and disruption of potential interventions. This technology has delivered preliminary results, unobtrusively guiding interactive stories with unpredictable participants.

PSI-Coach's artificial social intelligence (ASI) agent framework employs methods to decide whether to intervene immediately or wait until a more appropriate time. The method is based on automatically inferring team Theory of Mind (TToM) and Machine Theory of Teams (MToT). PSI-Coach has demonstrated increased agreement with inferences made by human coaches as well as their choices and timing on when to intervene. PSI-Coach demonstrated the ability to automatically identify team process problems unique to each team and their situation dynamics. The PSI-Coach system provided tailored intervention content with optimal timing that improved team processes within 60 seconds. Figure 1, below, illustrates one such demonstration where a Medic avoids a repeated team process mistake as a result of a well-timed PSI-Coach intervention.

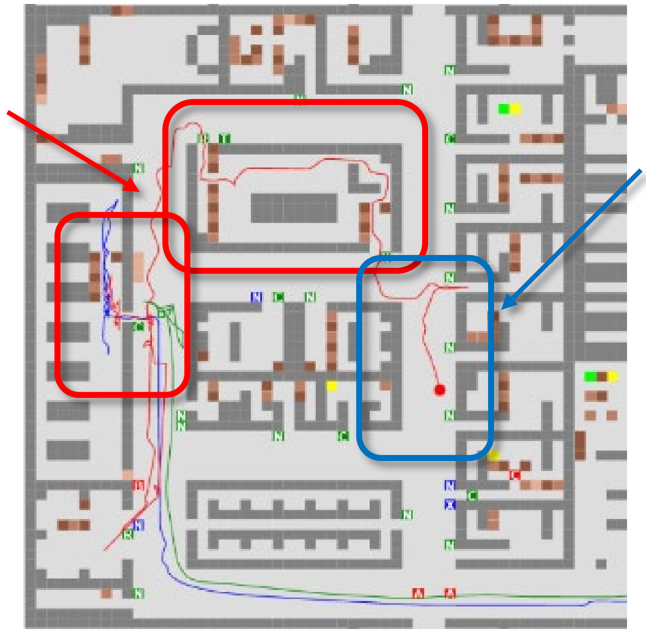


Figure 1: Demonstration of Medic decision-making before and after intervention from PSI-Coach. *Medic (thin red line) is ignoring team communication within the red boxes, entering rooms that have been communicated as clear. PSI-Coach issues intervention at the start of the blue box when Medic is about to ignore team communication again. Medic then avoids repeating mistake, starts reliably following better team process, and uses team communication more consistently.*

1.1 Task Objectives

PSI-Coach includes five key innovations that are organized around the two core ASIST goals: deeply understand what the team is doing and provide effective, non-annoying interventions to improve team performance. With these goals in mind, we developed PSI-Coach to achieve the following:

(1) Recognize goals, mental states, and behaviors without the assumption of rationality by extending an expressive AI cognitive architecture with probabilistic programming languages (PPLs). One of the foundational research challenges of ASIST is to understand the behaviors people exhibit, with all of their natural human variations. The idealistic assumption of rationality that pervades many artificial intelligence (AI) systems, cognitive theories, and team theories greatly restricts that understanding. PSI-Coach overcomes this weakness with an innovative *Cognition Inverter (Individual)* system that extends decades of work in expressive AI cognitive architectures for generating natural human behavior with recent breakthroughs in PPLs (Pfeffer, 2016) that allow generative models to be inverted for recognition. For example, it has long been believed that computer vision might be performed by inverting the computer graphics pipeline (i.e., “running it backwards”), but this concept only became practical recently with breakthroughs in PPLs (Kulkarni, Kohli, Tenenbaum, & Mansinghka, 2015). This general idea has subsequently been broadly applied, with examples in inverting physics simulators (C. Bates, Battaglia, Yildirim, & Tenenbaum, 2015) and seismic models (Arora, et al, 2013); this last breakthrough system was adopted by the United Nations for nuclear test ban monitoring and reduced missed events of interest by 60%. For PSI-Coach’s Cognition Inverter, we applied this emerging technique to infer the goals, mental states, and behaviors of team members by inverting an expressive cognitive model. We built on a complex generative cognitive model, Hap, which was designed to capture natural human behavior (Sliva et al., 2016; Loyall et al., 2004; Loyall, 1997). Hap supports personality expression, individual differences, and social intelligence (see Section 3.2.1) and has spawned six research and commercial variants, which have been used as a foundation for dozens of research projects in socially engaging interactive agents (e.g., at Carnegie Mellon, Stanford, Georgia Tech, University of California Santa Cruz, and several research labs). Versions of Hap have been used for multiple commercial products and as a foundation for over a dozen Ph.D. theses. Hap is currently on its sixth version, and AI agents based on Hap have been created for a diverse range of domains (e.g., modeling medical teams, squad-level AI behaviors, characters in interactive stories, fourth generation combat pilots, cyberattackers, and distractible missile defense operators). By using Hap for recognition, the Cognition Inverter will be able to recognize natural human behavior as a key foundation for assisting team performance.

(2) Robustly recognize details of dynamic open world behavior using inference over reactive cognitive architecture language features. After recognizing natural human behavior (whether rational, irrational, or idiosyncratic), the next ASIST research challenge was to recognize behavior with the dynamic complexity that people exhibit in open worlds. People pursue multiple goals simultaneously, switch between goals, change strategies, and adapt behaviors to changes in the world. Current technologies cannot capture this dynamic complexity. Plan recognition systems recognize a user's specific stage in a rigid plan but fail in the complexity of an open world. Activity recognition systems (e.g., those using deep learning, graphical Bayesian models, or rule-based systems) recognize coarse-grained activities within dynamic worlds but fail to provide the details necessary for effective assistance, such as where in the plan the individual is stuck. For PSI-Coach, we address this research question by extending the Cognition Inverter's inference capabilities for these dynamic changes. We built on architectural properties of Hap that generate these dynamic changes, including automatic support for parallel goals, dynamic switching between goals, and reactivity to changes in the world (Bates, Loyall, & Reilly, 1992). We extended these architectural elements with our PPL technology—developed under DARPA PAL and PPAML—to enable the large-scale inference of these reactive properties (Sections 3.2.1 and 3.4). Early results applying these techniques to the related, but simpler problem of recognizing dynamic workflows under the DARPA PPAML program showed the ability to recognize these dynamic properties with 99% or better accuracy. By generalizing these approaches to the complexity of Hap behavior types and dynamic capabilities, the Cognition Inverter combines recognition of specific details of individual goals and behaviors with the precision of plan recognition and the robustness to dynamics of activity recognition systems.

(3) Recognize aligned and skewed shared mental models within teams using joint behavior inference. A foundation of team effectiveness is the extent to which the team is working together, with aligned shared mental models. PSI-Coach extends the Cognition Inverter system with representations and inference for joint behaviors, first inferring joint behaviors individually, per team member, then performing analysis across the set of recognized hypotheses. By analyzing across the team, Cognition Inverter can recognize where the team's shared mental models are aligned and where they have skewed (Section 3.4), providing uniquely valuable insights for measuring team performance and planning effective interventions.

(4) Measure team performance indicators in real time using inferred mental states combined with novel, dynamic-task extensions to our partner's test of collective intelligence. The best human coaches of high-performing teams can identify indicators of effective team behavior and factors causing poor team performance. PSI-Coach's *Cognition Inverter (Team)* system provides this capability by computing real-time measurements of team performance that contextually score team contributions, based on empirical results of effective team performance collected over 10 years on more than 1,200 teams by our teammates in the Carnegie Mellon University (CMU) Collective Intelligence Lab. PSI-Coach computes team performance from behavioral indicators that map to the well-established Test of Collective Intelligence (TCI), including generally applicable measures of memory, attention, and motivation. PSI-Coach expands TCI to compute real-time scores for open environment tasks and calculates team performance scores based on the stream of inferred mental states from the Cognition Inverter (Section 3.4.2), then uses the results to inform effective intervention planning.

(5) Maximize team performance using experience management algorithms that predict team behaviors and reason about both the efficacy and disruption of interventions. Once it understands team mental states and performance indicators, PSI-Coach predicts possible futures and strategically plans interventions to maximize the team's performance. Intervention timing is just as important as intervention content (Hackman & Wageman, 2005), and good human coaches draw on Theory of Mind to optimize both. Current state-of-the-art, machine-based interventions are blunt in comparison; they show little social intelligence and typically intervene often throughout the task, creating a net effect of lower efficacy and sometimes annoying participants. For PSI-Coach, we developed a *Strategic Coach* (Section 3.5). that reasons about intervention timing, probable disruption, and the projected impact on team performance. The coach extends Monte Carlo Tree Search (MCTS) to project team futures and reasons about the timing and choice of interventions. Selection is guided by measures of intervention efficacy, need, and degree of disruption. This approach builds on decades of AI research, including foundational and recent work by our team to steer interactive agents, dramas, and stories toward satisfying and impactful experiences while minimizing the feeling of intrusion (Weyhrauch, 1997; Mott & Lester, 2006; Riedl & Bulitko, 2012; Wang, Rowe, Min, Mott, & Lester, 2017; Robertson & Young, 2018). We will include computational measures of intrusion and intervention effectiveness (Kelso & Weyhrauch, 1994; Woolley et al., 2010; Woolley et al., 2015) so PSI-Coach interventions feel well-timed, tactful, helpful, and immediately useful, and dramatically improve end-user acceptance.

2. Technical Problems/Challenges

To develop a foundational AI agent architecture that employs machine theory of mind to effectively facilitate teamwork, PSI-Coach must develop several innovative technical capabilities that each exceed the limitations of today's approaches. Though the PSI-Coach research plan was ambitious, we structured the R&D to systematically manage risk and maximize successful results. We decomposed the major technical challenges into elements that could be tested and created an empirically driven research plan with progressive milestones to validate new performance achievements. The remainder of this section summarizes the key elements of the progressive research and risk mitigation plan we followed.

Crawl. We established early baselines that can operate on single human variations. We created an initial Cognition Inverter (Individual) that can recognize unobservable human mental states. We tested this initial capability in early in Phase 1 to ensure that it can recognize simple, interleaved tasks performed by a single person in our internal Hap test environment. This validated that the inference algorithm operates in real time, accounts for human behavior variation, and resolves ambiguous observations. We also established an initial Strategic Coach intervention planning capability next, which processes the Cognition Inverter inference stream, tested at the first program dry run. This validates that the Strategic Coach operates in real time, predicts human moves (including off-task behaviors), and plans a basic set of interventions (e.g., task hints, strategy changes, and information cues). Before Study 1 in Phase I, we integrated an initial human sensing capability to empirically validate indicators of human state and task performance. After Study during Phase I program evaluation, we validated the primary fundamental risks in the core PSI-Coach technical approach so we could reevaluate elements of our approach and prioritize internal validation development and tests. A summary of PSI-Coach technical risks and our mitigation plan is shown in Table 1, which guided research and validation priorities throughout the project.

Table 1: Technical risks and mitigation plans

Risk Type	Risk Source		Mitigation Techniques
	Likelihood	Impact	
Sample-based inference does not scale to the cognitive complexity needed during Phase I.	Medium	High	Transition to lazy factor graph inference earlier than planned, moving some Phase II work into Phase I.
Task complexity requires undue engineering time, reducing focus on core research.	Medium	High	Coordinate closely with TA3 and PM to focus TA3 tasks on core elements of ASIST research goals, and minimize undue engineering complexity.
Open NLP is more complex than the program intends, and risks taking resources intended for other purposes.	Medium	High	Restrict text-input and consider menu-based chat communications to focus on key ASIST goals.
Probabilistic hypotheses of goals, mental states, and behavior inference explode combinatorally.	High	High	Restrict the probabilistic inference to maintain the most likely hypotheses. Recognize low-likelihood interpretations as soon as new evidence appears.
MCTS state space is too large for effective intervention planning.	Medium	Medium	Apply previously developed abstraction techniques to MCTS search to improve its performance.

Walk. By Month 12, we established a complete Phase I PSI-Coach baseline that perform with similar coaching and timing as human experts, recognize human internal states (e.g., workload situational knowledge, policies, decisions, and goals—in part validated with human sensing algorithms) for TA3 environment tasks, adapt to perturbations, and generate non-intrusive interventions.

Run. We followed this pattern of iterative research and rigorous validation of PSI-Coach performance as we extended the technology to include team reasoning and facilitating for multiple humans (Phase II) doing complex tasks (Phase III). Early in Phase II, by Month 18, we established a team-enhanced PSI-Coach version, since much of the representation infrastructure was initially developed early in Phase I. This establishes basic joint behavior recognition, real-time team performance measures, and team interventions. We conducted limited pilot studies to generate training data and ensure PSI-Coach generalizes across different human participants, tasks, and perturbations. This allowed us to assess risks and mitigate issues before the Phase II dry runs, therefore ensuring PSI-Coach was optimized for each Phase II evaluation. For Phase III, we will scale up team reasoning and joint behavior processing to 10 human participants. We tested the generalization of our representation and increased PSI-Coach learning capacity to reduce manual model creation. We also tested a variety of intervention to validate human state and intervention efficacy. By month 44, we plan to have proven PSI-Coach as a team-cognition experimental platform. We have structured system development and data rights assertions to be open, so PSI-Coach can serve as a community-wide resource for future team-based, social-intelligence research. This enables us to apply PSI-Coach across domains, which we are exploring with a related effort to apply PSI-Coach to open-world collective problem-solving tasks such as in a real-world puzzle-hunt environment, a task that shares similarity to teamwork performed by Intelligence Fusion Cells

3. General Methodology

3.1 Meetings and Presentations

In addition to supporting the December 2019 Program Kickoff Meeting and holding monthly status meetings, we actively supported cross-program experimental design through meetings with DARPA, Carnegie Mellon University (CMU), and our other collaborators. We participated in the following meetings and presentations:

- Led Experimentation and Data & Measures Working Groups (December 2019–September 2020) and Hypothesis Working Group (September 2020–present).
- Collaborated with Gallup team on feasibility and approach for registration of model-based hypotheses, both on human performance and on agent capabilities.
- Defined pilot experiment protocol, in close collaboration with CMU-TA2, shared approach program-wide, informing TA3 August/September 2020 evaluation experiments. Shared system testing insights and suggestions.
- Led effort to identify and prioritize events that seem to be missing from master variable in collaboration with DOLL-TA1, UAZ-TA1, CMU-TA1, CMU-TA2, and Aptima/ASU-TA3.
- Defined pilot experiment protocol. Shifted to 100% virtual experiment, in close collaboration with CMU-TA2, and shared approach program-wide, now used by TA3. Shared system testing insights and suggestions.
- Iterated the Minecraft Search and Rescue scenario to jointly meet ASU/program-wide experimentation and CMU-TA2's April/May 2020 pilot test, including creation of in-world player trail, observer view, pausing system, scoring visualization, and experimental controls.
- Contributed field-of-view "can see" algorithm to Testbed Working Group. Worked with CMU-TA1 and UAZ to provide system-wide field-of-view testbed capability.
- Updated pre-registration templates to better suit capabilities, and reviewed others' Hypo/Capability pre-registrations.
- Created strategy annotation method and annotated Hackathon data with USC ICT.
- Collaborated with TA3 to add map complexity elements into Falcon designs.
- Provided PiP map intervention solution support to ASU and CMU-TA2 experiments.
- Provided fNIRS training session and led TA1 session (DOLL, CMU, UAZ, SIFT, USC) during Hackathon.
- Collaborated with Center for Open Science (COS) team to define process and key information in capability registration.

3.2 System Architecture

PSI-Coach builds on two emerging research areas, cognitive architecture designed to capture complexity of human natural behavior and Probabilistic Programming Languages (PPL), which are emerging technologies in inverting generative models for recognition. PSI-Coach is made up of three core elements: (1) the Cognition Inverter, which recognizes individual and team goals, behaviors, and mental states from open world observations; (2) the Strategic Loop, which projects team actions/successes, measures real-time team performance, and strategically plans when and how to intervene; and (3) the Experimental Loop, which performs experimental testing of PSI-Coach effectiveness and informs research for the Individual and Team Cognition Inverters and the Strategic Coach.

Figure 2 shows the PSI-Coach architecture. The core interaction starts at (*) with the Cognition Inverter observing the team's interactions with the open world. It then (1) infers the goals, mental states and behaviors of individual team members in detail, whether they are acting rationally or irrationally; and recognizes those goals, mental states and behaviors in the presence of the dynamic, and adaptive changes that people regularly exhibit in open worlds. Next, the Cognition Inverter (2) infers the joint-behaviors, shared mental models and misaligned mental models of the team, and uses all of this inferred data to compute PSI-Coach's real-time measures of team performance. These measures are passed to the Strategic Coach (3), which predicts possible futures, assesses team effectiveness in those possible futures, plans potential interventions, and then executes interventions at the appropriate times to help the team achieve high performance. As PSI-Coach monitors team activities, repeating this cycle, the Cognition Inverter is also continuously learning the strengths and weaknesses of the team and using that information to inform its predictions and intervention planning. PSI-Coach also includes (4) an offline, rigorous experimental cycle to inform research and improve the PSI-Coach system.

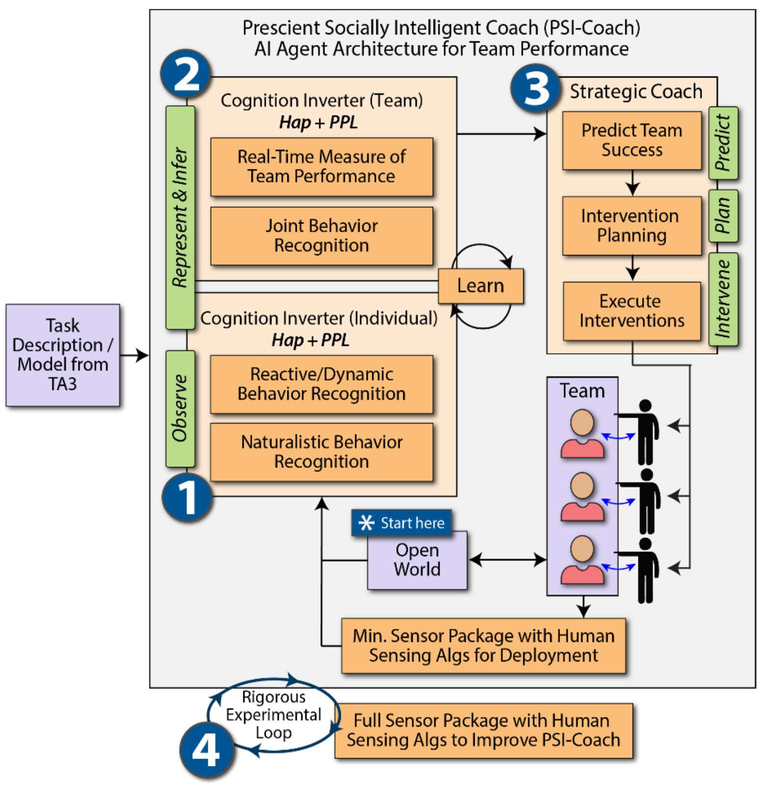


Figure 2: PSI-Coach functional architecture. *PSI-Coach extends the Hap AI agent architecture so it can understand and act on team goals, mental states, and behaviors.*

In the following sections, we describe each core PSI-Coach component using examples drawn from Minecraft as an illustrative open world environment. These examples are for descriptive purposes only; PSI-Coach is designed to work with any open world chosen by the program.

3.2.1 Hap Cognitive Model used in PSI-Coach

For PSI-Coach, we applied this emerging technology to a complex generative cognitive model, Hap, which was designed to capture natural human behavior. Charles River's Hap AI cognitive architecture (Figure 3) has been designed over 30 years to create believable, cognitively plausible, and socially realistic agents. Hap AI agents have been created for diverse domains; several examples are shown in Figure 3 (right). Examples include socially aware medical team training (CHARACTER), swarm collective reasoning (DARPA OFFSET), squad-level AI Army tactics (DREEMS), 4th generation combat pilots (NSGC), and cyber social-attack-vector simulation (MOC WAR). Several of these examples include multiple humans with specialized roles and skills in open world environments. Hap has also been used for dozens of research projects and a few commercial products; examples are shown in Figure 3 (bottom, right). Hap serves as the generative agent model for the PSI-Coach Cognition Inverter to recognize the complex unobserved mental states of human participants.

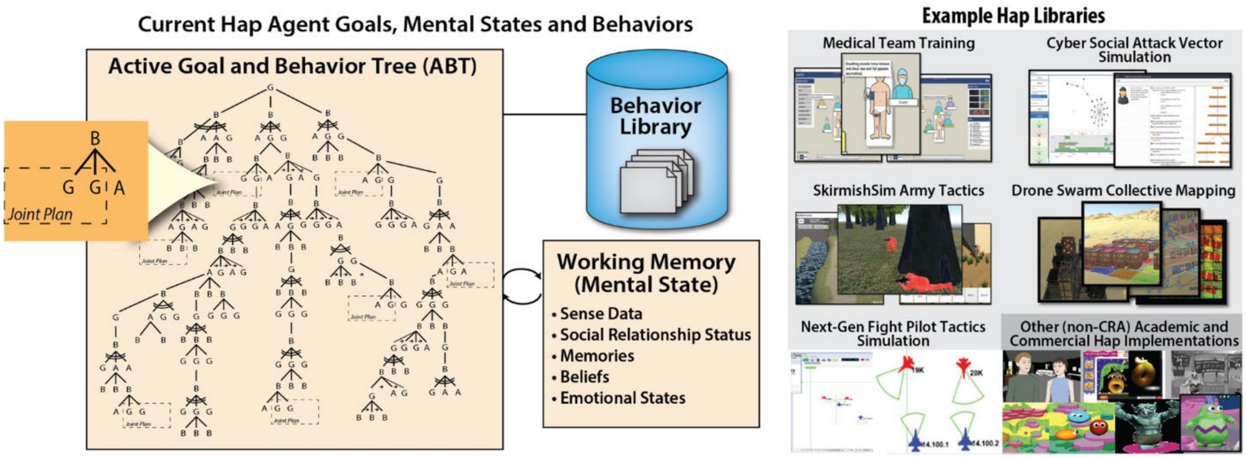


Figure 3: (Left) Hap AI cognitive architecture. (Right) Several example Hap applications.

The agent's mental state at any point in time is represented in its *Active Goal and Behavior Tree* (ABT) and *Working Memory*. The current set of goals and behaviors, and their current execution state, is represented in the ABT. Mental states are represented in Working Memory, including beliefs, sensed information from the world, emotional state, etc. Hap deliberately models *behaviors* rather than *plans* because they are used for both task-oriented activities, and for other human activities, such as "socializing," "playing around," or "exploring."

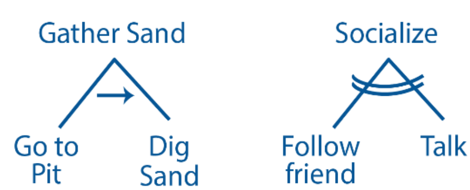


Figure 4: Simplified example Hap behavior fragments for multi-hypothesis recognition

For PSI-COACH, using PPL, we invert Hap to recognize the likely ABT fragments and mental states that could have given rise to the sequence of actions that were observed. For example, in a Minecraft world, if a team member is traveling with one of his team toward a beach, the Cognition Inverter might infer that he is going with him to help get sand, or that he is traveling with him to socialize, as shown in the two illustrative ABT fragments in Figure 4. Importantly, as illustrated by this simple example, when appropriate to the situation, the Cognition Inverter provides multiple hypotheses of the goals and behaviors that could give rise to the observed actions and the likelihood of each. For example, in this case, if there were prior evidence that the agent was trying to gather sand, such as previous actions that indicate a goal to use a furnace to make glass, the gathering sand hypothesis would be given a higher likelihood. This would also result in these behavior fragments having connections to the larger inferred behavior and goal of making glass, providing additional information about the collection of goals, subgoals, and behaviors that the person appears to be performing.

The Cognition Inverter updates its inferences regularly as new actions are observed. For example, if later the two people modified their trajectory so they were no longer traveling toward the beach, the hypothesis for socializing would be strengthened and the hypothesis for gathering sand would be weakened (or removed if weakened enough). The PPL algorithms that perform these inferences are described in Section 3.3.2.

3.2.2 Inverse Hap Modeling Framework

To implement models that can infer human cognition, we implemented a Domain Specific Language (DSL) we call Inverse-Hap. A model implemented under Inverse-Hap takes a sequence of observed actions from an individual or group as an input and infers their intent and behavior patterns. Specifically, the Inverse Hap interpreter has two key features. First, Inverse Hap utilizes the same grammar from forward running Hap, which enables the cognitive modeler to quickly implement a representative agent. Second, Inverse Hap leverages the inference capabilities of Scruff™ to convert the forward running Hap agent into a generative probabilistic model over possible agent state. The sub-sections below describe Inverse Hap in more detail. Sections 3.3 and 3.4 also describe example models implemented either directly using Inverse Hap or borrowing concepts from Inverse Hap.

Below we present a Backus Normal Form (BNF) version of the Inverse Hap Grammar:

ABTDefinition := BehaviorDefinition | GoalDefinition | ActionDefinition
Root := @model Steps
Goal := @goal SuccessTest <Args>
Action := @action SuccessTest <Args> <Func>
Args := {Arg…} | {} – An Array of Arguments
Arg := Any – Expression with Julia Constructs and Inverse Hap variable references
Func := Function
Behavior :=
    @sequentialBehavior ContextCondition <signature> Steps
    | @concurrentBehavior ContextCondition <signature
Steps := {Step…} | {}
Step := Goal | Action
SuccessTest := (Any => SFunc{Bool})
ContextCondition := (Any => SFunc{Bool})

3.2.2.1 Interpreting Inverse Hap Code

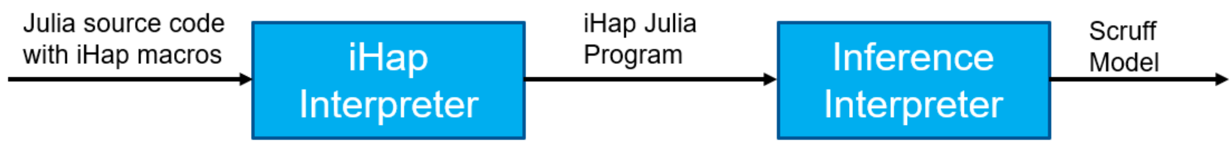


Figure 5: Inverse Hap has two layers of Interpretation

The Inverse Hap Grammar is comprised of several macro structures embedded in the Julia Programming Language. Specifically, a cognitive modeler can use the following macros to generate definitions for an Active Behavior Tree (ABT):

- @sequentialBehavior
- @concurrentBehavior
- @goal
- @action

These ABT definitions are automatically generated Julia code. This is the first layer of interpretation of Inverse Hap Code. The second layer takes the generated ABT definitions along with a definition of the agent's initial world state and feeds that into an Inference Interpreter to produce a Scruff Probabilistic model (specifically a Dynamic Bayesian Network) over possible ABTs and Agent World States.

3.2.2.2 Inference Using Scruff

Once we have our Agent defined using the Inverse Hap DSL, we can pass our Agent (i.e., the root of an initialized ABT) and our world simulator to an *Inference Container* which produces a probabilistic program in the form of a *Dynamic Bayesian Network (DBN)* that uses an *Asynchronous Particle Filter (aPF)* as an inference algorithm. The aPF under the hood uses the expansion/contraction logic around the traditional hap engine as a generative model to create probability distributions over possible ABTs. We can then assert evidence against this probability distribution by observing actions that a human performs to then infer the most likely ABT that would have generated said action(s).

3.2.2.3 Example Code

The following code is a simple Inverse Hap model of a frog living in a two-dimensional grid world. The macros defined here show the frog will simply move in the up and to the right. When interpreted, this will produce a single BehaviorDefinition data structure called Move, as well as two separate ActionDefinition data structures.

```
@hapRuntime()
```

The @hapruntim() macro must be called exactly once before defining any Inverse hap structures. This initializes the data structures necessary for capturing all Behavior, Goal, and Action definitions.

```
@sequentialBehavior move (i::Float64) (
    @preCondition(true)) (
    @contextCondition(true)) [
    @action moveUp (frog, grid) moveUp! (@successTest(false))
```

```
        @action moveRight (frog, grid) moveRight! (@successTest(false))
    ]

    root = @model main () [
        @goal move (Normal(0.0, 1.0)) (@successTest(false))
    ]
```

The inference mechanism for the above model simply takes as input the initialized root of the ABT, and the HypotheticalWorld representing the initial state of the frog agent.

```
    name = root.abtNodeFields.name
    id = root.abtNodeFields.id
    world = HypotheticalWorld(frog, grid, SequentialStructure(name, id, []))
    numParticles = 10
    inferenceContainer = @createRuntime(root, world)
    initRuntime!(numParticles, inferenceContainer)

    evidence = Dict{Symbol, Score}()
    runInference(evidence, inferenceContainer)
```

In this example, we run inference without evidence.

3.2.3 Adaptive and Reactive Behavior Complexity in Open Worlds

Once PSI-Coach can recognize natural human goals, mental states, and behaviors (whether rational, irrational, or idiosyncratic), the next ASIST research challenge is to recognize these mental states given the dynamic complexity that people pervasively exhibit in open worlds. People are complex and dynamic; they do multiple things at the same time, switch between goals and tasks, pause activities and return to them later, and react to opportunities and changes in the world. Traditional plan recognition and activity recognition systems cannot handle this dynamic behavior. Plan recognition systems recognize a user's specific stage in a rigid plan but fail in the complexity of an open world. Activity recognition systems (e.g., those using deep learning, rule-based systems or Bayesian graphical models) robustly recognize coarse-grained activities within the dynamic complexity, but they fail to provide the specific context of where in a plan an individual is stuck.

The Cognition Inverter provides detailed recognition of team members' goals and behavior execution state in these natural, overlapping flows of human activity by inverting Hap's automatic mechanisms for these reactive, adaptive, and dynamic cognitive processes. Unlike many other cognitive architectures, Hap was designed specifically to model reactive and adaptive cognition and behaviors. It has direct architectural support for multiple goals, for recognizing and adapting to opportunities and contingencies in the world, expressing multiple ways of accomplishing goals, deciding which method is more advantageous, and switching to new approaches when the current one becomes less advantageous. For example, as shown in the Minecraft ABT fragment in Figure 6, it is normal for Hap agents to gather gold and stone at the same time (as indicated by the parallel marker on the top level goal, and the absence of a "conflict" link between the two searching goals); dig stone after they find it (left); then switch to digging gold when a vein appears—the “dig gold” goal on the right is mutually exclusive and higher-priority, causing the switch.

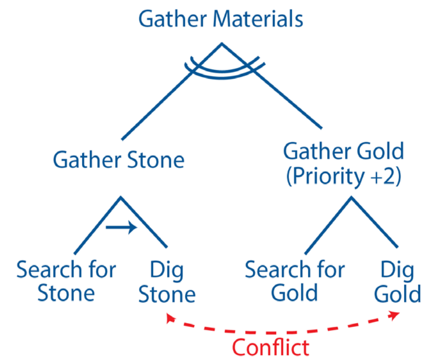


Figure 6: Example inferred behavior: User gathers stone then switches to mining gold.

To infer these properties, we extended the Hap system by: (1) replacing random choices in Hap with probability distributions inferred through PPLs; (2) using the Working Memory variables as latent variables to be inferred; (3) adding latent internal variables for goal switching and reactive changes so additional dynamic changes can be learned and recognized; and (4) adding dynamic noise to deterministic choices in the behavior models (e.g., step skipping) and changes in step ordering to support learnable exceptions to these choices. With these extensions, Hap can be used by the Cognition Inverter’s recognition; that is, when its PPL inference sees a sequence of actions that do not fit a single plan or activity, but instead fit a mix of partially executed activities, it can recognize these behaviorally plausible combinations. By only recognizing behaviorally plausible combinations, the inference is efficient and avoids spurious recognitions.

3.3 Cognition Inverter (Individual)

The first element of the PSI-Coach cycle is the Cognition Inverter (Individual) system, whose goal is to recognize individual human activities (e.g., behaviors, goals, mental states) in open world environments. At each point in time, the Cognition Inverter takes the stream of observed actions executed by an individual team member and returns the likely goals, behaviors, and mental states that could have given rise to those actions. The Cognition Inverter builds on recent breakthroughs in inverting complex generative models for recognition using probabilistic programming languages (PPLs). Inverting generative models to recognize unobservable states for graphics, illegal nuclear-test monitoring, and other domains has outperformed existing approaches, e.g., 60% fewer missed events of interest (Arora et al., 2013).

We used this approach in ProPL (Pfeffer, 2005) to invert generative AI agent models to recognize goals from observed human actions. Figure 7 shows how ProPL infers goals and behavior choices from logged office task actions performed by numerous human participants.

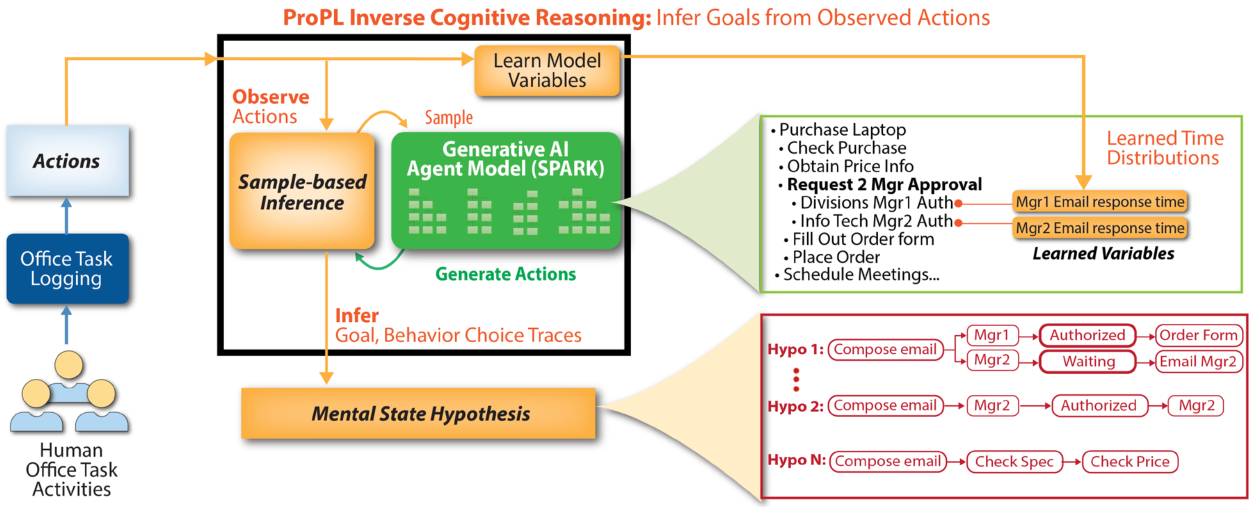


Figure 7: ProPL, our early implementation of a Cognition Inverter that infers goals from observed human actions in 4.6 ms with up to 93% accuracy

ProPL uses probabilistic sampling of SRI's SPARK agent generative model to infer unobserved human mental states in real time (0.8-4.6 ms). ProPL begins by reading a stream of office task actions as input. For each time step, ProPL runs a sample-based inference on SPARK agent models. SPARK models include general task steps that are parallel, sequential, or optional (at the top right are subtasks for laptop purchasing). We extended these models to include variables for performance (e.g., expected email response time), which our probabilistic language system learned during runtime or via offline training. The output of ProPL's sample-based inference is the set of agent goals and behavior choices that best explain the observed actions so far. ProPL's highest probability mental state hypotheses, shown at the lower right, explain the current state and predict likely actions. In this case, the output narrative for the first hypothesis is "The user reviewed the laptop purchase spec, browsed for prices, and emailed two managers for approval. Manager 1 authorized the purchase, but Manager 2 has not replied, most likely because he is historically slow to respond to email." This is a preliminary implementation of what we used as a PSI-Coach Cognition Inverter. On a simpler, non-open-world environment, it predicts various office task behaviors with 75%–93% accuracy. Subsequent work for PPAML enhanced the algorithms for higher performance and complex, interleaved behaviors.

3.3.1 Inferring Individual Cognition

The probabilistic model built for team cognition was implemented using the Figaro Probabilistic Framework but utilized many of the same concepts from Inverse Hap. Specifically, these inferences were achieved using a Dynamic Bayesian Network (DBN) running on a Particle Filtering Algorithm. Our Figaro models inferred the following individual based cognitive states:

- *Navigation Strategy:* Represents whether an individual is loitering and moving aimlessly, moving towards a particular location to perform a task, or is return to base to re-equip a new device
- *Victim Prioritization Strategy:* Represents the extent to which an individual is prioritizing critical victims, non-critical victims, or is simply opportunistically saving victims as they appear

- *Marker Semantic:* Represents what a particular individual believes a particular marker means. For instance, some individual might believe that a marker might indicate that there is a Regular or Critical Victim within the vicinity, while others might believe the marker indicates an absence of any nearby victims.

The Figaro Model infers theses outputs based on a sequence of observed actions from an individual. For example, if an individual enters a room with a regular victim and then leaves the room without having triaged said victim, the model likely will infer that the individual is prioritizing Critical Victims. Additionally, if another individual lays down a marker adjacent to an empty room, the model will likely infer that this marker semantic means No Triage-able Victim, in the context of this individual.

3.3.1.1 Additional Example in Cyber-Social Domain

We also implemented an example Inverse-Hap model that infers the intent of an adversary conducting attacks in a cyber-social campaign. In this example we implemented the following agent design, with the varying possible behaviors and actions to accomplish each goal.

```
  Agent Hypothesis:
  - Justify Russian military action in Ukraine
    - Promote domestic support within Russia for military action
      - BEND-Dismay, Target: Russian nationalists
        - Behavioral Strategy, Narrative Family Values
        - Behavioral Strategy, Disgust
      - BEND-Neutralize, Target: Russian dissidents
        - Behavioral Strategy, Spiral of Silence
        - Behavioral Strategy, Echo Chamber
    - Increase polarization abroad
      - BEND-Distract, Target: NATO & NATO Allies
        - Behavioral Strategy, Firehouse of Falsehood <- evidence
    - Promote international support of Russia for military action
      - BEND-Explain, Target: Non-traditional allies (Q Anon, anti-vax) &
traditional allies (e.g., China)
        - Behavioral Strategy, Secret Knowledge
        - Behavioral Strategy, Fear Uncertainty Doubt
```

We ran two separate experiments, wherein we fed the resultant Scruff Probabilistic Model an observed action that was unique to a particular behavior. In the first experiment, we fed an observation unique to the behavior “bendExplainSecretKnowledge”, in which case the Model inferred with *0.977* confidence that the behavior was “bendExplainSecretKnowledge”. In contrast, in our second experiment, we fed a unique observation for “bendExplainFearUncertaintyDoubt” and the model also yielded *0.977* confidence on the corresponding behavior.

3.3.2 Probabilistic Programming Inference Algorithms in the Cognition Inverter

Building on the recent successes in inverting generative models, PSI-Coach’s Cognition Inverter uses probabilistic programming inference to invert its generative model. Specifically, we use PPL inference to invert the Hap cognitive model to infer the goals, behaviors, and mental states that could produce the observed sequence of actions. For PSI-Coach, we provided rapid research progress with sample-based inference and scalability to large-scale behavior inference with lazy factor graph inference. This builds on decades of work in probabilistic programming inference by the principals, including advances made under DARPA PAL and PPAML.

Dr. Avi Pfeffer, our probabilistic programming lead, has done extensive work in probabilistic programming inference algorithms. His previous work forms the basis of our approach to the Cognition Inverter. Under the DARPA PAL program, he developed a probabilistic process language called ProPL (Pfeffer, 2005). Although ProPL is not as expressive as Hap, it includes many similar language features for describing probabilistically developing processes, including sequential and parallel execution of subprocesses, multiple choices of steps to achieve a process, conditional execution of a subprocess depending on state and events, and preemption of one process through completion of another process. ProPL used a sampling-based inference algorithm that was applied to office processes such as filing an expense report, arranging a meeting, or purchasing a laptop. This algorithm could then reason about details of the state of these processes with accuracies ranging from 77% to 93% in roughly one second. Refinements to the algorithm using a compact state representation achieved orders of magnitude speedup, delivering similar accuracy in only ten ms. These results were achieved on mid-2000s hardware. Although the models to which ProPL was applied were significantly simpler than the Hap behavior models for PSI-Coach, these preliminary results, combined with the significant developments in the core algorithms by ourselves and others and improvements in hardware over the past decade, give us confidence that we can achieve real-time performance under ASIST Phase I. As the inference complexity expands in Phase II and III, we expand from sample-based inference to lazy factor graphs (Pfeffer, Ruttenberg, & Kretschmer, 2016; Pfeffer, Ruttenberg, Sliva, Howard, & Takata, 2015a) with variable elimination (Rina Dechter, 1998), belief propagation (McEliece, MacKay, & Cheng, 1998; Pearl, 1988), and Gibbs sampling (Geman & Geman, 1987). Lazy factor graphs can scale to the size needed for ASIST by providing the efficiencies of factored inference, combined with the recent advance of lazy representation and processing, while maintaining accuracy.

In prior work with factored inference, our structured decomposition algorithm obtained orders of magnitude speedup on a variety of tasks over non-decomposed methods (Pfeffer, Ruttenberg, Kretschmer, & OConnor, 2018), and Hap cognitive models lend themselves to factor-graph structured decomposition because they are inherent hierarchical.

To scale to Hap's recursive structure, and large size, we extended factor graphs with lazy inference, which disregards irrelevant or less-relevant variables in a principled way. By disregarding the variables, inference efficiency can be dramatically improved. By doing this in a principled way, lazy factor graph inference can quantify the amount of error introduced by ignoring certain variables on the inference, allowing the method to intelligently select the variables to consider and ignore to achieve the required accuracy. Using our lazy inference methods (Pfeffer et al., 2015a), we obtained accurate inference results on very large and infinite grammar models that could not be solved by standard approaches. We were able to obtain very accurate (over 99% correct) answers to queries about grammar models with infinite factor graphs.

3.3.3 Cognitive Spatial Reasoning

We also designed and developed a cognitive spatial reasoning representation for inference over goals, strategies, mental models and knowledge related to movement, searching and acting in the open worlds, as shown in Figure 8. We explored algorithms to address the following research questions:

- How many hierarchical layers are necessary to simulate human cognition?
- How do we partition the two-dimensional world map in a natural way?
- Do the partitions need to be disjoint?
- How do we convert abstract concepts in the hierarchy into actuation; how does the partition relate to player movement in the simulated world?
- How can the hierarchical representation scale to the larger world map?

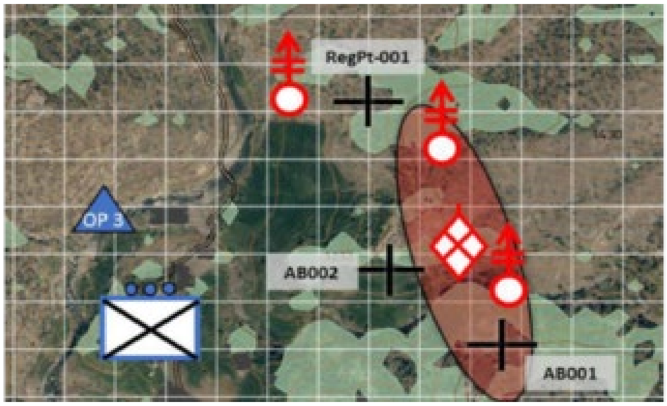


Figure 8: Spatial aggregation techniques based on past work

3.3.4 Cognition Inverter (Individual) Iterations and Improvements

During the Phase I effort, we developed an initial Cognition Inverter (CI) capable of inferring a player's navigation in a small 2D Minecraft simulation. A visual representation of this is shown in Figure 9. Our CI can recognize that a player is showing more optimal navigation versus "goofing off," from continuous, streaming observation data.

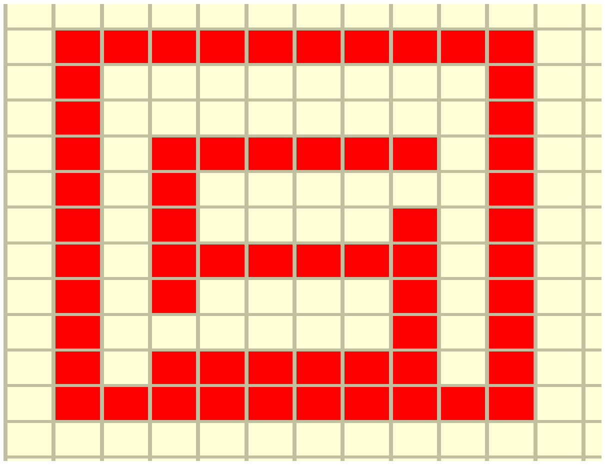


Figure 9: Cognition Inverter infers player's apparent goal based on navigation behavior style

We expanded the initial Cognition Inverter (CI) capability for inferring a player's search behavior during the search and rescue (SAR) task. The CI can recognize the type of search strategy, optimizing, satisficing, greedy, or hall-by-hall anchoring, and the "look-ahead" in the satisficing search. We also created models of location confusion, and memory and confusion models of completed areas. The CI models player movement with a geometric reasoning layer to handle movement between rooms, and a graph-based layer to handle movement across the entire building.

We integrated the CI with a Data Abstractor to demonstrate inference on the latest CMU experimental playback data and created visualizations to view and debug these inferences. Figure 10 and Figure 11 show two different inferences from two different player runs searching for victims.

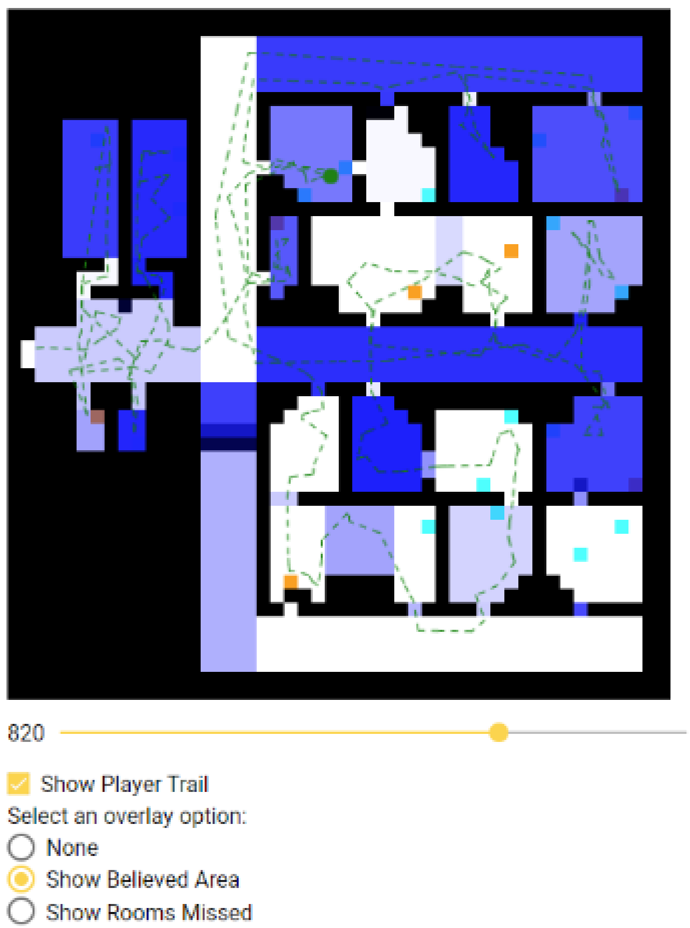


Figure 10: Inference from player run one. *Purple colored overlay indicates the Cognition Inverter inference of the player's confusion or knowledge of their location, with a scaled opacity gradation from 30%–100% showing probabilities from 0%–100% where the player thinks they are. In this case, the CI infers the player is confused.*

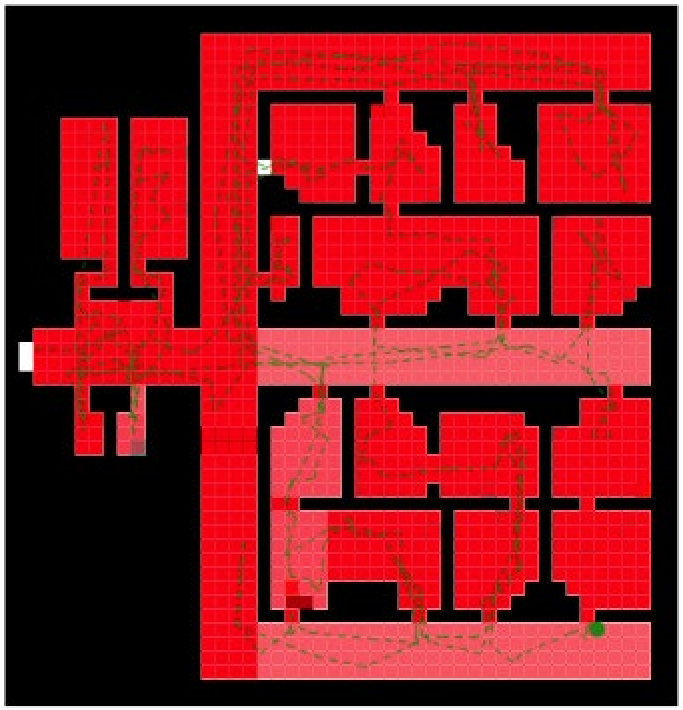


Figure 11: Inference from player run two

These visualizations show which rooms the CI infers the player thinks they have cleared or still need to search. These tools allow for deeper understanding of the multivariate, probabilistic inferences made by the CI, facilitating development of coachable predictions. For example, Theory of Mind (ToM) inference of the player may mistakenly believe they have already cleared the lower right room, but their predicted path will have them “stumble back in,” enabling intelligent “wait-and-see” intervention reasoning.

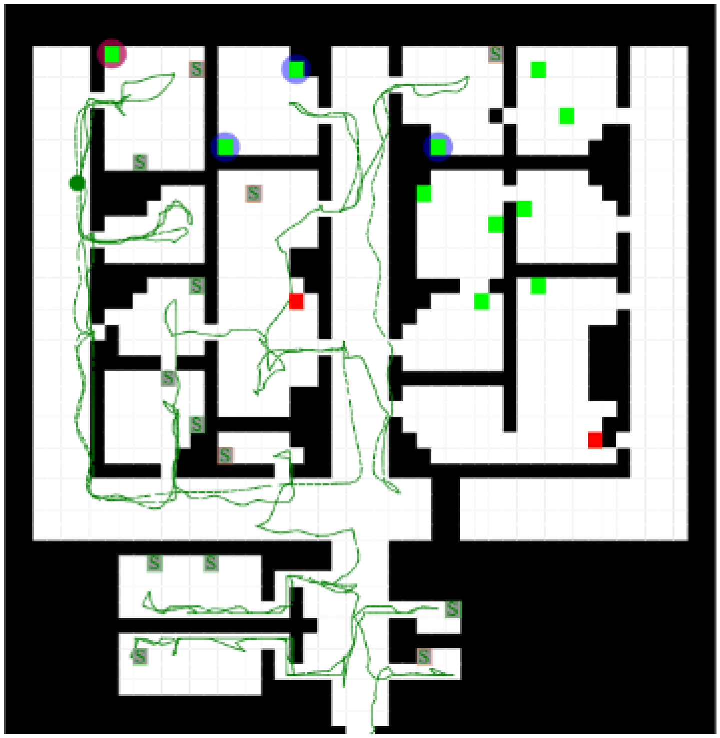


Figure 12: Cognition Inverter infers player's apparent victim triage goal (skipped on purpose—circular blue highlight around green victim square, skipped on accident—circular red highlight around green victim square) based on navigation behavior style

We also created visualizations to view and debug Cognition Inverter inferences. Two different inferences from a Hackathon participant's SAR mission observations are shown in Figure 13 and Figure 14.

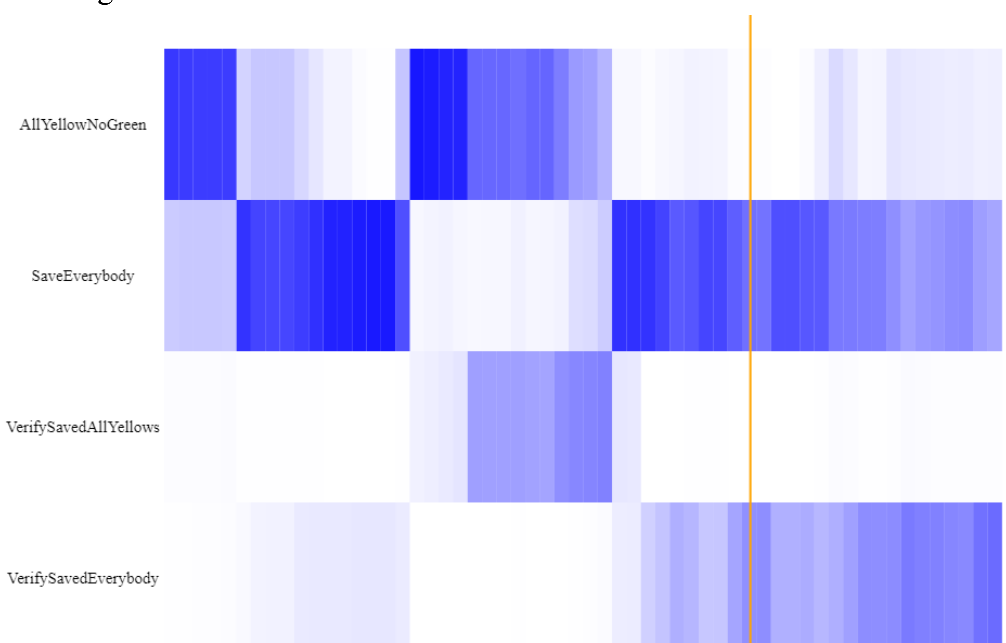


Figure 13: CI Inference visualization one shows PSI-Coach's inference of the human participant's strategy, in this case dominantly switching between the top two up until the current task time (orange vertical bar)

In Figure 14, the CI infers four possible player intents. The blue bands show confidence in each intent based on the movements of the player and which victim(s) they have decided to triage.

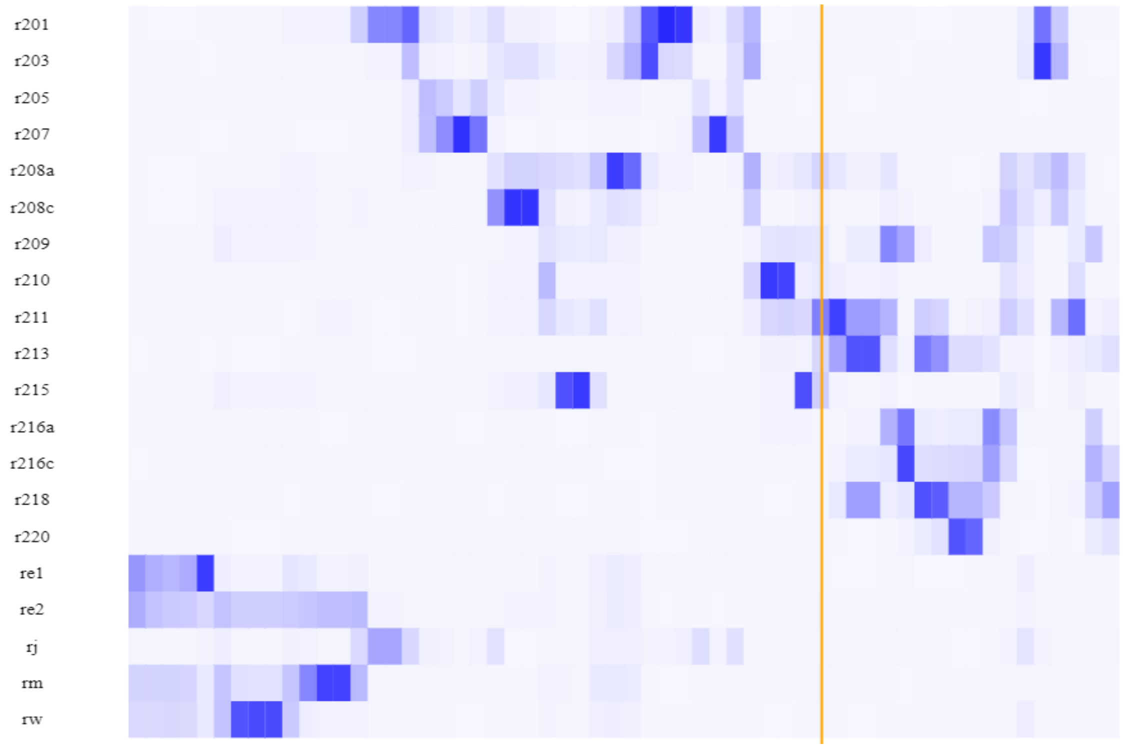


Figure 14: VI Inference visualization two shows PSI-Coach's inference of the human participant's intention to visit a certain room during their collaborative plan for this urban search and rescue session.

The CI also predicts rooms the player is most likely to enter in the next five event steps. The lefthand labels in Figure 14 are names of rooms. The blue bands show confidence in visiting that room within the time period.

We completed Cognition Inverter (CI) ToM inference and action prediction based on several key cognitive elements, including: Plan (e.g., victim triage prioritization strategy), Intent (skipped on purpose, missed on accident), Task Activity (navigation strategy), Resource Usage and Knowledge (device usage), Mental Model (semantic spaces), and Open World Observations (subjects' field of view). We improved inference performance, reducing processing time, and completed cognitive spatial reasoning for inference over goals, strategies, mental models, and knowledge related to movement, searching, and acting in the open worlds. Through our work, we refactored Path Planning and Victim Strategy Inference variables in the Cognition Inverter to eliminate duplicate computation when running inference. This resulted in four times the performance speed and slight reduction in memory usage. We also analyzed diminishing returns tradeoffs with memorizing paths in advance.

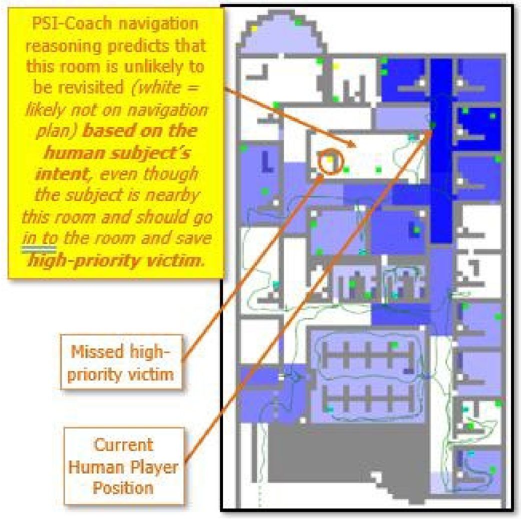


Figure 15: Cognition Inverter infers likely navigation plan based on player's individual navigation behavior style and intent (e.g., skipped on purpose/accident). *Darker blue indicates likely areas to visit. White/unshaded are predicted as unlikely to be revisited.*

3.4 Cognition Inverter (Team)

In addition to the individual goals, mental state, and behavior execution state of the team members, the aligned and misaligned shared mental models of the team are central to understanding and offering effective assistance to teams. PSI-Coach recognizes the team's shared mental models with a Cognition Inverter (Team) that has joint behavior representations, infers which joint behaviors are being executed, and then processes these inferences to compute alignment and skew. Our joint behavior representation builds on (Mateas & Stern, 2004), and simultaneously infers both the joint behavior a team member is trying to execute, and where they believe they are in their execution state.

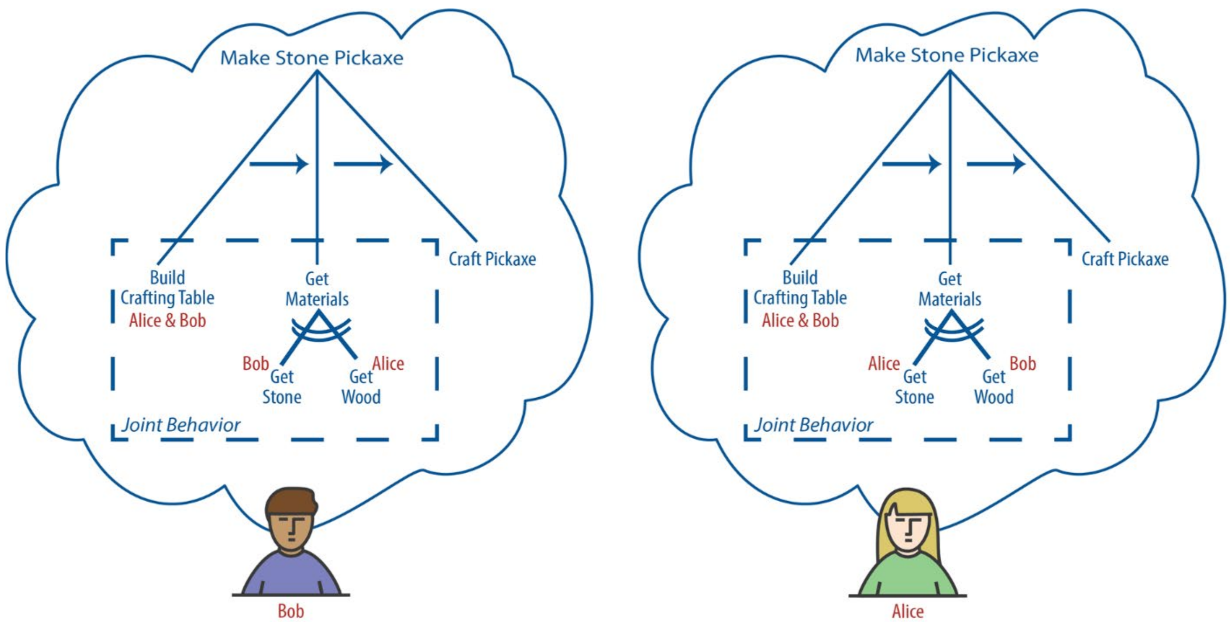


Figure 16: Example of shared mental model recognition with alignment and skew

As shown in Figure 16, these joint behaviors are recognized individually, per team member. Because people do not always have the same view of the plan they are executing as a team, PSI-Coach must recognize where the team's view is shared and where it is skewed. After inferring these joint plans, the Cognition Inverter compares views of team members to determine where they align and where they are either pursuing different joint plans, or are pursuing the same joint plan, but with misaligned models of the plan state. For example, in Figure 16, both Bob and Alice are performing a joint behavior to make a stone pickaxe in Minecraft. They were aligned at the beginning of the joint plan and were successful at building a crafting table together as the first step. In the second step of this joint plan, they are aligned in gathering the necessary materials, but they are misaligned regarding who is doing what, with both of them getting stone for the stone pickaxe head, and neither gathering wood for the handle.

PSI-Coach computes these areas of alignment and skew in two steps. Using PPL inference, the Cognition Inverter produces multiple hypotheses for the recognized joint plans and execution state for each team member, with inferred likelihoods for each. It then uses these hypotheses to compute the most likely matches of the recognized behaviors using a graph-matching algorithm. By considering multiple hypotheses, it can recognize shared mental models that would otherwise be overlooked. After computing matches, the Cognition Inverter computes the areas of alignment and misalignment for use by the Strategic Coach.

3.4.1 Inferring Team Cognition

The probabilistic model built for team cognition was implemented using the Figaro Probabilistic Framework but utilized many of the same concepts from Inverse Hap. Our Figaro models inferred the following team based cognitive states:

- *Victim Sharing Strategy:* Represents the extent to which teammates were sharing their knowledge victim locations with their teammates and the accuracy with which they were conveying said information. For instance, if one teammate indicates that there is a critical victim in a particular room, but in fact there was only a regular victim in the room, this would indicate the player was actively sharing information, but the accuracy of said information was low (albeit not completely incorrect, as there was indeed a victim in the room)
- *Task Allocation Strategy:* Represents the extent to which teammates were dividing the work efficiently, or if they were doing overlapping work. For instance, if there are two engineers clearing rubble in the same exact location, while there's rubble blocking access to victims in another location, the efficiency of this task allocation would be quite low.
- *Team Alignment:* Represents the extent to which teammates were performing tasks that synergized well together, or if some teammates were engaging in activities that may inadvertently prevent another teammate from completing their task. For instance, if a medic intended to triage a victim in one section of the building that's blocked by rubble, and the engineer is clearing rubble on the other side of the building, this would indicate a *mis-aligned* team.
    - Another version of this model would infer the extent to which all team members were engaging in the same high-level objectives (i.e., the same Victim Prioritization Strategy)

The Figaro Model infers theses outputs based on multiple sequences of observed actions from all team members. For example, if one teammate (a medic) is shown to be prioritizing Critical Victims because they have been actively ignoring regular victims while actively triaging Critical ones, while in contrast another player (the stretcher) is shown to be prioritizing regular victims by only moving regular victims, this would indicate significant misalignment in the team, and practically, would show that these two teammates are not operating synergistically.

3.4.2 Automatically Computes Real-Time Measures of Team Performance

The PSI-Coach Cognition Inverter (Team) computes practical real-time measurements of team performance for each team member across a number of dimensions using behavior indicators. We derive these measurements from studies of thousands of observed teams, and map them to generally applicable, higher-level team processes, including effective use of team memory, attention, and motivation. The Strategic Coach uses these processes to plan effective interventions. In this section, we describe our team performance measurements, how they are computed, and our approach to identifying effective, generalizable methods to measure them in real time.

3.4.2.1 Test of Collective Intelligence (TCI)

For PSI-Coach, we built on the well-established Test of Collective Intelligence (TCI) developed by our teammates at the CMU Collective Intelligence lab. TCI is based on empirical results of effective team performance collected over 10 years on more than 1,200 teams by the Collective Intelligence lab and is a cross-theoretic metric proven to be robustly predictive of team performance. TCI measurements consistently demonstrate team performance prediction despite broad differences in communication media (face-to-face versus online), group contexts (short-term ad hoc groups versus long-term groups), and cultural settings (US, Germany, and Japan), among other factors (Engel et al., 2015). TCI predicts team performance across diverse tasks, including complex video games like League of Legends (Kim et al., 2016).

CMU led the development of a real-time version of TCI for PSI-Coach based on their recent work to predict knowledge worker team performance (Gupta, Kim, Glikson & Woolley, 2019). Currently, TCI automatically calculates four measures: *accuracy* of task execution, *skill* of task execution (accuracy over a set of tasks), task *coverage* (number of subgoals and time remaining), and *level-of-effort* (behavior output over time and subgoal completion attempts). For PSI-Coach, we extended the TCI by computing it in real time and adding reasoning over human internal states. For example, a combination of TCI scores can indicate a person is failing at a task (e.g., level-of-effort is high, accuracy is low, and task coverage is unchanged, despite many attempts). PSI-Coach's inferred mental state information can pinpoint likely causes of failures (e.g., sensor indications of frustration and a missing prerequisite task resource). Figure 17 shows how we extended the underlying TCI theory and measurement approach to integrate with PSI-Coach's ability to recognize goals, mental states, and behavior choices.

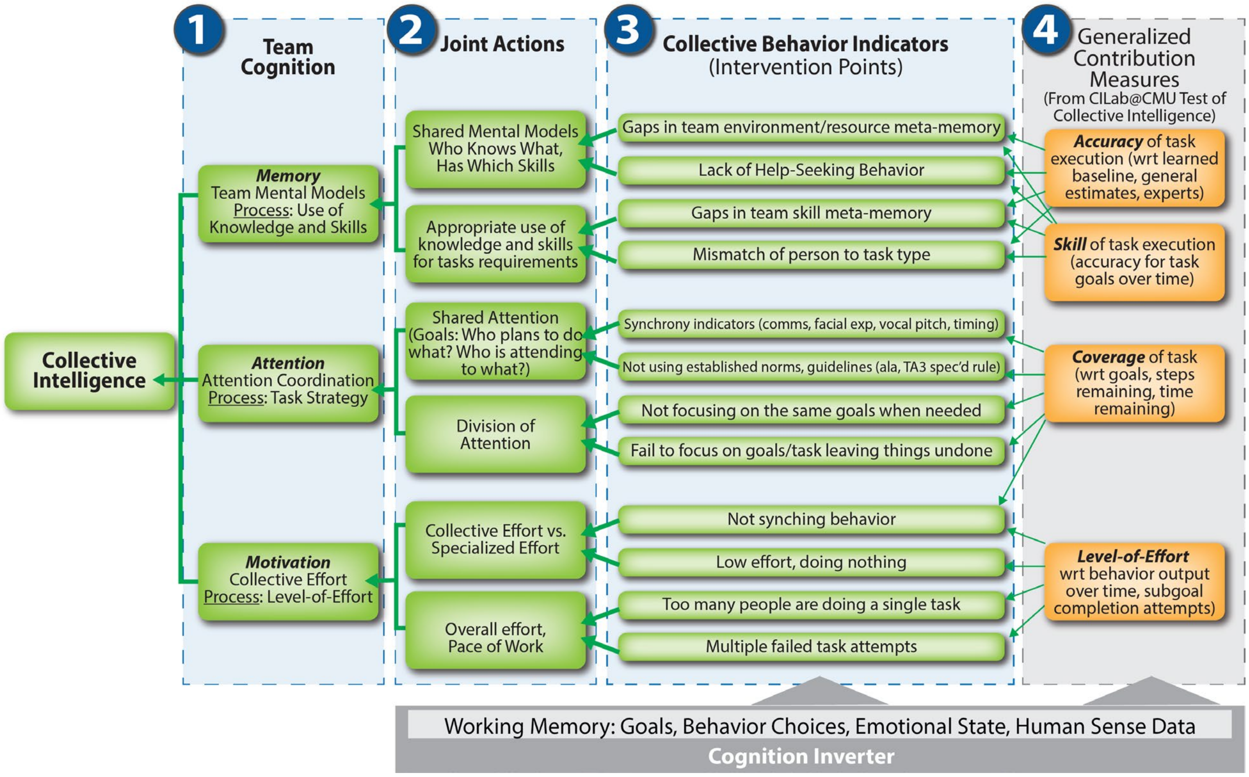


Figure 17: PSI-Coach extends collective intelligence metrics to compute real-time measurements of team performance, which enable effective interventions

3.4.2.2 Theory-Based Hierarchy

Similar to base TCI, a theory-based hierarchy (Figure 17, 1–3) provides a causal context for organizing and interpreting team and individual actions. Collective intelligence (CI) has three primary contributing elements of *team cognition* (1): team use of memory, attention, and motivation. Elements of team cognition can be instantiated as *joint actions* (2), e.g., "shared knowledge of who knows what" is a memory joint action; "who is planning to do what" is an attention joint action, and "collective effort" is a motivation joint action. Each joint action has many candidate *collective behavior indicators* (3). For example, "lack of help seeking" is a (negative) indicator of "shared knowledge of who knows what," which is a joint action for "memory." TCI contribution measures (4) can provide quantitative evidence for various collective behavior indicators. For example, a team member with high *level-of-effort*, low *accuracy* for a task and a history of low *skill* for the task type will have low contribution measures, which is evidence for the "lack of help seeking" because they do not seem to know they are trying hard and continually failing at a task they are not good at. The Cognition Inverter provides rich primary evidence for collective behavior indicators and generalized contribution measures for calculation.

This characterization helps us organize behavioral indicators and reason about specific faults, as well as construct reusable behavioral indicator recognizers for identifying and scoring collective intelligence at various granularities. For example, if three team members are collecting sand together when only one is needed, the Cognition Inverter (Team) activates the general recognizer for the collective behavior of "too many people are doing a single task." In this case, PSI-Coach assigns a low collective behavior score to the "too many people" indicator because level-of-effort is high, but coverage is unchanged, indicating poor team progress. Propagating through the hierarchy, this higher-level "collective effort" joint behavior is also scored low. The Strategic Coach can suggest people reflect on this higher-level "collective effort" team process and redistribute their effort to other tasks.

3.4.2.3 Approach for Generalizable Behavior

Our approach to identifying the most effective, generalizable behavior indicators draws from research on team process (Hackman, 1987) that shows high functioning teams: (1) match *member knowledge and skill* with roles and tasks for that member; (2) optimize *team task performance strategy* to allocate member contributions to the team's work; and (3) equalize and maximize member *effort*. We also explored using human sensing data from our minimal sensor package (e.g., keyboard tracker) for individual human state indicators or synchronized indicators such as eye-gaze or conversational turn-taking, which occur in teams that are more collectively intelligent (Tomprou, Kim, Chikersal, Woolley, & Dabbish, 2019). All these observables provide "honest signals" (Pentland, 2010) to continuously measure real-time TCI performance.

3.4.3 Cognition Inverter (Team) Iterations and Improvements

We implemented a PPL version of Hap cognitive architecture and language to enable larger-scaled Cognition Inverter (CI) models. We developed an interpreter for a restricted subset of the Hap language, explored inference algorithms, executed simple agents in a grid world environment, and performed rejection sampling inference on simple agents.

We designed and completed the implementation of an embedded domain specific language (DSL) version of probabilistic Hap and extensions in Julia, including the initial design and prototype of inference algorithms; we prototyped particle-filter inference over mind model chunks with simplified cognitive test model and completed a significant portion of lazy factored inference algorithm in Scruff for scalable cognition inverter inference. Figure 18 shows an example of using an embedded domain specific language, currently a subset of the Hap language, to reason about participant's active goal and behavior trees.

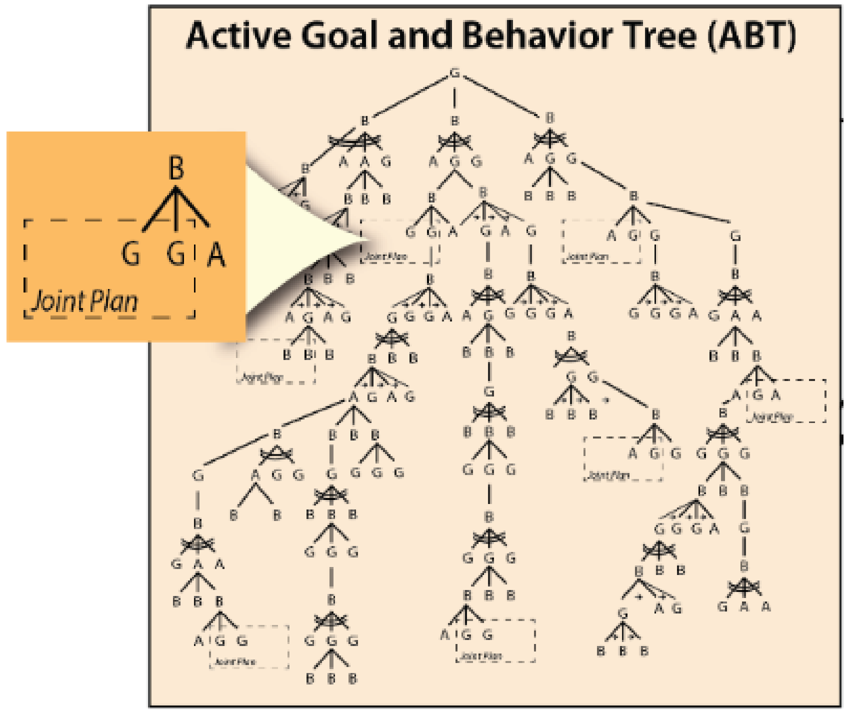


Figure 18: Embedded domain specific language reasons about participant's active goal and behavior trees

We then developed a probabilistic programming compiler plugin that enables probabilistic modeling of mental state in agent programs to augment the existing probabilistic language runtime. We completed the construction of more complex test agents, which include path planning capabilities and initial, rudimentary probabilistic reasoning, and initiated the development of agent models with more complex probabilistic representations of internal reasoning.

Through our continued efforts, we completed the PPL version of the Hap language with full-scope features (including post hoc probabilistic modeling of agent mental activity), and implemented a PPL factor construct that allows approximate Bayesian scoring of sequences of agent behavior. Our full scope version allows particle filtering inference with soft rejection sampling controlled by factor. Our additional language features included shared memory and language scoping.

Inference in a full scope system presents a few scalability challenges: agent programs express probability distributions over high-dimensional combinatorial spaces, with constraints expressed through approximate Bayesian scoring. To address these constraints, we developed a novel perspective on inference over agents: parsing sequences of agent action observations as generated from a probabilistic grammar. We also completed a working prototype of a system that generates agent grammars and completed initial tests with importance sampling inference. We designed an extended inference strategy that synthesizes agent programs from grammars applied to behavioral subsequences. Our expanded factor-based inference construct supports island-driven inference: performing inference over behavior fragments and combinations of fragments for increased scalability and robustness. This extends the inference capabilities from traditional inference over random choices facilitated by a modeler to inference over entire behavior models from observed actions in the domain. In addition, our initial designs for inference compilation technology speeds up inference and increases model coverage.

We completed the TeamPlayerModel API that supports each player's cognitive reasoning regarding their teammates (without treating the entire collection of players as a single entity). This API improved runtime performance when handling large amounts of tasks. Inferences include individual for role selection strategy (why a player selects a particular role), navigation/execution strategy (what a player is trying to do given their role), collections points, tasks (what role-specific actions this player is planning), teammate states (role and location), and how each member of the team is dividing and conquering. The TeamPlayerModel API allows team alignment and skew as well as collective effort inference across role selection, navigation, execution, and tasks. We streamlined skew calculation for ease of use and extensibility. The CI now includes inferences for handling Freeze Blocks and Marking Strategies and infers which heat map a player is using (if they have one).

We extended the Cognition Inverter API to work with multiple team members with end-to-end pipeline and demonstrated initial results on engineering data. The Cognition Inverter API supports backwards compatibility with single player model, enabling smooth scaling of team sizes.

By December 2021, our efforts ensured that the Cognition Inverter can now distinguish between strategies of teaming tighter versus divide and conquer, as well as victim prioritization strategies. The CI now separates deterministic and stochastic unit tests to increase test coverage by using strong typing for complex data structures.

3.5 Strategic Coach

The Strategic Coach predicts potential team futures, plans minimal, non-disruptive interventions, and intervenes judiciously using Charles River's experience management algorithms, extended with measures drawn from CMU's cross-theory metric of CI. Our Strategic Coach work extends research in interactive narrative guidance and real-time measures of team performance. The Strategic Coach projects a large number of possible futures: with and without Coach interventions and with different timing, projecting different team futures for each path, choosing minimal strategic places to "nudge" experience, and maximizing team success while minimizing disruptions and annoyances.

Intervention timing is just as important as intervention content (Hackman & Wageman, 2005), and good human coaches intuitively draw on Theory of Mind to optimize both. For example, if the team is building a pyramid out of glass, stone, and gold, the team must create the necessary tools in the correct order, including a wood, stone, and iron pickaxe. A timely intervention might include reminding the team that they must craft an iron pickaxe before they can mine gold. However, providing this guidance too early is ineffective, and providing it too often can become annoying.

The Strategic Coach reasons about intervention timing, probable disruption, and the projected impact on team performance and CI by extending Monte Carlo Tree Search (MCTS). This approach builds on decades of AI research, including foundational and recent work by our team, to steer interactive stories towards satisfying and impactful experiences while minimizing the feeling of disruption or intrusion (Weyhrauch, 1997; Mott & Lester, 2006; Riedl & Bulitko, 2012; Wang et al., 2017; Robertson & Young, 2018). The Strategic Coach includes computational measures of intrusion and steering effectiveness that enable measures of CI, consideration of timing and order (Weyhrauch 1997; Kelso & Weyhrauch, 1994), and intervention effectiveness (Woolley et al, 2010; Woolley et al, 2015). After projecting a massive number of possible futures, intervention selection is guided by measures of intervention efficacy, need, and degree of disruption. For example, based on predicting possible futures, the Strategic Coach may decide to provide an immediate reminder now about a nearby victim who needs help, or may determine that *on balance*, it is better to wait and see if the team figures it out themselves because more important hints are coming in the near future, and giving a hint both now and later risks efficacy and annoyance because the participant might begin to tune out the hints. This approach enables PSI-Coach interventions to feel well-timed, tactful, and immediately useful, dramatically improving efficacy and team acceptance.

In our previous work, we used the insight that experience management can be cast into the paradigm of an abstract, two-sided game (Figure 19, left). Using this paradigm combines *prediction* and *planning* into a single algorithm where multiple possible futures of user activity and interventions are projected. One side of the game is the human team performing a task. The other side of the game is the strategic coach executing an intervention or doing nothing. This insight enables the application of adversary search algorithms (e.g., Silver et al., 2016) to experience management. Unlike in Chess or Go, this is a win-win situation, because both sides of the game (team and strategic coach) are trying to succeed together, which allows this kind of turn-taking-based search to converge with more stability than in adversarial games, which facilitates its use.

After the adversary search algorithm completes its prediction and planning, the Strategic Coach applies a second layer of rule-based reasoning to capture additional factors not fully modeled in the adversary search algorithm. For example, if an intervention is suggested based on specific features of the current world state, the intervention might be delayed if a recent intervention that might affect those features has not had sufficient time to cause a detectable effect. This secondary reasoning layer enables the Strategic Coach to incorporate simple heuristics to guide interventions without unduly increasing the complexity of the projection of possible futures, which must be highly performant.

Finally, the Strategic Coach *intervenes*, selecting either the most effective intervention, or waiting. Interventions are accomplished by communicating information or hints to the team or individuals using our focused NLP technology (Section 3.5.3.1). For example, an intervention might use text chat or speech to say to the team member who seems to be searching for gold, "Mining gold requires an iron pickaxe, so don't forget to bring one along." In our previous work using these techniques for interactive narrative experiences, an experience manager improved the story experience at a 95% confidence interval for average users vs. no guidance (Weyhrauch, 1997). This foundational work has been applied and extended to many interactive AI and entertainment systems (Mott & Lester, 2006; Nelson, Mateas, Roberts, & Isbell, 2006; Nelson & Mateas, 2008; Riedl & Bulitko, 2012; Wang et al., 2017; Robertson & Young, 2018), including successful commercial games (Left 4 Dead 2, Valve 2009), and PSI-Coach leverages recent advances in search efficiency, machine learning, and problem representation to maximize the impact of this approach.

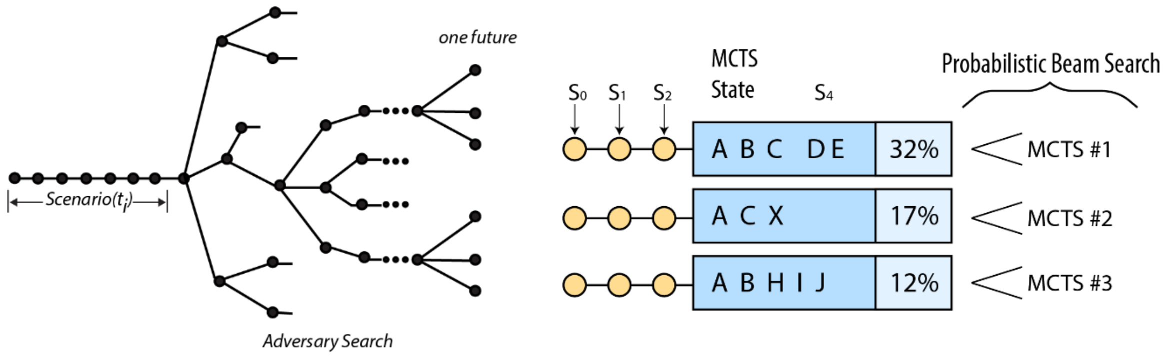


Figure 19: (Left) Strategic Coach prediction and planning cast as adversary search; (right) the Cognition Inverter infers the most likely mental states and the Strategic Coach searches to predict future actions and information needs so it can plan interventions. *The Strategic Coach intervenes based on the best possible choice, or waits, if that is a better option.*

Inputs to the Strategic Coach come from the Cognition Inverter, which give a probabilistic interpretation of "what is the team doing and thinking now" by matching data to possible individual and team mental models. Because actions can have multiple purposes (or no discernable purpose if the actions are non-rational), this probability cannot be inferred with 100% certainty based on behavior. Instead, the Cognition Inverter provides a best match of available data against possible options for including confidence measures for each mental model. Should perturbations occur that shift team or individual goals or tasking, the Cognition Inverter recognizes them and the Strategic Coach then plans for these changes.

For example, imagine a team attempting to build a pyramid in Minecraft (goal A). The team's mental model may contain facts about the world state and ways to achieve goal A, including subgoals that culminate in goal A (i.e., gathering sand, forming blocks, stacking blocks). Based on team actions, the Cognition Inverter infers a set of potential likely mental models. Figure 19 (right) shows three of these potential mental models that describe different ways to achieve goal A, based on observations of team actions. These three mental models represent only likely candidates—there is potentially a large tail of unlikely models that are discarded from consideration. Both the upper and lower models think they have evidence of subgoal B (crafting a stone pickaxe), but the middle thinks the data supports subgoal C (gathering sand as a precursor to making glass) as part of A. The confidence Cognition Inverter assigns to the model is shown to the right, using a percentage. (D, E, X, H, I, and J represent other subgoals and team beliefs.)

We used a beam search approach, considering only the most likely interpretations. To simplify the search space, the Strategic Coach uses the set of inferred active goals rather than the entire mental state. Each possible future is explored using MCTS to determine the move that maximizes the value of the current position (including doing nothing). Each search returns a set of possible next moves, valued by their benefit. These moves are combined according to the likelihood of the initial state to determine which move to make (including no move).

3.5.1 Monte Carlo Tree Search

Our approach included an abstract search space that captures team moves (those made by the team, Strategic Coach moves (interventions), and an evaluation function that determines the value of the current and projected game.

We defined **Team moves** in terms of the Hap and the Cognition Inverter Theory of Mind. Changes in the behavior state (as represented by changes in the set of active goals of the team) represent the team moves for Strategic Coach reasoning. For example, if the team is building a pyramid, and they finish collecting gold, that is a “move” in the game state. At a lower level, tactics for finding and making tools are also moves (e.g., “smelting iron” or “harvesting wood.” Another type of goal might be more team oriented, like “balance the workload among team members” or “communicate progress”). Based on the Hap theory of mind, goals can refine (e.g., add subgoals), complete (succeed or fail), choose new strategies (switch goals), and other transformations. The Strategic Coach reasoning component predicts possible future actions by choosing sequences of updates to the set of active goals according to this theory to show the transitions in the team state, starting with the Cognition Inverter current best estimates of active goals.

Strategic Coach moves represent the interventions available to influence and guide team performance and CI. How interventions are designed and how they are strategically and judiciously used to influence performance and CI (while not annoying the team) is a major element of the Strategic Coach. Example interventions include reminding a person mining gold to remember to craft an iron pickaxe first, since it is required to mine gold, or reminding team members to communicate their goals if two people are engaging in redundant behavior. Our intervention design augments the approach in (Weyhrauch, 1997) with the approach in (Woolley et al., 2010; Woolley et al., 2015).

The MCTS **evaluation function** ingests the entire sequence of moves to determine the score for the particular future being projected. Unlike chess, the evaluation function cannot simply use the “final position” to evaluate the choice of the interventions because interventions have an effect over time in how they promote team effectiveness and generate annoyance. Building on (Weyhrauch, 1997), we calculate the evaluation function over the sequence of each projected futures, which allows calculation of CI, task success, and disruptiveness. After projecting potential future paths, this evaluation function scores each path with these measurements allowing it to choose the interventions that are projected to lead to the highest quality outcome.

3.5.2 PSI-Coach Seamlessly Assists Teams During Perturbations

PSI-Coach is designed to seamlessly assist the team through perturbations as they occur, and advise the team to high levels of performance in the perturbed situation. From a system point of view, PSI-Coach’s reasoning is performed moment-by-moment, so it can always use the most recent information to understand team members and predict and plan possible interventions. Each PSI-Coach is instantiated *per team member*, so the PSI-Coach system can perform its functions whether the team is connected or disconnected (although remote activities are less visible when disconnected). From a team-performance-theory point of view, PSI-Coach’s core mechanisms—understanding team mental states, measuring collective intelligence indicators, predicting futures and subtly guiding the team members to ways of thinking and acting that overcome suboptimal behaviors—are designed precisely to shine in situations where perturbations lead to suboptimal choices. Table 2 describes how PSI-Coach responds and addresses each perturbation.

Table 2: PSI-Coach perturbation response

Perturbation	System Response	Cognition Inverter Adaptation	Intervention Help to Team Adaptation
Disconnected from team	Each individual PSI-Coach continues assisting team members.	Most team understanding and CI calculations remain fully effective. Interventions based on team communication, and calculations of full team accomplishment are reduced, and worked around until comms are restored.	Effort, appropriate task strategy, use of knowledge, and skill-based interventions continue to assist the team through the perturbation. Comms-based actions are prioritized if restoring comms is possible. Focus shifts to sub-team performance as uncertainty grows about disconnected members.
Member substitution	Create new PSI-Coach for new member; start learning strengths and weaknesses; remove old team member.	Appropriate use of skill uses new skill mix; shared mental models start fresh for the new member.	Interventions urging team communication for skill coordination increase. Strategies leveraging new member strengths increase. All others are unchanged.
Retasking	Reasons with new task	Task-based reasoning uses new tasks vs. old. All others unchanged.	Interventions based on inappropriate use of skills on old vs. new tasks are more likely.
Environment change affecting strategy	Cognition Inverter removes deprecated strategy and adds any new ones.	Individual/team efficacy calculations are all based on efficacy in environment, so change propagates seamlessly	Strategic Coach urges remaining and new strategies, and steers away from deprecated ones.

3.5.3 PSI-Coach Provides Additional Team Sensing with Appropriately Scoped Natural Language Understanding and Human Sensing Tailored to the Minimal Sensing Package

3.5.3.1 Appropriately Scoped Natural Language Understanding

PSI-Coach is designed to handle any direct natural language communications between the team members necessary to meet the research goals of ASIST. Our natural language processing (NLP) approach is focused so that it does not distract from the main research priorities of the program, while still able to handle the types of informal communication that people use when they work in teams. We built on the open-source Ravenclaw/Olympus Architecture (Bohus et al., 2007), a well-tested framework for robust natural dialog, with several applications deployed in both research and real-world settings, including the Intelligent Procedure Assistant, a prototype system that supports astronauts at the international space station with large procedural tasks, and TeamTalk, an interface that allows a human to direct a team of robots to complete a task in a shared space. Ravenclaw/Olympus is based on robust semantic parsing. By using semantic parsing and dialog management technologies, we can handle common flexible language with targeted semantic grammars to limit the scope of the NLP effort. Ravenclaw/Olympus supports both speech and text-based communications, allowing flexibility to the communication design with TA3 performers.

3.5.3.2 Human Sensing Tailored to the Minimal Sensing Package

We developed a minimal sensor package to provide conformational evidence for inferring human internal state. We select sensors and algorithms from the experimental sensor package (discussed in Section 3.6.1) for inclusion in the minimal sensor package that most strongly indicate human cognitive state by tightly correlating mental states to physiological measures and minimizing required hardware.

Keyboard and mouse devices are the core of our minimal sensor package. These sensors reliably predict cognitive workload or stress (e.g., Lim, Ayesh, & Stacey, 2015) and they do not require additional hardware. These indicators are computationally efficient—they include metrics such as mouse idle time, left mouse click time, and keystroke rate—and PSI-Coach tracks them on the fly to produce state indicators with high temporal resolution. These activity levels from keyboard and mouse will validate team contributions to improve state inference accuracy.

We also considered webcams that provide measures of arousal and cognitive workload such as pupil dilation (Elkin-Frankston, Bracken, Irvin, & Jenkins, 2017). We will use these indicators to register the degree of synchrony between team members. Physiological and state synchrony in members of a team is an effective predictor of team performance. When available, webcams provide insight into these same physiological states (e.g., Maaoui, Bousefsaf, & Pruski, 2015), while also examining facial synchrony across team members (as used to predict negotiation winners in Li, Curhan, & Hoque, 2015). Based on our prior work, there are many combinations that may be strong predictors of internal human state. We empirically validated correlations and tuned our processing algorithms for efficient human sensing for distributed participants.

3.5.4 Strategic Coach Iterations and Improvements

We designed Strategic Coach projection of futures without interventions for Phase I.

We met with Dr. Broach, a UMass expert in Disaster Medicine, and documented several high-value behaviors, goals, mental states, and knowledge elements for use in the Cognition Inverter and Strategic Coach components. We determined that useful candidates for initial Theory of Mind (ToM) research in Search and Rescue tasks include the following:

- Revisiting the same areas needlessly, purposefully, or fortuitously while looking for victims
- Missing important rooms (with yellow victims early or green victims late) or not correctly clearing rooms
- Recognizing accidental versus purposeful skipping of victims
- Missing victims to be found later versus inefficient/large future cost

We created initial designs for integration of Cognition Inverter outputs to Strategic Coach for longer-duration, aggregation-based predictions, and intervention reasoning. We implemented an initial Strategic coach for an initial set of capability statements, focusing on creating combined predictions that (1) help distinguish between effective and annoying interventions, and (2) facilitate reasoning on best timing of the interventions.

The Strategic Coach reasons about uncertainty, e.g., maybe the Medic missed the room mistakenly or maybe they are planning to go to it later. The Strategic Coach intervenes based on the full collection of uncertainty, including determining when it would be "too late"—can the intervention "wait and see" because the Medic won't be too far away from the room or does it need to occur immediately? We created the framework for effective interventions, including the following guiding principles: "Say just the right thing at just the right time," no joysticking, and avoid interventions that are acceptable in an artificial task (simulation) but negative in real-world operational settings.

The Strategic Coach incorporates a task-oriented model of cooperation to align with theories of Collective Intelligence and enable projection of appropriate allocation of knowledge, skills, tasks, and effort. The general plug-and-play intervention reasoning framework enables analytic component insertion.

We also increased ToM projections and intervention reasoning to teammates (two additional players). And we designed methods to search potential interventions tractably and initial methods to score potential interventions.

3.6 PSI-Coach Performance Validation

To continually improve PSI-Coach, we validated PSI-Coach performance in a number of ways. We developed a series of test metrics to assess PSI-Coach component and system performance, and tested these metrics during a series of lightweight pilot studies and during program-wide evaluation events. We used the pilot studies to gather training data and validate that PSI-Coach generalizes across different human participants, tasks, and perturbations prior to program evaluation experiments. Both experimental cycles allow us to adapt the most effective human sensing technologies, assess intervention efficacy, and incrementally validate the most technically challenging aspects of the PSI-Coach research program.

3.6.1 Human Sensing

The human sensing capability within the rigorous experimental loop serves two purposes. First, a broad set of human sensing hardware and techniques within the rigorous experimental loop will establish which most tightly link with human cognitive state. The most deployable algorithms that can predict the broadest range of human states will comprise the PSI-Coach minimal sensor package (Section 3.5.3.2). The broader set of human sensing techniques used during our rigorous experimental loop also provide feedback on the function and effect of PSI-Coach by linking validated observable indicators of human cognitive states (e.g., cognitive workload, stress, fatigue, misbalancing of workload across team members) with in-world behaviors.

We conducted human sensing technical work and early experimentation in partnership with ASU until COVID-19 restrictions restricted in-person human experimentation. The following description captures the research context and progress up until early 2020.

We used Charles River's mature (TRL8) Sherlock™ software product to identify and validate links between team member physiology and behavior (e.g., with eye trackers and standoff cardiac monitors embedded in laptop webcams, mouse/keyboard analytics) that correlate with task-, team- or communication-relevant cognitive states such as cognitive workload, attention, or fatigue (Elkin-Frankston, Bracken, Irvin, & Jenkins, 2017; Vidulich, Stratton, Crabtree, & Wilson, 1994; Wilson, 2000; Xu, Slagle, Banerjee, Bracken, & Weinger, 2019) during task performance. For example, synchronous changes in cardiac information, mouse movement, facial expression, or even body sway can indicate individual and team optimal and sub-optimal states (Bracken, Alexander, Zak, Romero, & Barraza, 2014; Chikersal, Tomprou, Kim, Woolley, & Dabbish, 2017; Hasson, Furman, Clark, Dudai, & Davachi, 2008; Yun, Watanabe, & Shimojo, 2012). Table 3 shows a candidate list of sensors for inclusion in PSI-Coach for each Phase. In the traditional "crawl, walk, run" approach, we included as many sensors as possible during the first evaluation to provide complementary signals validating our model output (crawl). Based on these initial experiments, we reduced this sensor suite to those minimally necessary for accurate modeling of individual and team state to determine when PSI-Coach can make maximally valuable interventions (walk). Sensors included in the "minimal sensor package" column are those unobtrusive enough to deploy in real-world environments, while providing enough information to accurately generate actionable human-state intelligence for PSI-Coach (run).

Table 3: Sensors implementation plan for each stage of PSI-COACH validation

Sensor	Phase I experiments	Phase II experiments	Minimal Sensor Package	Human State Indicator
Eye tracker	If possible	If possible	No	Individual cognitive workload, fatigue, situational awareness, attention
Facial expression (webcam)	If possible	If possible	If possible	Individual affect, team state
Heart rate (webcam or worn)	Yes: webcam and worn	Yes: webcam only	If possible: webcam only	Stress, cognitive workload, team state
Text chat analyzer	Yes	If possible	If possible	Stress, cognitive workload, team state
Keyboard tracker	Yes	Yes	Yes	Stress, cognitive workload, team state
Mouse tracker	Yes	Yes	Yes	Stress, cognitive workload, team state
Seat position	Yes	If possible	If possible	Cognitive workload, attention, team state

3.6.2 Intervention Efficacy and Counter-Efficacy

To improve the efficacy of interventions and avoid counter-efficacy, we applied research on the efficacy and disruption of interventions from Charles River's Interactive Narrative work and research on the performance impact of interventions from CMU's Collective Intelligence work to create an initial set of effective interventions. Over the course of the program, we used the experimental loop to continually extend and refine this set for efficacy, disruptiveness, and context.

According to Hackman and Wageman's Theory of Team Coaching (2005), effective interventions focus on use of member knowledge and skill, task strategy, or effort. Three transition zones provide the best time for deployment: early, before norms are established; at a midpoint or milestone; and after accomplishment of a major objective, to consolidate lessons learned. Gupta et al. (2019) recently conducted a study using a synthetic facilitator to enhance team use of knowledge and skill, task strategy, or effort. In online, short-term teams of strangers, the most effective facilitator type focused on use of member knowledge and skill; other intervention attempts had a neutral or negative effect on performance. They reasoned that the strategy facilitator interjected too often and distracted teammates, and the effort facilitator led the hardest-working team members to withdraw effort. In addition, research from CMU Prof. Gonzalez indicates that although the memory support provided by group interactions is performance enhancing in a stable environment, it impedes adaptation in a changing environment (Lejarraga, Lejarraga, & Gonzalez, 2014). When using interventions to facilitate team performance, content, timing, and team composition must inform the intervention selection. Using the experimental loop, we extended this combined body of knowledge to both existing and new interventions using empirical measures of efficacy, human sensing data, and interviews with participants to establish the degree of annoyance and effectiveness of interventions, and to refine the context in which effectiveness manifests.

3.6.3 Validation Activities throughout Effort

This section provides a basis for our validation activities. See Section 3.7 for the results from our Internal Pilot Studies.

We led ASIST's Data and Measures Working Group to establish the requirements for a program-wide data and performance measurement approach and the Experimentation Working Group to establish the research hypotheses to drive the program-wide experimentation approach. We submitted the following hypotheses specifically for CRA techniques:

1. The PSI-Coach agent can infer participant mental state from their actions.
2. The PSI-Coach agent can distinguish four levels of a participant's success in applying a specific strategy:
   a) Participant is not applying the strategy.
   b) Participant appears to know and be attempting the strategy but is unsuccessful in executing it.
   c) Participant knows and is successfully applying strategy without improving performance.
   d) Participant knows and is successfully applying the strategy and is improving performance.

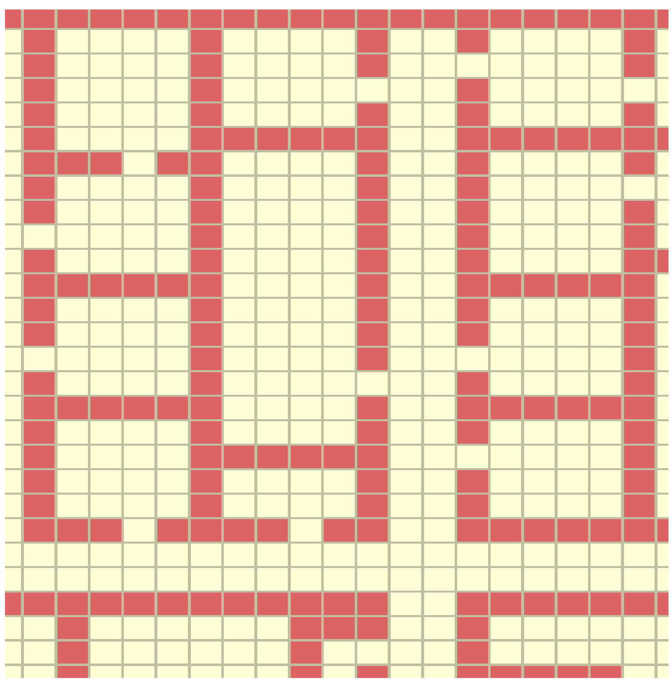


Figure 20: An example test designed to drive PSI-Coach capabilities based on quantitative data and to validate results

Figure 21: Provided fNIRS hardware to ASU

We implemented the following Multiple Capability Statements for Predicting Elements of a Human's Mental Model:

- Focusing on predictions with enough richness of information to enable effective coaching that is helpful rather than annoying in subsequent phases of the program (e.g., predict when a subject is missing green victims accidentally or skipping them on purpose)
- Recognizing and predicting individual differences
- Recognize aspects of social cognition that can naturally expand from our initial experiments with individuals to subsequent experiments with teams

Figure 22: Provided fNIRS hardware support to ASU

We processed and shared fNIRS data from ASU participants during the last in-person hackathon. Due to COVID-19, further physiological data collection and analysis is on hold due as we are unable to collect physiological data at ASU. Instead, Northrop Grunman will collect data on internal personnel using their fNIRS sensor. This sensor provides 16 channels of data (compared to the Explorer's 2 channels). Northrop Grunman has these sensors readily available. Therefore, we planned to use 16 channels of data during the Spring 2020 data collection, if possible, to provide data consistency and additional channels of data (the larger sensor is not seen as prohibitive).

Figure 23: Review of human actions to validate strategic coach's goal predictions using hackathon data

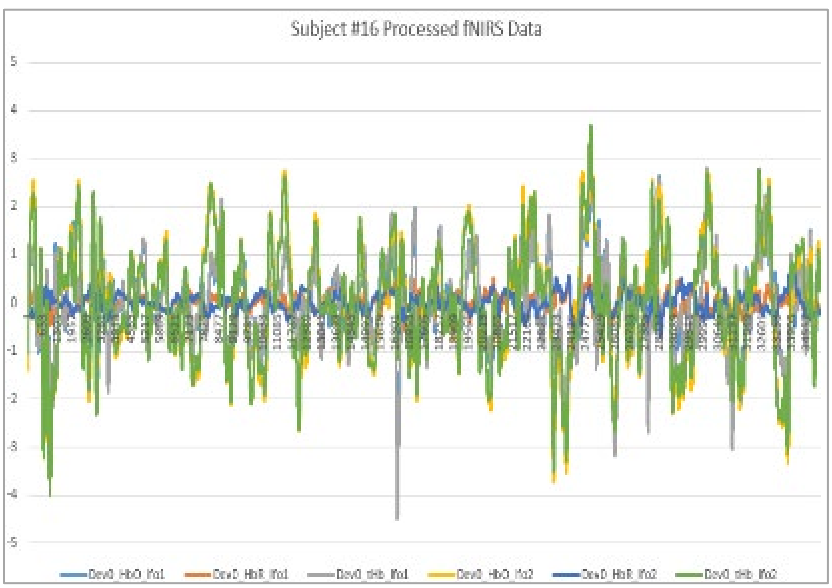


Figure 24: Designing tests to drive PSI-Coach capabilities based on quantitative data, also used to validate results

We completed initial findings from ASIST Study 1 by comparing PSI-Coach with human coach annotators (using annotation UI, see Figure 25) and a baseline coach (i.e., triage yellows until expire, then triage greens). Study 1 verified that **PSI-Coach agrees with human triage strategy prediction significantly more than a baseline coach.** In addition, PSI-Coach improves miss versus skip inference for subjects with strategy variations, PSI-Coach intervenes earlier in greater volume than human coaches, and humans intervene later (in the last few minutes) in greater volume than PSI-Coach.

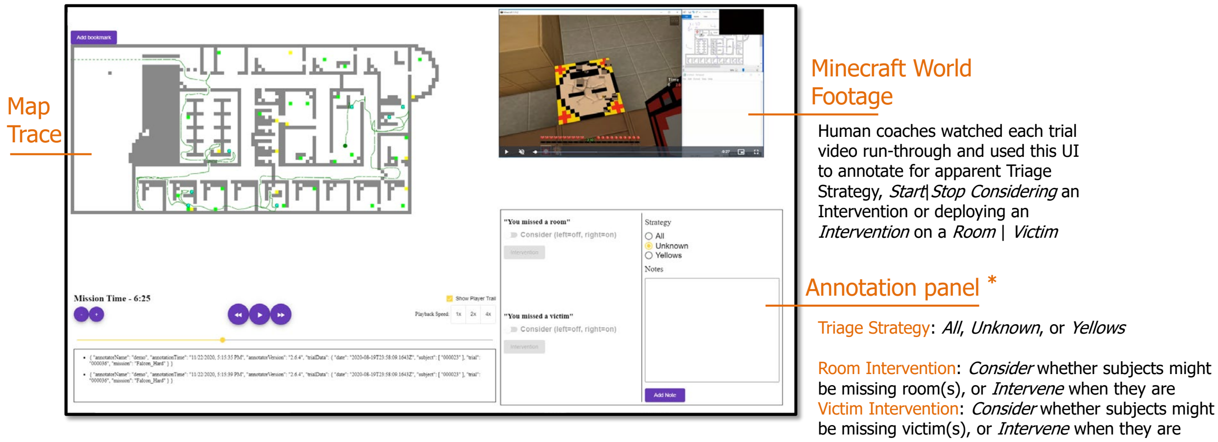


Figure 25: Comparing PSI-Coach and human annotators. * *This coding was not on "God mode," so a victim was only coded as missed if they were in the subject's FOV as represented by the in-world footage. This is the same for PSI-Coach's inferences.*

For judgements of "intervene" and "consider or intervene":

- PSI-Coach agrees more with Human Coaches than Baseline*
- PSI-Coach considers or intervenes more than both Baseline and Human Coaches*
- PSI-Coach misses less Human Coaches actions than Baseline*

For judgements of "considering", "consider then intervene", and "consider then does not intervene":

- PSI-Coach considers more than Human Coaches (not supported by evidence)
- PSI-Coach "considers then intervenes" less often (by percent) than Human Coaches*
- PSI-Coach "considers then does not intervene" more often (by percent) than Human Coaches*

* *Supported by evidence*

Figure 26 shows the results of human/baseline coach agreement on triage strategy changes over the course of the task.

Figure 26: Human/Baseline Coach agreement over time

There are three periods of low agreement across the time course of the task:

1. Low agreement in the first minute of the task came from different approaches to the cold start problem. Both PSI-Coach and the base coach assume that the subject begins with a yellows-only strategy, while human coaches waited for evidence, which took a varying amount of time by coach and subject.
2. Agreement dips between 360-260s, during the minute before and after the yellows expire. This is the period of highest instability for the subject's strategy (and, thus, for inferring it).
3. Since there is only one available valid strategy after the yellow victims expire at 300s, we would expect 100% agreement. However, two subjects continued to ignore greens and beeps indicating greens in favor of searching for yellows until ~250s. Approaches to inferring and coding this unexpected behavior varied across humans and agents, leading to less-than-100% agreement.

Figure 27 shows the results of relative agreement over time: PSI-Coach agrees with human consensus significantly more than baseline coach.

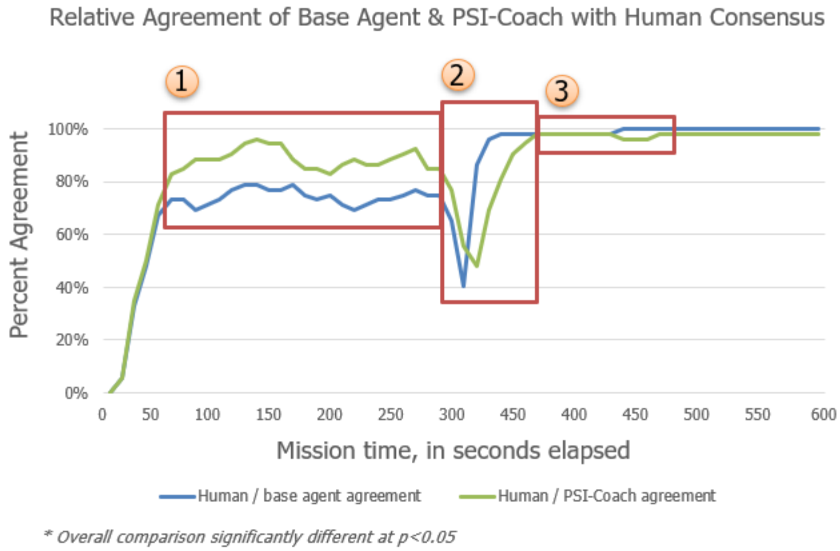


Figure 27: Relative PSI-Coach and Human agreement over time

There are three periods where the degree of agreement between PSI-Coach and human coaches, and between the base coach and human coaches, differ:

1. During the portion of the task when yellow victims can be triaged, PSI-Coach agreed with the human inference of subject strategy more than the base coach did (significant at the $p<0.05$ level overall).
2. PSI-Coach's detection of the change in strategy lags immediately after the 300s shift due to the fact that it infers subject strategy at room-departure events. We expect to expand this inference point to other event types for the experiment in March
3. Neither the human coaches nor the base coach had an accurate coding option for the two subjects who attempted to continue a yellows-only strategy after yellow victims expired; the humans improvised coding approaches. This produced less than 100% agreement despite the fact that there was only one valid strategy.

We identified subjects to review for deviations from a consistent yellows-first strategy by the percent agreement between the human consensus and the base coach, shown in Figure 28.

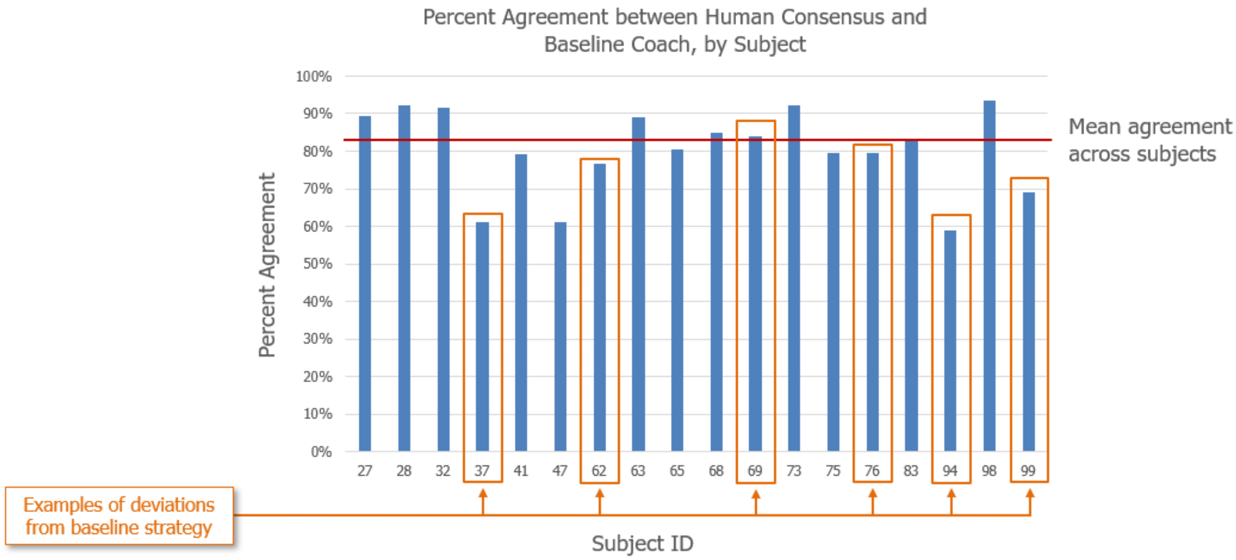


Figure 28: Percentage Agreement between human consensus and baseline coach, by subject

3.6.3.1 Results: Skipped versus Missed Victims

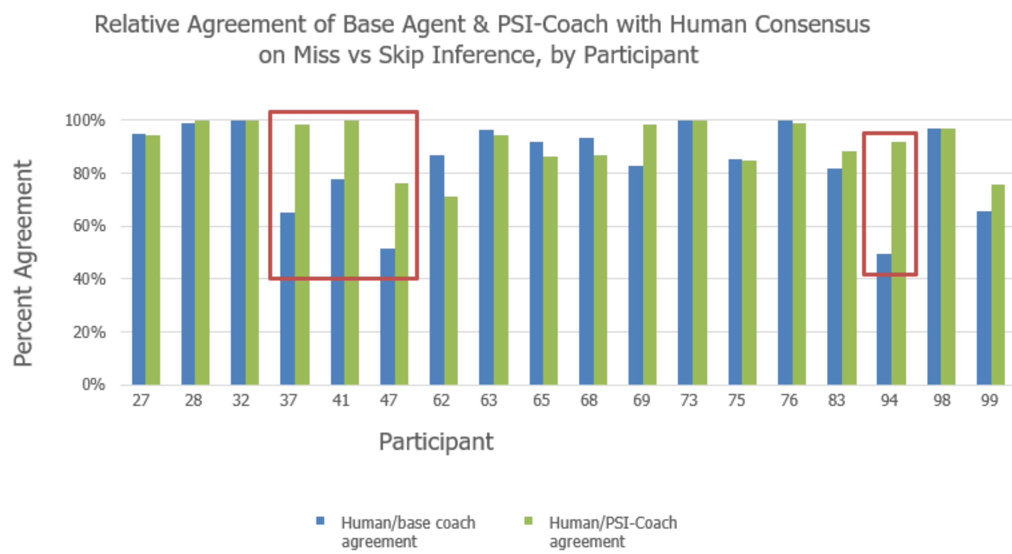


Figure 29: Relative agreement of base agent and PSI-Coach with human consensus on missed versus skipped victims. *PSI-Coach improved miss versus skipped inference for participants with strategy variations.*

The baseline coach infers whether a victim was skipped or missed inflexibly based on the assumed strategy for that time point (first 5mins yellow-only, green-only thereafter), so its inference of miss vs skip will not agree with human judgment on subjects who applied a notably different strategy. PSI-Coach continuously re-assesses the relative probability of a miss or a skip without rigid dependence on overall strategy, so it performs more in line with human judgment on subjects with varying strategies.

One limitation of this analysis is that most subjects applying a variant strategy were triaging victims opportunistically, which strongly decreased the number of bypassed victims in those subjects' data.

PSI-Coach and human coaches agree on missed green victims before yellow expiration.

Before the yellow expiration point, the participant entered the HerbaLife conference room and saved one out of three green victims before exiting.

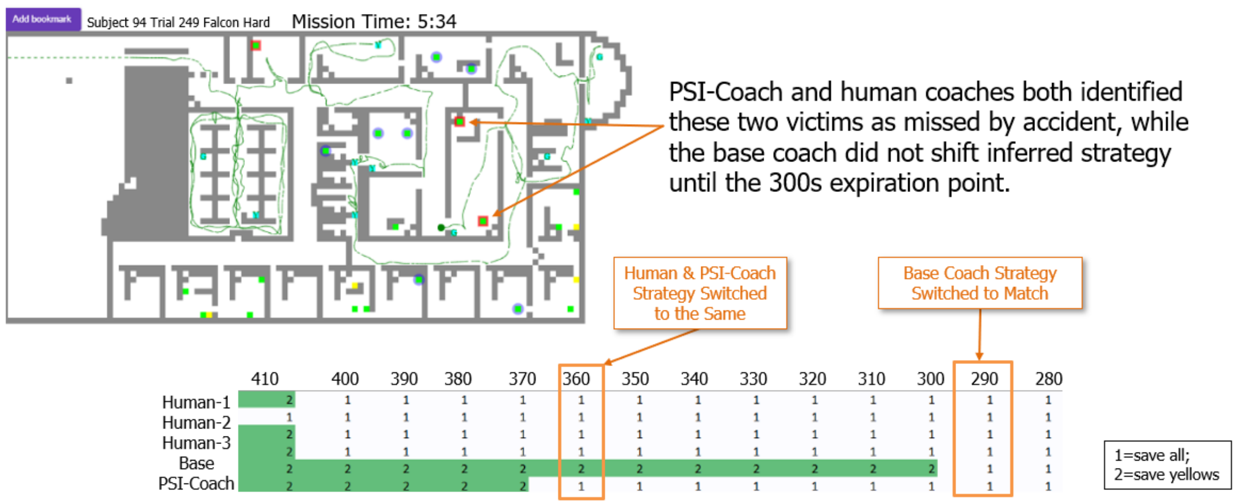


Figure 30: Graphics of skipped versus missed victims

Three human coaches coded:

- Participant strategy (Yellows, All, Unknown)
- Victims they believe the participant missed accidentally, in two levels
  - Consider: When they are considering an intervention on a specific victim they judge missed (Annotated as a period of time with a beginning and, if no intervention was chosen, an end)
  - Intervene: When they are sure enough to intervene and inform a participant that they missed a victim (Annotated as a single point)

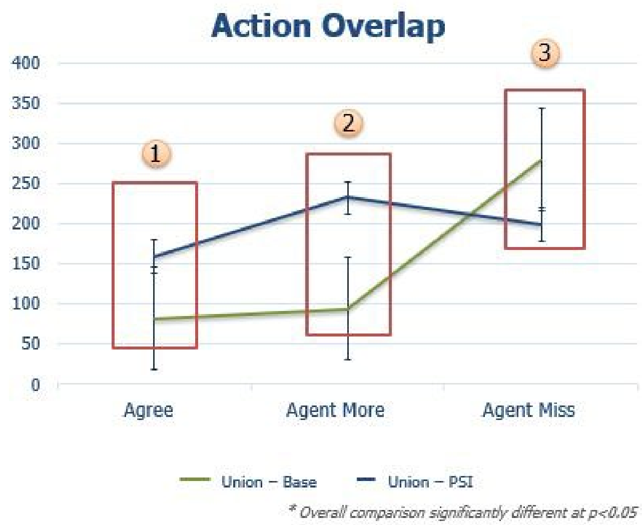
Instructions: "Use *consider* if you believe the participant may have missed a victim or may miss a victim in the future; and *intervene* if you believe they did miss a victim. Don't worry about whether or not the participant can act on your intervention, whether it would be useful or polite; consider primarily how strongly you believe they have actually missed the victim accidentally."

Practice: Human coaches annotated three trials from a different condition with each other using a think-aloud protocol, and noted policy decisions on edge cases or questions that arose. Out of 1,836 total victims (102 per participant across three trials), coaches considered intervening on 441 and intervened on 299.

	Consider	Intervene
Coach0	17% (312)	8% (152)
Coach1	1% (24)	4% (66)
Coach2	6% (105)	4% (81)

PSI-Coach agrees with Human Coaches significantly more and PSI-Coach misses significantly fewer Human Coach actions than does the Baseline Coach.

Figure 31: When both consider and intervene judgments are combined, Human Coaches agree more with PSI-Coach than with the Baseline Coach. *PSI-Coach acted more often when Human Coaches didn't than Baseline Coach. The Baseline Coach missed more Human Coaches actions than PSI-Coach did.*

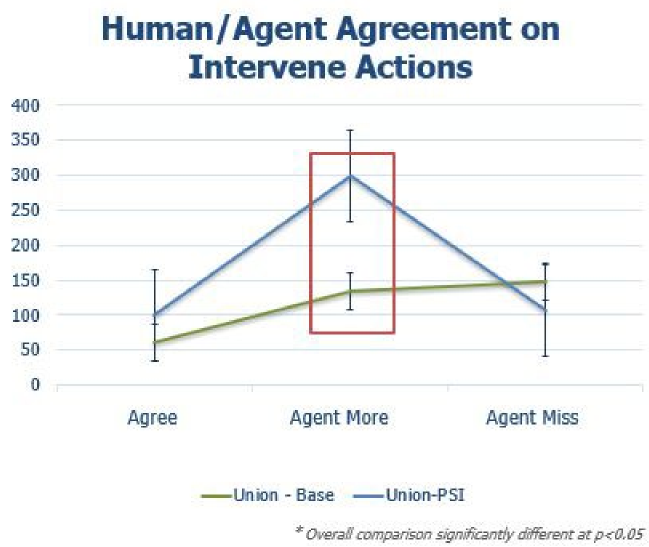


Agree: (PSI-Coach 27%, Baseline 18%, p<0.05)

Agent More: (PSI-Coach 39.2%, Baseline 20.6%, p<0.05)

Agent Miss: (PSI-Coach 55.4%, Baseline 77.2%, p<0.05)

Figure 32: When only considering interventions, the results are similar: there is a smaller difference between humans and PSI-Coach than between humans and the Baseline Coach.



Agree: (PSI-Coach 25.1%, Baseline 17.2%, p<0.05)

Agent Miss: (PSI-Coach 38.7%, Baseline 66.4%, p<0.05)

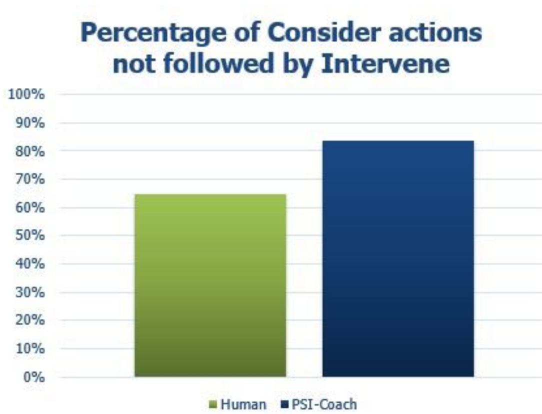


Figure 33: Intervene actions agreement

Overall takeaway: PSI-Coach intervenes less than humans after considering. Humans follow a consider annotation with an intervene marking twice as often as PSI-Coach. This can be altered by tuning PSI-Coach's conversion thresholds between no action and considering, and considering and action, but the performance implication of this ratio is unclear.

3.6.3.2 Results: PSI-Coach Considers then Deploys Interventions

PSI-Coach demonstrated increased agreement with human coaches' inferences and interventions relative to a non-socially-intelligent comparison agent (using heuristics and ground truth to infer and a greedy strategy to intervene). Inferences inform not just *when to intervene*, but also *when to stay out of the way and let the team do its job.* PSI-Coach combines weak signals (as human coaches must often do) to create islands of confidence on the need to intervene; inferences are designed to be intervention ready, and provide framing to give assistance in the context team's goals and strategies rather than idealized optimal strategy. It then combines multiple dimensions of team mental models to inform intervention decisions and infers continuously to facilitate timeliness of intervention and "wait and see" choices. PSI-Coach extends probabilistic programming and cognitive modeling to capture individual theory of mind without the assumption of rationality.

PSI-Coach considers intervention based on the subjects' strategy, victim bypass intent, and navigation plan. Below, it considers intervening on a missed "on accident" yellow victim, in a room that in is not on subject's navigation path.

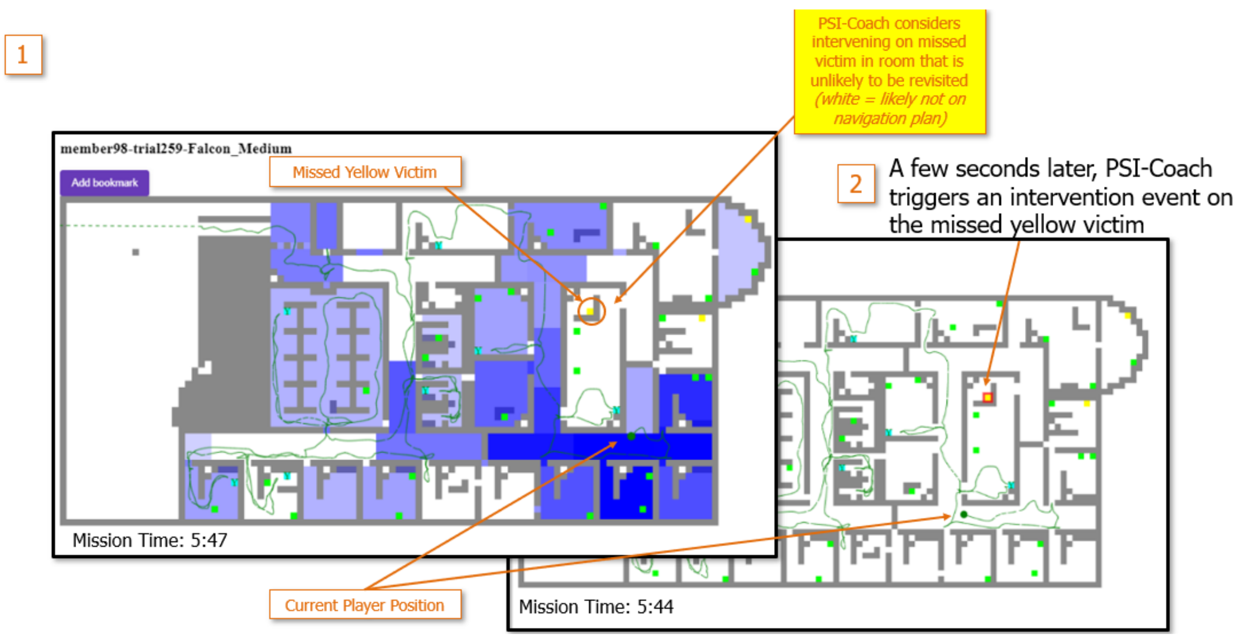


Figure 34: PSI-Coach considers intervening due to missed victim

3.6.3.3 Results: PSI-Coach Waits on Interventions to Best Match Subjects' Plans

In different contexts, PSI-Coach times interventions especially based on strategy, bypass intent, time urgency, and navigation plan. Below, PSI-Coach intervenes on 3 missed greens as soon is subject is nearby, which happen with 30 seconds left in the mission (Capability G).

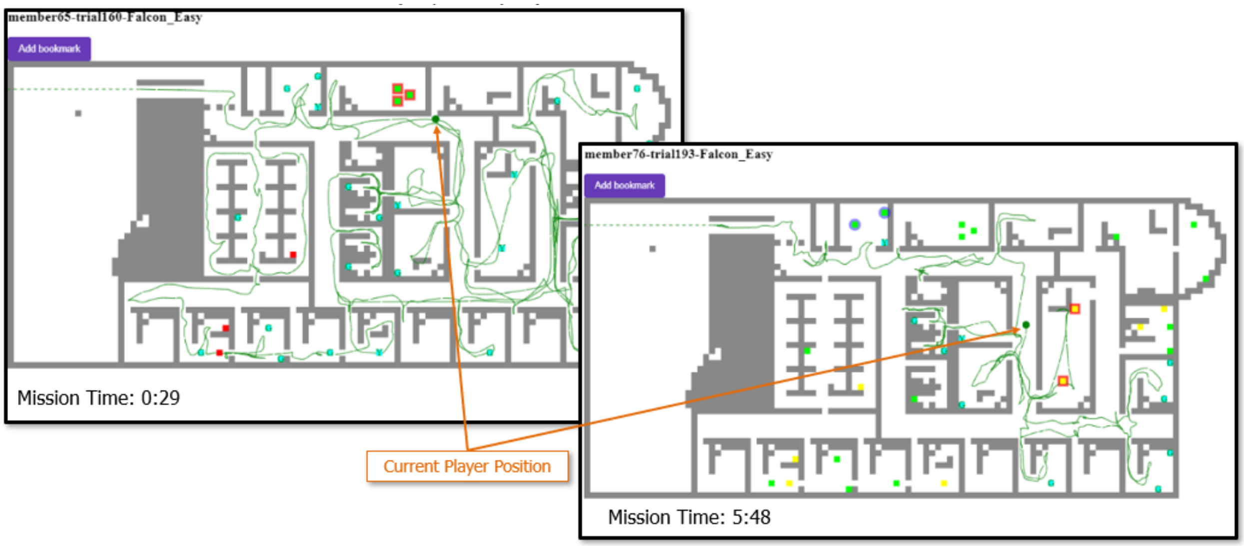


Figure 35: PSI-Coach matches interventions to subjects' plans. *PSI-Coach's Intervene triggers because subject is leaving the area with missed yellow victims (Capability F).*

3.6.3.4 Overall Results: Measures of Team Performance

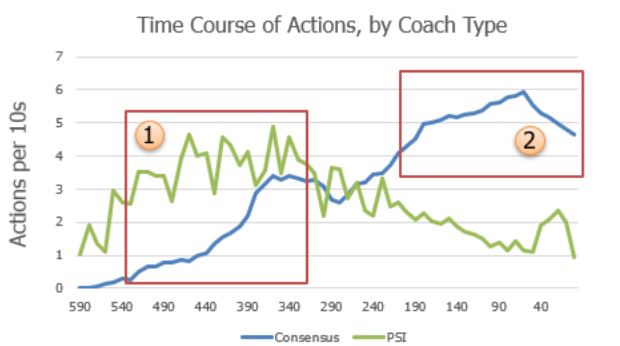


Figure 36: Mission time in seconds elapsed. *In preliminary analysis, actions were analyzed regardless of their targeted room(s) or victim(s).*

Preliminary analysis shows that PSI-Coach intervenes most in the 500-300s period in the task, when the most rooms and victims are available for consideration (that is, entered or seen but not yet triaged). By contrast, this is when humans act (either consider or intervene) least. Human coaches acted most in the closing 200s of the task, possibly due to decreased certainty criteria (lower cost of intervening), increased urgency on behalf of the subjects, or enough previous experience with a subject to raise confidence in the coaches' assessment.

PSI-Coach demonstrates the ability to automatically recognize what the team is doing, identify team process problems, and tailored interventions and timing of the interventions to improve team processes. As shown in Table 4, experiments demonstrated the ability to rapidly (within 60 seconds) improve team processes across diversity of teams: 28 advisor versus 28 no advisor trials:

- Improved Aligned-Team-Priorities by 60% (p=0.09)
- Improved Coordinated Team Communication by 13% (p=0.09)

Table 4: PSI-Coach vs. No-Advisor PSI-Coach improves team processes within ~60 seconds

Intervention Type	Team Process Improvement (p=0.09) within 60 seconds
Aligned Team Priorities: Prioritize Critical Victim	**60%**
Coordinated Communication: Victim Marker Following	**13%**

Table 5: Across multiple dimensions, PSI-Coach Agrees with human coaches more than a greedy agent does

Metric	Category	Definition	Human/ Greedy	Human/ PSI
PM1	Individual Mental Model	Individual Victim Prioritization Strategy	.410*	.220
PM2	Team Misalignment	Marker Block Semantics and Usage Misalignment	.458	.615*
PM3	Team Misalignment	Spatial Strategy Misalignment	.645	.520
PM4	Team Misalignment	Time Pressure Reaction Misalignment	.750	.802
PM5	Team Misalignment	Role Assignment Strategy Misalignment	.458	.625*
PM6	Intervention Point Identification	Victim Prioritization Strategy Intervention	.541	.770*
PM7	Intervention Point Identification	Marker Block Semantics and Usage Intervention	.291	.489*

Team research quality improves with:

1. **Multiple viable Team Strategies** from which teams can choose, switch between, and mix.
2. Teams that effectively use Team Strategies and apply **Team Processes** generate higher scores that are reliable, with statistically significant measurable differences. In other words, teams need to coordinate instead of apply individual skill (i.e., not run fast as fast as you can).
3. An agent can **observe indicators** of Team Strategies and Team Processes (i.e., don't rely upon voice comms for 90% of teams process indicators).
4. Interventions based on Team Processes and Team Strategies **make a difference** (players can apply suggestions, and when applied improve team process and therefore has a measurable different result).

3.7 Internal Pilot Studies

We supported or conducted pilot studies to generate data to validate and improve PSI-Coach. Charles River Analytics is not running an HSR, only supporting others. We supported pilot and full studies conducted by ASU and CMU; see approval status below. All Human Subjects Approvals were complete. Note that the New England IRB ruled Charles River use of data from ASU protocol exempt from review.

Institution	PI Name	Protocol Name	IRB Record ID	IRB Expiration	HRPO Record ID	HRPO Expiration
New England IRB	Krysta Chauncey	ASIST Evaluation – Phase 1	17-1272445-1	N/A (Ruled exempt)	ARL 20-083	N/A (Ruled exempt)
Carnegie Mellon IRB	Anita Woolley	Artificial and Collective Intelligence in Teams--Part 2	study2017_000000198	N/A; modification pending	N/A: In prep	N/A
Data Sharing Agreement	Bryan Loyall	ASIST Evaluation	N/A (Complete)	N/A (Complete)	N/A (Complete)	N/A (Complete)

We supported an initial observational study at CMU, by reviewing the plans and updating our Minecraft environment. CMU gathered data on 13 participants and common participant behavior and errors with a Gods-eye-view. We found that a participant believed they had searched the whole room, when they actually only searched half. Some navigational problems that occurred included: visiting the same room multiple times; missing a box; location uncertainty; aimless exploration; too much of a methodical approach that resulted in not getting to all blue boxes in time.

The experiment verified that data exists for potential PSI-Coach participant state inferences, including: rooms to be searched/not searched, Navigation Strategy, Triage Prioritization Strategy, Confusion, Frustration. We found that participants were missing knowledge about mapping mental models and not triaging victims successfully.

We defined key **performance metrics** and mental states to be used as input to our PSI-Coach Cognition Inverter, and we validated the Cognition Inverter's performance.

Figure 37: Participant revisited rooms multiple times; stuck in bathroom as a result of improper use of skill

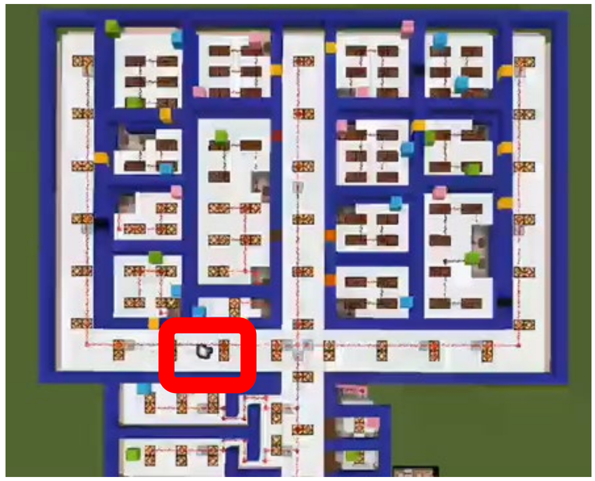


Figure 38: Participant with excellent spatial memory

Figure 39: Observer's God's-eye-view of Participant conducting USAR task

We conducted system testing on our Minecraft environment to rapidly fix bugs, iterate the environment features, and inform pilot study experimentation protocols. We ran 22 systems tests (approximately 30 hours) with a first-person view, as shown in Figure 40. Figure 41 shows the participants' confidence levels over time.

Figure 40: First-person-view of a participant using a "save-critical-victims-first" prioritization strategy

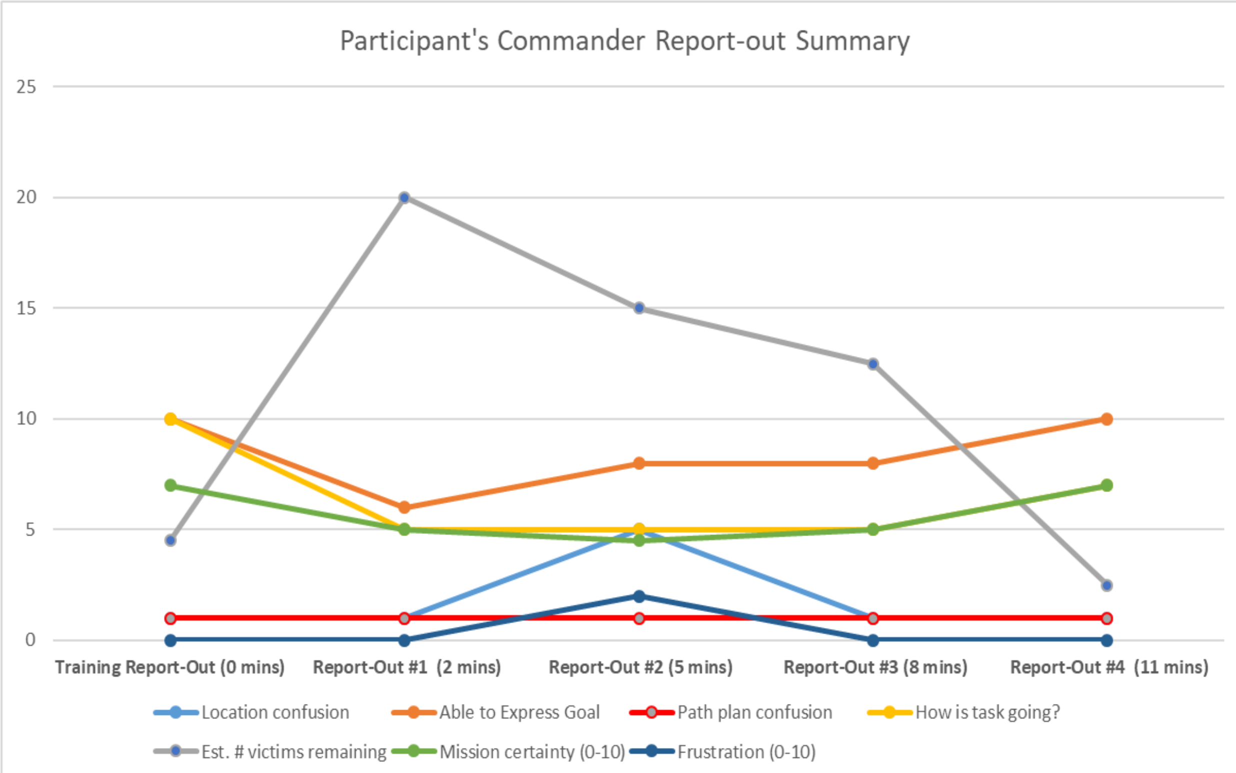


Figure 41: Plot of a participant's mental state over four commander report-outs. *Gray line shows a human participant's estimates of the remaining victims. These become more accurate as the participant's mission certainty increases (green line); the growth in the participants confidence is aligned with a big estimation accuracy jump (drop in gray line that matches the ground truth of two remaining victims at the final report-out).*

Our system test runs confirmed that participants display individual differences relevant to a Search & Rescue task in Minecraft. These tests also validated that individual differences can be observed and potentially predicted by PSI-Coach, and predictions will be relevant to effective coaching (improving search strategy versus predicting the next victim participant will save).

We shared our system test results with ASU's SAR Minecraft developer to jointly improve features. We also routinely shared results at weekly Experimental Working group meetings. We also provided data for CMU-TA2 and ASU pilot study protocols, and posted relevant data for program-wide use. We led the Incident Commander, Self-Report Questions Fast-Track and SAR Task Complexity Fast-Track breakout groups.

We received full IRB, HRPO, and DSA approval to use TA3 protocol data, and we worked with IHMC and IST to outline a behavioral coding schema to serve as ground truth. We ran qualitative review of pilot data and in-world video from TA3 for behavioral expectations and insights.

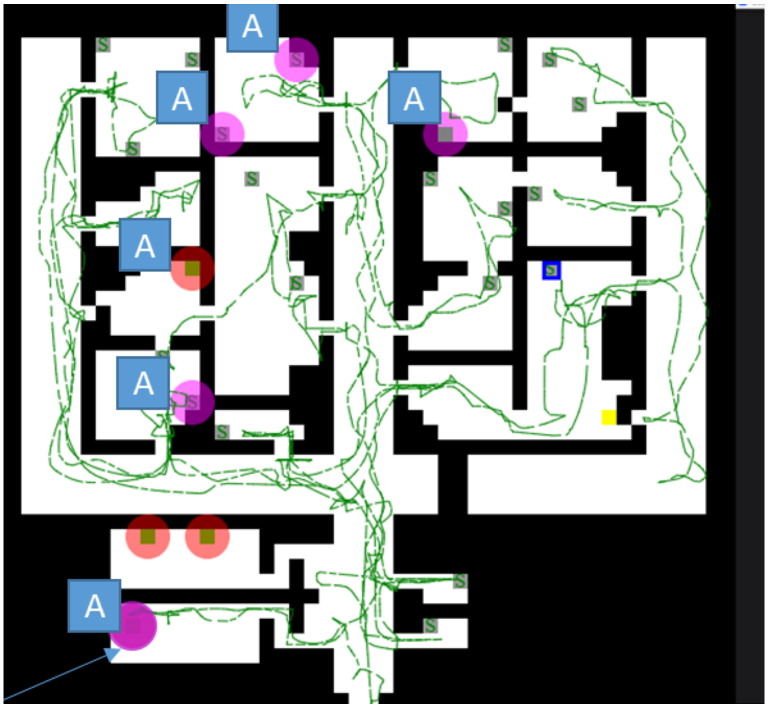


Figure 42: PSI-Coach Capability A testing: 5 victims skipped (purple halos) and 1 victim missed (red halo)

To measure PSI-Coach performance, we compared the Cognition Inverter's inferred/predicted values with that of a human expert using post experiment annotation. Figure 43 shows a concept map of this comparison.

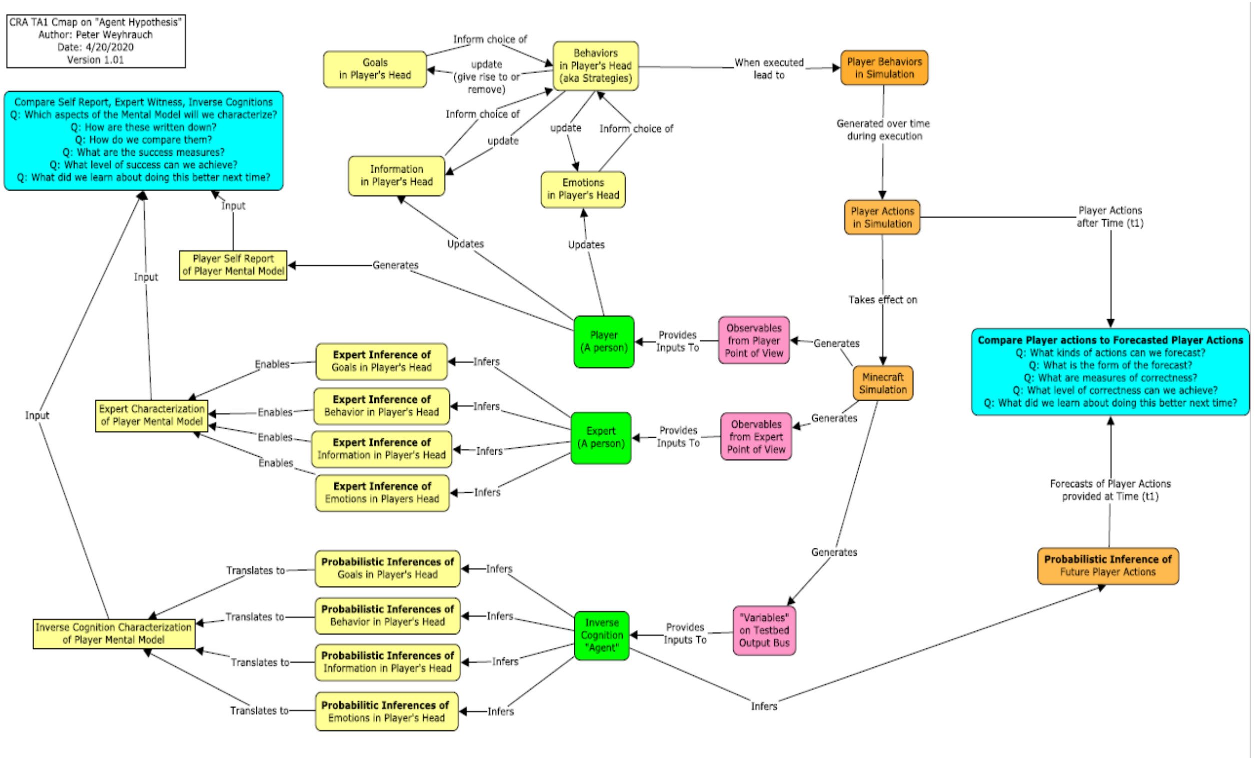


Figure 43: Concept map of PSI-Coach's capability performance measured against a human expert's ability to interpret the observable behavior from participants' goals and intentions

We created an annotation tool to capture human participants' goals and intentions. Offline, this tool annotates participants' goals and intentions by reviewing pre-recorded video synced to a 2D map with the current player location and trail. As shown in Figure 44, the annotator marks if a victim was skipped on purpose (purple) or skipped by accident (red). We shared this code base for program-wide use to help TA3 annotation and other TA1 and TA2 visualization and data annotations.

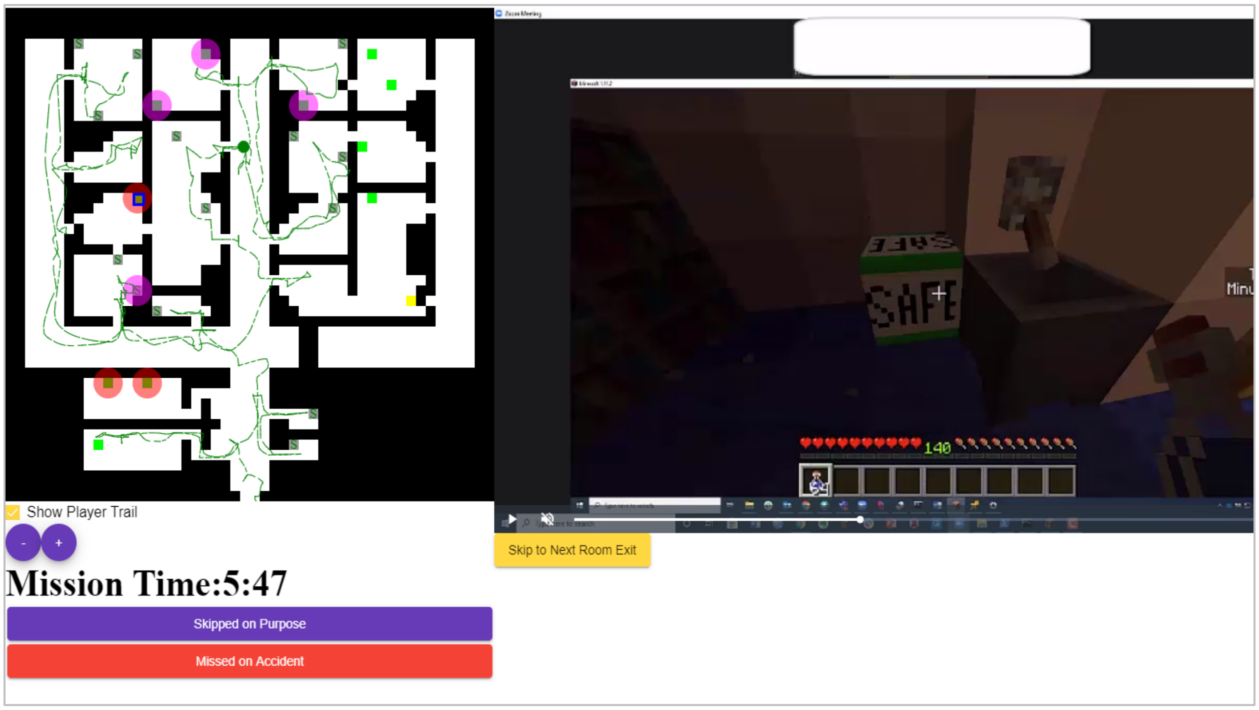


Figure 44: Human behavior annotation encoding tool used to capture human intentions and goals (as determined by an offline human observer)

3.7.1 *Mission Complexity for MToM Research*

We generated guidelines for increasing mission layout complexity to increase observable SAR strategies and participants' goals based on pilot testing and hackathon data review, as shown in Figure 45. We worked with TA3 ASU to create three mission complexities by manipulating blockages, wall openings, victim layouts, and occlusions. We also factored in navigation strategy analysis from IHMC and workload/complexity correlations from UCF's hackathon data analysis.

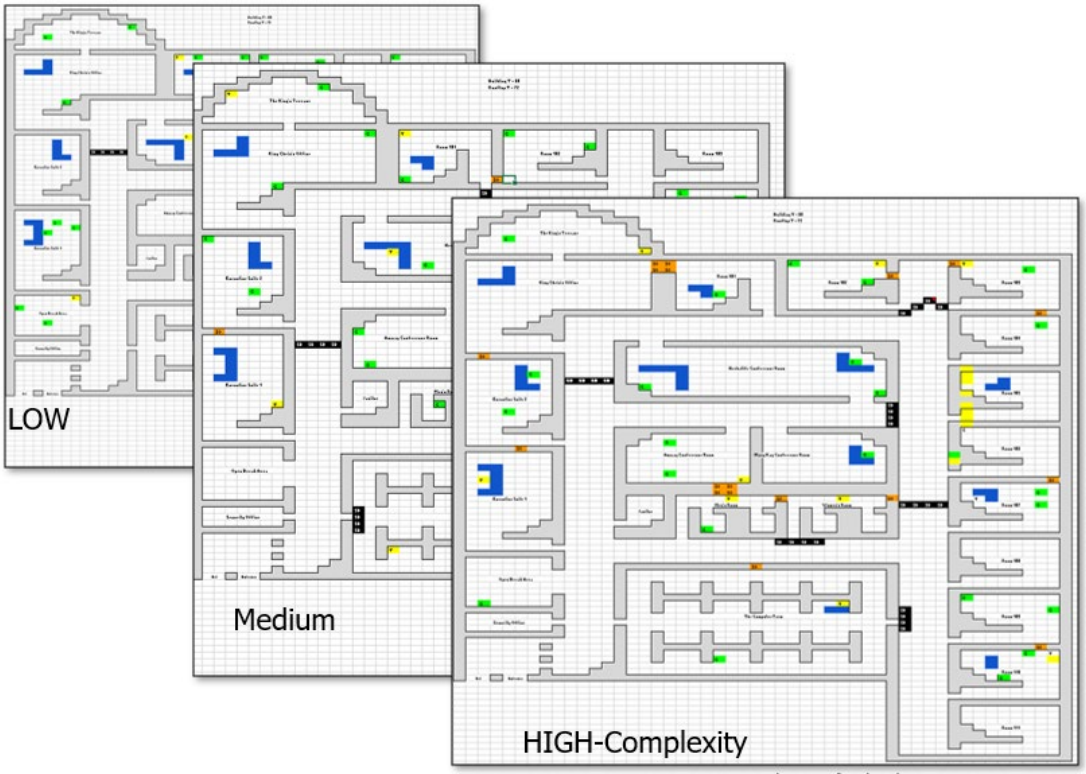


Figure 45: Three levels of mission layout complexity. *We generated these complexity options in close collaboration with TA3 ASU, as defined by the experimental goals and constraints specified in the Falcon Complexity Fast-Track breakout group meeting.*

Charles River Analytics supported CMU's theory of collective intelligence pilot study to explore emotion, coverage, use of skill, and task accuracy. We performed an initial review of ASU's Human Subject Research (HSR) data to prepare for experiment execution. Our team tuned PSI-Coach models on sample data (Dynamic Map condition) and engineering data, and quantitatively tested on hold out data set (TriageSignal-StaticMap condition). We expanded the number of Charles River staff with CITI certification to expedite development and testing.

We created inference visualizations to refine PSI-Coach Capabilities, as shown in Figure 46.

- 10 Map Overlays (*right: Capability D highlights bypassed victims on the map*)
- 7 Inference Graphs (*right: Victim Triage Strategy shows subject is switching*)

Figure 46: PSI-Coach inference visualizations

For the March 2021 experiment, we collaborated with the CMU-TA2 team comparing a participant's understanding of their beliefs, intentions, desires, and strategies with an observer's. In close collaboration with CMU-TA2, we created interview guide, study protocol, and other assets needed for IRB approval of this study at both Charles River and CMU. We worked with Charles River and CMU IRBs to define needs for a single-site IRB protocol.

The team updated the Minecraft world file to add CMU pilot study requirements, such as player's trail on maps to be used as an intervention.

We made the following updates to the Data Abstractor to support the experiments:

- Added support for new events from testbed: RoleSelected, RubbleDestroyed, ToolDepleted, ToolUsed, VictimPickedUp, VictimPlaced
- Added entity tracking for victims that are moved
- Added logic to update graph connectivity as rubble is removed
- Added new secondary event, PlayerAppears, that triggers whenever one player appears in another player's field-of-view
- Added support for gravity-sensitive stacked rubble and victims hidden under rubble
- Updated block occupiability logic to account for new experimental constraints on players, such as no jumping

In addition, we updated testing support for the experiment:

- Enabled dockerization of PSI-Coach and deployed it on TA3 testbed
- Updated local testbed with updates from TA3 development testbed
- Updated agent to support various iterations of Saturn map data published before, during, and after Hackathon

3.7.2 Study Results, Findings, and Stories

Highlights from each of the three studies are described in the subsections below. Section 3.7.3 details the results for each Capability (A–G).

3.7.2.1 Study 1

Summary Findings. The team qualitatively assessed all seven PSI-Coach Capabilities (A–G) and found that PSI-Coach was particularly strong at inferring subjects' field of view, **victim triage prioritization strategy** (plan), skipped/missed victim bypasses (intent), and navigation strategy including missed rooms. PSI-Coach was often delayed in recognizing strategy switches and needs work on inferring device usage.

The team also quantitatively tested PSI-Coach Capabilities, comparing to human coach annotations:

- Human observers "coaches" annotated 52 trials from 18 subjects (*TriageSignal-StaticMap* condition)
- Human coaches annotated a few elements that are foundational to all PSI-Coach capabilities:
    - Key ToM elements (*Triage Strategy* and *Victim-Bypass Intention*), which are fundamental to PSI-Coach Capabilities A–G
    - *Intervention* decisions (or if human coaches were *considering* to intervene on *Victim* or *Room*), which is the aggregate, and ultimate design goal, of all of the PSI-Coach Capabilities A–G
- Human coach annotators agree enough to calculate a straightforward **consensus coach** used to compare Human-annotated ToM Elements to PSI-Coach ToM elements that are all used in Capabilities A–G
- Human coaches intervened at much higher rates toward the end of the task (final 2–3 minutes), also with a small spike before yellow victims expire

Preliminary Results. PSI-Coach identification of variation in subjects' strategy is **closer to human coach identification** than that of a baseline coach (significant at $p < 0.05$). Baseline coach (i.e., first 5 minutes yellow-only, green-only thereafter) is correct in the majority of cases (~80%), but both PSI-Coach and human coaches identify novel variations of strategy missed by the baseline coach. In addition, PSI-Coach can map complex sequences of behavior (including multiple weak signals) to subjects' intention, such as correctly inferring victims that subjects missed on accident, in line with human annotation.

SAR Task Results and Observations:

- Relatively little **use of strategy** (e.g., 20% of TriageSignal-staticMap conditions varied in Victim Prioritization Triage Strategy) and, based on qualitative observation, the use of strategy seemed to have little effect on score:
    - Second half "2nd game" (after the yellow victims expire) negligible triage strategy to infer and intervene on
    - Task Design: High-quality interventions likely have little effect on player performance (likely same as player skill differences

- Human annotation of 52 trials showed **sparse intervention points**, even with the lower bar of "considering" an intervention; good interventions are rare
- Many subjects, especially high-skilled Minecraft players, had problems **using the device** effectively (device alerts late if sprinting, can miss doorways, a few device beep-type errors, participants unclear on device limitations—e.g., that it won't beep at a doorway if no victims in that room but there are in a room accessible from that room)

Suggestions on SAR Task Design:

- Ensure **strategy is used throughout the task**, e.g., yellows expire at variable times, introduce marking strategies
- Ensure good use of at least one strategy materially improves performance to enable **effective interventions**
- Emphasize **knowledge-oriented behavior patterns** (e.g., inform subjects that Easy map does not have occluded victims behind obstacles) rather than Minecraft skill effects (e.g., faster movement searches more area)
- Consider changing the relative **point reward or triage cost** (e.g., yellows are 100 points) to increase intervention effectiveness

Suggestions on Intervention Design:

- Annotate world or the map with subject's navigation history or the device's sensing history
- Prompt subject of their navigation history to highlight missed areas of the building
- Remind subject of the yellow/green reward comparison, especially if yellows are 100 points

Methods: Comparing Human Coaches, Baseline Coach, and PSI-Coach

- Human Coaches:
    - Watched in-world video and time-linked map trace of 18 subjects playing 3 trials each of the *TriageSignal-StaticMap* condition of the Study 1 Minecraft SAR task
    - Marked subjects' triage strategy (yellow, all, unknown), victim-bypassing intent (Skipped vs Missed), and coach's intervention decision (Start|Stop Considering Intervention or Intervene on Victim or Room) at times chosen by the human coach
    - Three human coaches annotated all trials to limit the impact of individual variation in annotation timing and content
- PSI-Coach Inferences and Intervention Decisions:
    - PSI-Coach agent processes *.metadata* file generated by TA3 testbed for subject in Study 1 Minecraft SAR task
    - Processed 18 human subjects performing 3 trials each (did not include trial 90 for subject #41 due to .metadata errors AND trial 23, which was a duplicate .metadata file for subject #99)
    - PSI-Coach does not know that yellow victims expire in 5 minutes for inferring triage strategy

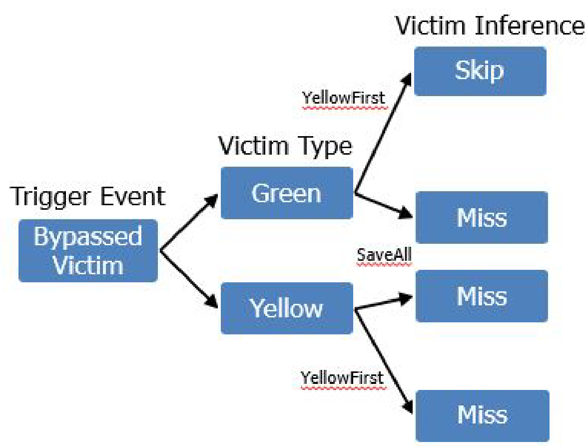
- PSI-Coach does know victim placement and type, just like our human coaches
- Inferred subjects' triage strategy (yellow, all) and victim-bypassing intent (Skipped vs Missed)
- Generated an intervention decision (Start|Stop Considering Intervention or Intervene on Victim or Room)

- Baseline Coach:
    - Default logic for subjects' triage strategy (yellow, all), victim-bypassing intent (Skipped vs Missed), and coach's intervention decision (Start|Stop Considering Intervention or Intervene on Victim or Room) based on subjects' *.metadata files*
    - Used as common comparison point to asses magnitudes of human coaches and PSI-Coach differences

Methods: Inference and Comparison

We created a "consensus coach" as a summary of the three human coaches by reproducing annotations consistent across 2 of 3 human coaches in each time point (in the case of the strategy analysis) or event (in the case of the miss vs skip and consider/intervene analysis).

To limit repetitive annotation requests to human coaches and the poor-quality annotations they would elicit, we asked coaches to "consider" or "intervene" on specific victims or rooms that they believed the subject had missed (instead of a more direct missed vs skipped judgment at each bypassed victim). For other victims, miss vs skip was inferred from a victim's green/yellow status, and the human coach's annotation of the subject's strategy based on the logic shown at right.



Agreement between pairs of coaches was calculated by comparing the judgment on a time point or event between the consensus coach and the base coach, and between the consensus coach and PSI-Coach. These two agreement scores were tested for significant differences using Student's t-tests.

***Results:* Victim Triage Prioritization Strategy Established Human Coach Consensus and Baseline Coach**

- Human Coaches:
    - Human coaches could annotate apparent switches in victim triage prioritization strategies at any time
    - 20% of subjects diverged from a basic yellows-first strategy
    - Human coaches differed in their assessment of behavior of strategy differences and strategy durations, but were largely in agreement in strategy type and timing
    - An example trial with 3 human coach strategy annotation is shown below 0=*Unknown*, 1=*All* , 2=*Yellows*

- Consensus of the human coaches is determined by 2 out of 3 agreement
- Baseline Coach:
  - The baseline coach (depicted above in the orange highlight, where 1= *All,* 2=*Yellows*) applied the reasonable triage strategy default logic that a subject would only save **yellows for the first 5:00 mins** and then only save green victims
  - This simple default strategy agreed with human coaches much of the time (81.9%) but identified interesting differences in strategies in subjects for further investigation.

PSI-Coach infers victim prioritization triage strategy, as shown below in an example trial.

Figure 47: Example trial triage strategy results

- As shown in Figure 47, by 6:50mins mission countdown timer, PSI-Coach infers three key changes in the Subject's Victim Prioritization Strategy
  - (1) Subject starts out with a Yellow-first Strategy, with some significant uncertainty
  - (2) At 8:55mins, Subject switches to a Save ALL Strategy, and at 8:05mins, Subject switches back to SAVE Yellow, both with high certainty
  - (3) At 7:10mins, PSI-Coach is uncertain if Subject temporarily switches back to SAVE ALL, and then at 6:10mims, Subject switches with high-certainty back to Yellow-only Strategy

From 4:55mins, Subject switches to a SAVE ALL Strategy for the remainder of the mission, intermittingly conducting verification sub searches

There are three periods of low agreement across the time course of the task:

1. Low agreement in the first minute of the task came from different approaches to the cold start problem. Both PSI-Coach and the base coach assume that the subject begins with a yellows-only strategy, while human coaches waited for evidence, which took a varying amount of time by coach and subject.

2. Agreement dips between 360–260s, during the minute before and after the yellows expire. This is the period of highest instability for the subject's strategy (and, thus, for inferring it).
3. Since there is only one available valid strategy after the yellow victims expire at 300s, we would expect 100% agreement. However, two subjects continued to ignore greens and beeps indicating greens in favor of searching for yellows until ~250s. Approaches to inferring and coding this unexpected behavior varied across humans and agents, leading to less-than-100% agreement.

And there are three periods where the degree of agreement between PSI-Coach and human coaches, and between the base coach and human coaches, differ:

1. During the portion of the task when yellow victims can be triaged, PSI-Coach agreed with the human inference of subject strategy more than the base coach did (significant at the $p<0.05$ level overall).
2. PSI-Coach's detection of the change in strategy lags immediately after the 300s shift due to the fact that it infers subject strategy at room-departure events. We expect to expand this inference point to other event types for the experiment in March
3. Neither the human coaches nor the base coach had an accurate coding option for the two subjects who attempted to continue a yellows-only strategy after yellow victims expired; the humans improvised coding approaches. This produced less than 100% agreement despite the fact that there was only one valid strategy.

The results showed that humans and PSI-Coach correctly identify ***saves greens "on the way" strategy.***

Figure 48: Several subjects followed high-value-victim-locating-device beeps but also triaged greens "on-the-way" who didn't require path deflection.

The human coaches didn't have a specific code for this—but each coach either flipped their assessed strategy repeatedly or made a note of it in their coding. PSI-Coach inference visualization also shows switches from *AllYellowNoGreen* Strategy to *SaveEverybody*, similar to human coaches.

The results also showed that humans and PSI-Coach correctly ***"early" switch to save everybody strategies***. Several subjects switched strategy before the yellows' time expired (or they had triaged all the yellow victims). Human coaches identified this switch correctly. PSI-Coach is often delayed in inferring switch (~20secs).

In addition, humans and PSI-Coach correctly identify ***interleaving strategies***. Several subjects switched strategy before the yellows' time expired (or they had triaged all the yellow victims). Human coaches and PSI-Coach correctly identified every switch type and duration. On the last switch to *SaveEverybody*, PSI-Coach's confidence was not high enough to switch back, but interleave still showed

And, finally, humans and PSI-Coach correctly identify ***"save whoever comes first" strategy***.

Skipped/Missed Inference Results. The baseline coach infers whether a victim was skipped or missed inflexibly based on the assumed strategy for that time point (First 5mins yellow-only, green-only thereafter), so its inference of miss vs skip will not agree with human judgment on subjects who applied a notably different strategy. PSI-Coach continuously re-assesses the relative probability of a miss or a skip without rigid dependence on overall strategy, so it performs more in line with human judgment on subjects with varying strategies.

One limitation of this analysis is that most subjects applying a variant strategy were triaging victims opportunistically, which strongly decreased the number of bypassed victims in those subjects' data.

PSI-Coach and human coaches agree on missed green victims before yellow expiration.

Figure 49: PSI-Coach and human coaches both identified these two victims as missed by accident, while the base coach did not shift inferred strategy until the 300s expiration point.

Intervention Decision Results. In Figure 50, PSI-Coach considers intervention based on the subjects' strategy, victim bypass intent, and navigation plan. Here, it considers intervening on a missed "on accident" yellow victim, in a room that in is not on subject's navigation path. A few seconds later, PSI-Coach triggers an intervention event on the missed yellow victim.

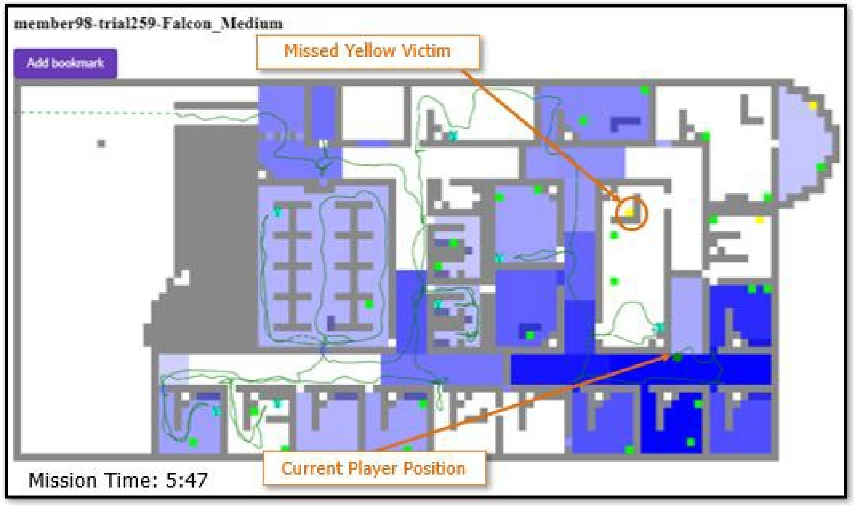


Figure 50: PSI-Coach considers intervening on missed victim in room that is unlikely to be revisited *(white = likely not on navigation plan)*

In different contexts, PSI-Coach times interventions especially based on strategy, bypass intent, time urgency, and navigation plan. Figure 51, for subject #65, PSI-Coach intervenes on 3 missed greens as soon as subject is nearby, which happens with 30 seconds left in the mission (Capability G).

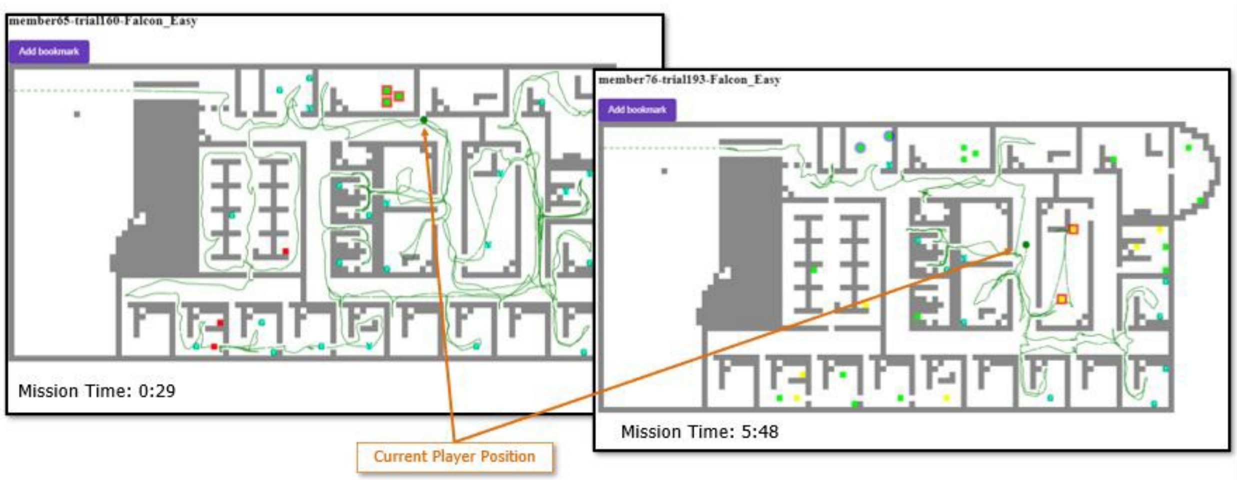


Figure 51: PSI-Coach's Intervene triggers because subject is leaving the area with missed yellow victims (Capability F)

Preliminary results regarding consider versus intervene show that PSI-Coach and human coaches act most during different halves of the task. Preliminary analysis shows that PSI-Coach intervenes most in the 500–300s period in the task, when the most rooms and victims are available for consideration (that is, entered or seen but not yet triaged). By contrast, this is when humans act (either consider or intervene) least. Human coaches acted most in the closing 200s of the task, possibly due to decreased certainty criteria (lower cost of intervening), increased urgency on behalf of the subjects, or enough previous experience with a subject to raise confidence in the coaches' assessment.

PSI-Coach intervenes most in the 500–300s period in the task (1) while human coaches act most in the closing 200s of the task (2). In preliminary analysis, actions were analyzed regardless of their targeted room(s) or victim(s).

Human bandwidth constraints do not apply to PSI-Coach, limiting direct comparisons. The selection of baseline and normalization is the primary challenge in this comparison, as human bandwidth limitations in tracking and marking multiple entities do not apply to PSI-Coach, which fundamentally skew the data. To address this, we will identify the set of PSI-Coach intervention points that human coaches also marked (either to consider or intervene) and that PSI-Coach has calculated as the highest certainty for further analysis. We are currently investigating methods to limit the impact of the human coaches' necessarily sparse annotations compared to PSI-Coach's continuous inferences, as well as the difficulty creating a consensus coach across such sparse data. One possible approach for the consensus coach is to use the single human coach with the greatest number of annotations. When analyzed without reference to intervention target, intervention rates normalized by the intervention frequencies for PSI-Coach and the human coaches respectively may clarify whether the time course difference seen in the preliminary investigation is real.

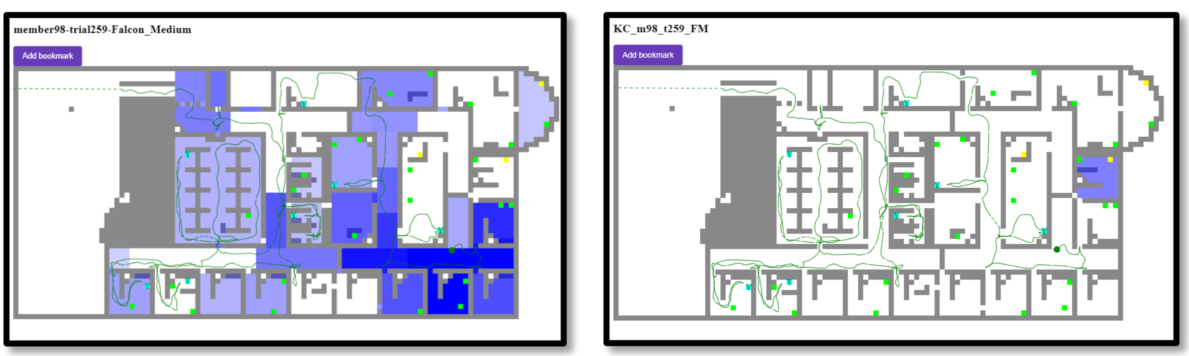


Figure 52: (Left) A typical "consider" annotation from PSI-Coach versus (right) a typical "consider" annotation from a human coach

Closing Task Design Insights:

- There are no valid alternative strategies available after yellows expire, so opportunities to infer a subject's **mental model are very limited for half of the data** collected on every subject.

- Map difficulty manipulation this round did not produce a clear decrease in scores; it was intended to increase confusion, but **confusion was not directly measured**.
- The technically expedient choice to allow **obstacles to remain in place in all maps**, even the Easy map, erased the difference between levels—although there were no victims near/behind obstacles in Easy, subjects didn't know this and thus spent time checking behind obstacles anyway.
- In order to infer knowledge of strategy from behavior, the task must involve a concrete strategy whose consistent application produces **discernible changes in behavior or performance**. Potential examples include:
    - Varying the length of lines of sight across conditions—possibly by placing victims on top of obstacles in Easy and behind obstacles in Hard—and training a subset of subjects on the placement of victims.
    - Remove victims (or block off a wing of the building) to allow subjects less time pressure, which they may use to revisit and verify triaged victims
    - Introduce a marking strategy for subjects to indicate rooms cleared of victims—this would create useful effects in the second half of the task, which is currently minimally informative on triage strategy
    - A perturbation where yellows do not always expire at 300s; or a "super medkit" can save red victims (expired yellow)
    - "Short cuts" in the Easy map, which require the same amount of time to traverse as the longer indirect path but simplify the spatial cognition required, could provide a contrast to the existing Hard map.

3.7.2.2 Study 2

Overview of Study 2 Findings. PSI-Coach demonstrated increased agreement with human coaches' inferences and interventions relative to a non-socially-intelligent comparison agent (using heuristics and ground truth to infer and a greedy strategy to intervene). For PSI-Coach, inferences inform not just *when to intervene*, but also *when to stay out of the way and let the team do its job.* The approach combines weak signals (as human coaches must often do) to create islands of confidence on the need to intervene. Inferences are designed to be intervention ready and provide framing to give assistance in the context team's goals and strategies rather than idealized optimal strategy. It combines multiple dimensions of team mental models to inform intervention decisions, and infers continuously to facilitate timeliness of intervention and "wait and see" choices. PSI-Coach extends probabilistic programming and cognitive modeling to capture individual theory of mind without the assumption of rationality.

Study 2 Target: *PSI-Coach continuously infers and combines individual-level mental models to identify team fissures and bolster team-level effectiveness.*

Metric	Category	Definition
PM1	Individual Mental Model	Individual Victim Prioritization Strategy
PM2	Team Misalignment	Marker Block Semantics and Usage Misalignment
PM3	Team Misalignment	Spatial Strategy Misalignment
PM4	Team Misalignment	Time Pressure Reaction Misalignment
PM5	Team Misalignment	Role Assignment Strategy Misalignment
PM6	Intervention Point Identification	Victim Prioritization Strategy Intervention
PM7	Intervention Point Identification	Marker Block Semantics and Usage Intervention
M1	Team performance	Predict team score at minutes 4, 9, and 14
M3	Mental model	Infer map information type at minutes 2, 7, and 12
M6	Mental model	Infer provided block meaning at minutes 3, 8, and 13
M7	Theory of mind	Predict whether P1 enters vs does not enter a room that P2 marks with a block meaning "regular victim here" or "no victim here"

Methods. The PSI-Coach agent has two key components. The first, the Cognition Inverter, recognizes goals, behaviors, and mental states from open world observations at both individual and team levels; in order to do this, the Cognition Inverter builds on recent advances in cognitive architectures for expressing human reasoning using machine learning based in Probabilistic Programming Languages (PPL). The second component, the Strategic Coach, projects a range of futures using Monte Carlo Tree Search to select the most effective and least annoying intervention, using interactive narrative research to guide sequences of action into productive channels.

Using Study 1 metrics of inferred strategy, we saw that PSI-Coach was able to infer individual theory of mind (ToM) with similar accuracy to humans (see Figure 53).

Figure 53: PSI-Coach agreement with humans (green) compared to a base agent compared with humans (blue) on an inference of player strategy

PSI-Coach was most effective where traditional approaches fail most, and it agreed significantly more with humans than a base agent did in the five participants where a heuristics-based approach was least effective.

In Study 2, PSI-Coach's Cognition Inverter expanded this individual inference to misalignment in the mental models of three-person teams. The Cognition Inverter again inferred the most likely specific strategy in victim prioritization techniques (see Table 6 for modeled strategies) for each of the three players.

Table 6: Individual player strategies inferred by Study 2 coders. As CV-interrupt was added after the final pilot run, it is not modeled by either agent.

Strategy	Definition	PSI	Humans	Greedy
Opportunistic	Player triages or moves victims of either type as long as they require minimal path deflection.	✓	✓	✓
CV-Interrupt	Player interrupts current activity to triage a critical victim but is otherwise opportunistic.		✓	
CV-First	Player ignores regular victims in favor of finding and triaging or moving critical victims.	✓	✓	✓
CV-Deferred	Player triages or moves regular victims but not critical victims	✓	✓	✓
Other	Player is applying a distinct, observable strategy not previously defined.	✓	✓	
Unknown	Coder cannot determine what strategy the player is applying.		✓	✓

Degree of alignment on the three-person team was modeled using an ensemble of distance metrics. Specifically, the *Bhattacharyya* coefficient, which measures the similarity of two probability distributions. This enables us to assess how closely two teammates shared the same strategies, given that Cognitive Inverter would produce probability distributions over possible strategies.

In order to understand how PSI-Coach's inferences and intervention points line up with existing options for social intelligence, we created two pairs of comparisons: first, we compared human coders to a heuristics-based inference model with a greedy approach to intervention; second, we compared those same human coders to PSI-Coach's inferences and interventions. Comparing the agreement between these two pairs allows us to understand how PSI-Coach improves upon current state-of-the-art social intelligence in reproducing human judgments, as well as where PSI-Coach differs from the human understanding.

For the human point of comparison, three coders watched each team's performance in the test trials simultaneously, using the in-world video and CRA's data visualization tool to show movement history and role (see Figure 54). Each coder was assigned a specific player to attend to for the duration of each trial. At minutes 13, 11, 9, 7, 5, 3, 1, and 0, coders paused the video, summarized the actions and intentions of each player, and completed the annotation visible in the right panel of Figure 54 to indicate their inference of each player's individual victim prioritization strategy, whether there were team-level intervention points on victim strategy or marker block usage, and whether there was misalignment between players on any of the coded dimensions (most crucially, marker blocks, spatial strategy, and role needs or strategy). Additional coding items seen in the Figure 54 annotation panel were collected to support later, qualitative data review and do not figure in the quantitative analysis.

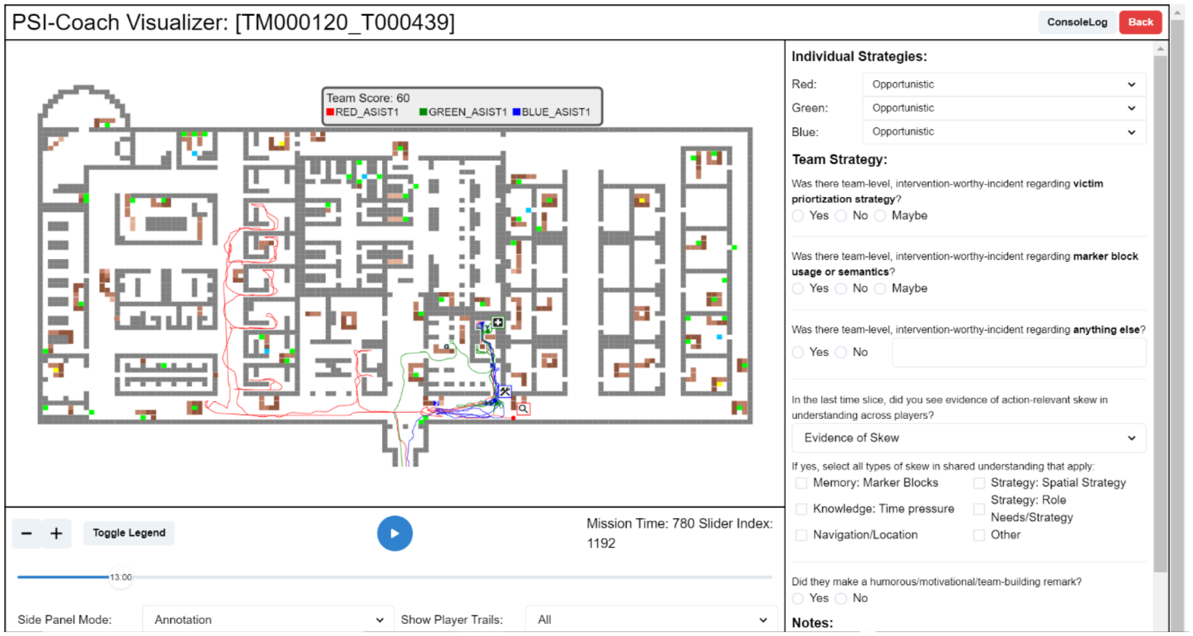


Figure 54: PSI-Coach visualizer, annotation view: Movement history and role

Results. Out of the eight metrics collected, PSI-Coach was able to exceed the greedy agent's degree of agreement with human coders on four: misalignment on marker blocks and role assignments, as well as the intervention points for victim prioritization strategy and marker block semantics and usage (see Table 7). On a fifth comparison, that of inferring individual-level victim prioritization strategy, the greedy agent agreed with humans more than PSI-Coach did, due to the high proportion of Opportunistic strategy inferences from humans intersecting with a higher proportion of Opportunistic inferences from the greedy agent than from PSI-Coach. All five of these comparisons were significant at $p<0.05$.

Table 7: PSI-Coach performance on Study 2 performer-specific metrics

Metric	Category	Definition	Human/ Greedy	Human/ PSI
PM1	Individual Mental Model	Individual Victim Prioritization Strategy	.410*	.220
PM2	**Team Misalignment**	**Marker Block Semantics and Usage Misalignment**	**.458**	**.615***
PM3	Team Misalignment	Spatial Strategy Misalignment	.645	.520
PM4	Team Misalignment	Time Pressure Reaction Misalignment	.750	.802
PM5	**Team Misalignment**	**Role Assignment Strategy Misalignment**	**.458**	**.625***
PM6	**Intervention Point Identification**	**Victim Prioritization Strategy Intervention**	**.541**	**.770***
PM7	**Intervention Point Identification**	**Marker Block Semantics and Usage Intervention**	**.291**	**.489***

*Statistical significance

On the inference of degree of marker block alignment (that is, whether all players were using the same semantic mapping for the types of marker blocks), PSI-Coach was more likely to agree with humans when they said there was no misalignment than the greedy agent was (see the dark blue component in Figure 55), and also less likely to identify misalignment when the human agent does not (see the dark gray cross-hatched component).

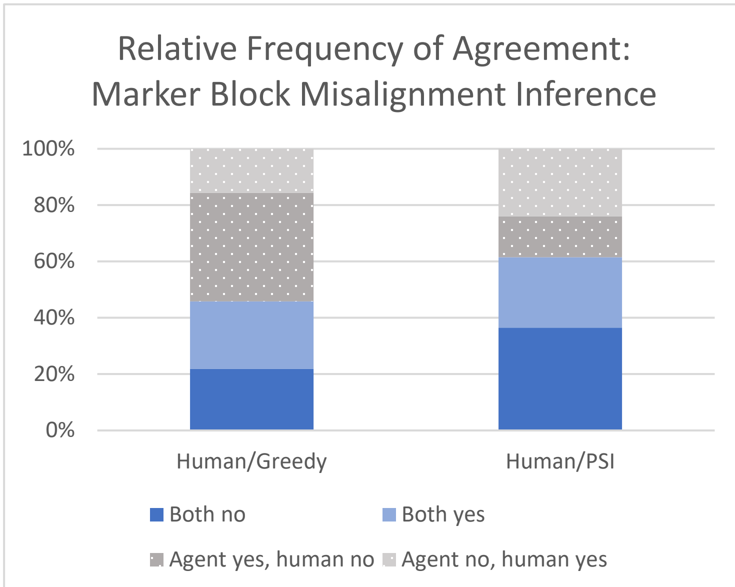


Figure 55: Relative frequency of agreement on inference of misalignment in marker block representations

On the identification of intervention points for marker block misalignment, again, PSI-Coach was more similar to humans than the greedy agent was (Figure 56). PSI-Coach was more likely to identify an intervention point overall than the greedy agent, resulting in more agreement with humans when they did identify an intervention point but also more disagreement with humans when they did not.

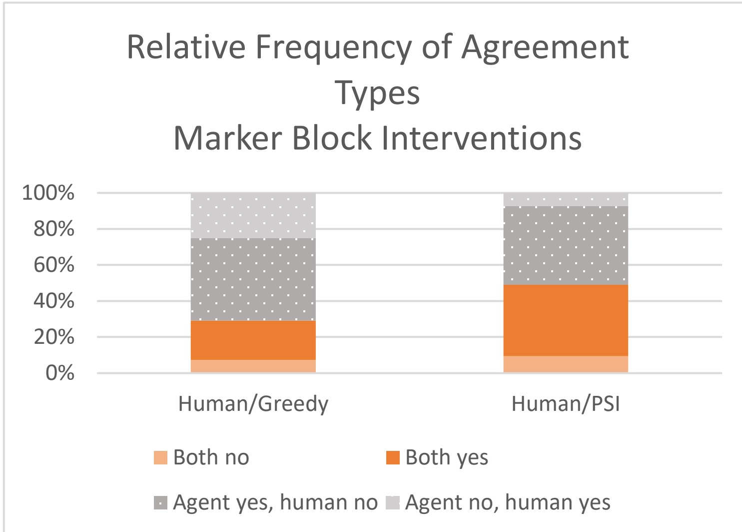


Figure 56: Relative frequency of agreement on intervention points for misalignment in marker block representations

Discussion. PSI-Coach has demonstrated increased agreement with human coaches' inferences and interventions relative to a non-socially-intelligent comparison agent (using heuristics and ground truth to infer mental models and a greedy strategy to identify intervention points). Such inferences inform not just when to intervene, but also when to stay out of the way and let the team do its job; these weak signals can be combined to create regions of confidence on the accuracy of the inference as well as the need to intervene. These inferences provide framing to identify the need for intervention in the context of the team's goals and strategies rather than an idealized optimal strategy, and are modeled continuously to facilitate the timeliness of interventions as well as "wait and see" choices.

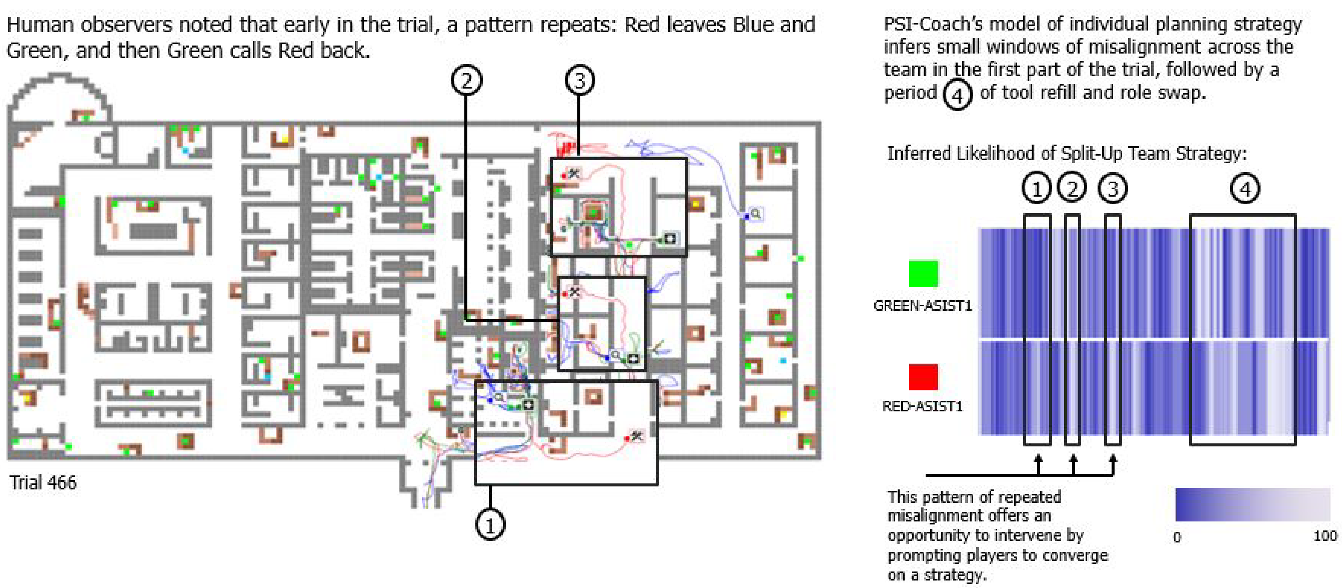


Figure 57: PSI-Coach automatically infers potential intervention moments from team strategy misalignment

Implications for Future Work. The inferences we modeled—of marker block semantics and intervention points—were prioritized based on their ability to undergird interventions. As ASIST moves forward into Study 3 and TA2 performers create Analytic Components to inform and improve the inferences and interventions of TA1 agents, PSI-Coach is well-positioned to integrate metrics from Analytic Components which meet the following generalizable criteria:

1. Can identify change points that indicate an intervention *online* during the performance of a task or trial (rather than retrospectively after its completion);
2. Can identify time periods of high certainty in either the correctness of the metric in representing the team's theory of mind, or the utility of the metric in identifying important intervention points;
3. Can identify the Theory of Mind representations of an individual, the coherence or incoherence of such representations across the team, as well as the task performance context of the team;
4. Can identify a causal relationship between the metric and performance on teamwork measures (so that intervening to change the metric may plausibly be expected to improve team performance).

3.7.2.3 Study 3

Overview of Study 3 Accomplishments. PSI-Coach Cognition Inverter individual and team models integrated Analytic Components for the following: (1) Collective Intelligence (CMU-TA2 TED and BEARD); (2) Joint Activity (IHMC); (3) Shared Knowledge in the Head/World (UCF); and (4) PSI-Coach team alignment (Charles River TA1).

Within our effort, we demonstrated the following with real-time team process models:

1. Roles and Abilities – *Appropriate use of specialized skills*, e.g., Role-specific skill use (slide 4)
2. Communication – *Alignment and misalignment of generation and consumption of knowledge-in-the-world*, e.g., marker block placement and usage (slide 3)
3. Coordination – *Alignment and misalignment of team goals and strategies*, e.g., victim prioritization strategy
   – *Enabling supporting behaviors via team situational awareness*, e.g., grouping behaviors
4. Motivation and Awareness – *Maintaining and building shared awareness of team processes*

PSI-Coach Strategic Coach reasoning combined weak signals of moment-by-moment AC outputs to intervene with just the right thing at just the right time. The system understands what the team is doing (M12, M13) and helps them do it better (M1-M3, CMU-TED measures), rather than joysticking. PSI-Coach focuses on team processes (CMU-TED measures, IHMC, UCF, misalignment AC scores) over task performance and infers not just when to intervene, but also when to stay out of the way and let the team do its job (PM1), build credibility, and improve uptake (M9).

Intervention Approach. Capabilities infer and predict ToM to identify potential interventions. Each includes information (e.g., ToM components and sets of weaker signals) that inform choice of intervening now versus benefits of wait-and-see.

Reasons to <u>intervene now</u>, reduce ineffective team processes, or take advantage of task opportunities

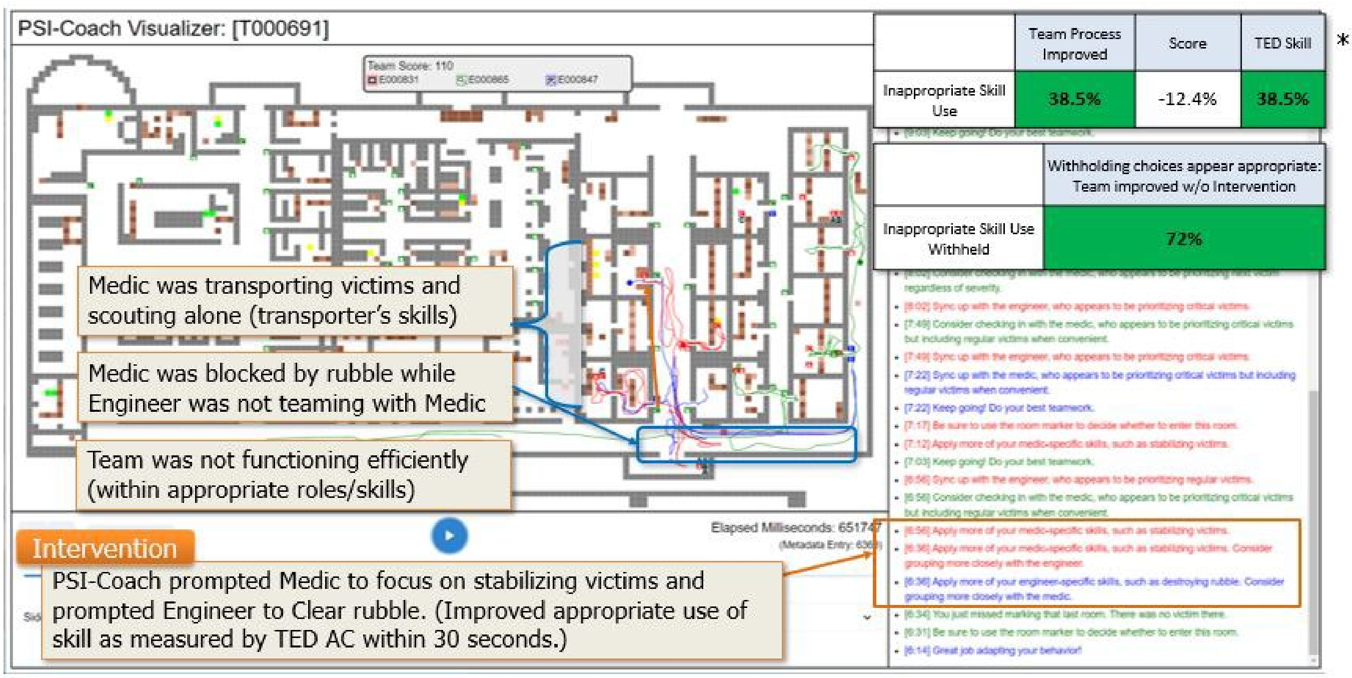
1. Recognize oversights and accidentally missed tasks and opportunities that are on strategy
2. Roles and abilities: Recognize ineffective use of specialized skills, e.g., not using role-specific actions
3. Communication: Recognize misaligned communication strategies, e.g., victim location sharing/use
4. Coordination: Recognize misaligned goal prioritization, e.g., CV/regular victim prioritization
5. Coordination: Recognize misaligned and inefficient supportive behaviors, e.g., grouping

Reasons to <u>wait-and-see</u>, avoid unnecessary interventions, or wait for better time

- Team may have skipped on purpose, or it will not be costly to address later
- Team is conducting supportive or collaborative behaviors to address potential gaps
- Communication is not currently critical to task accomplishment, time to fix later
- Goals are not currently interdependent, time to fix later before goals may become interdependent
- Supportive behaviors are not currently needed

MToT insights **combine** to choose both <u>when</u> & <u>how</u> to intervene.

Strategic Coach research extends these combinations with time-based reasoning, reasoning to minimize interventions, and reasoning about choices that affect communication efficacy.



* Comparison of when PSI-Coach intervened vs. No Advisor trial where PSI-Coach would have intervened

Figure 58: Demonstration of PSI-Coach prompt for appropriate use of skill

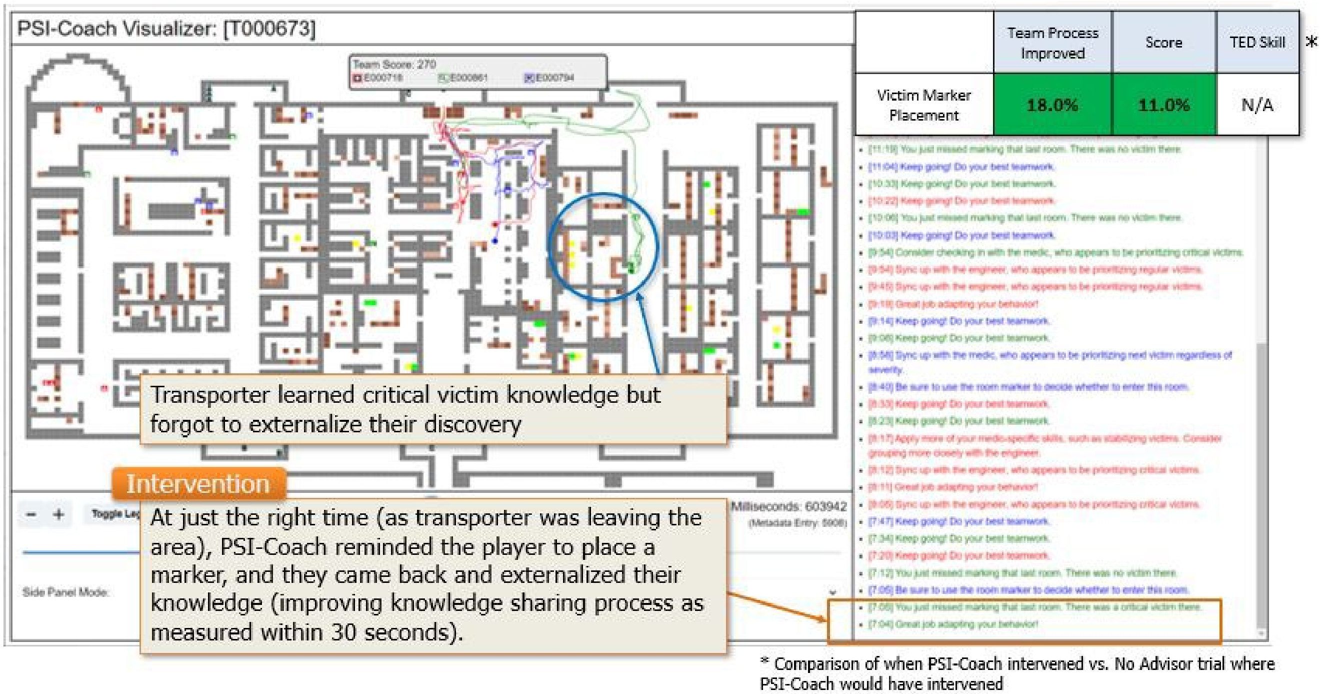


Figure 59: Demonstration of PSI-Coach prompt for knowledge sharing

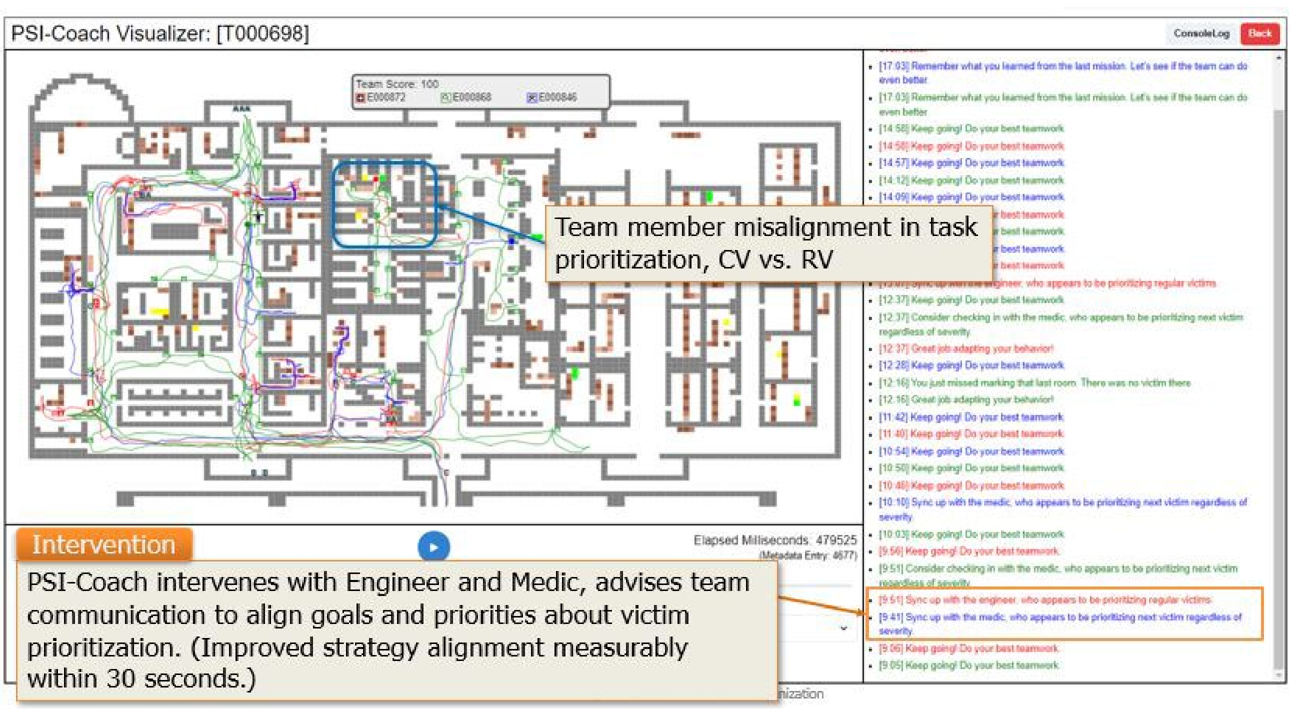


Figure 60: Demonstration of PSI-Coach prompt for team strategy alignment

3.7.3 Capability Results

3.7.3.1 Capability A: Infer Intentions from Subjects' Behavior

A: Recognize when user may be skipping victims, and whether they are skipping them accidentally or skipping them on purpose.

- Predict probability of a bypassed victim being intentionally **skipped "on purpose"** (weighted round blue highlight on a victim) or **missed "on accident"** (weighted square red highlight on a victim)
- Map shows 1 yellow victim missed on accident (8:05mins mission countdown timer), while the subject skipped 8 green victims on purpose
- Graph shows the running inference of **victim triage prioritization strategy** (AllYellowsNoGreens at this point), which is used to infer situated subject behavior such as leaving a room without triaging victims

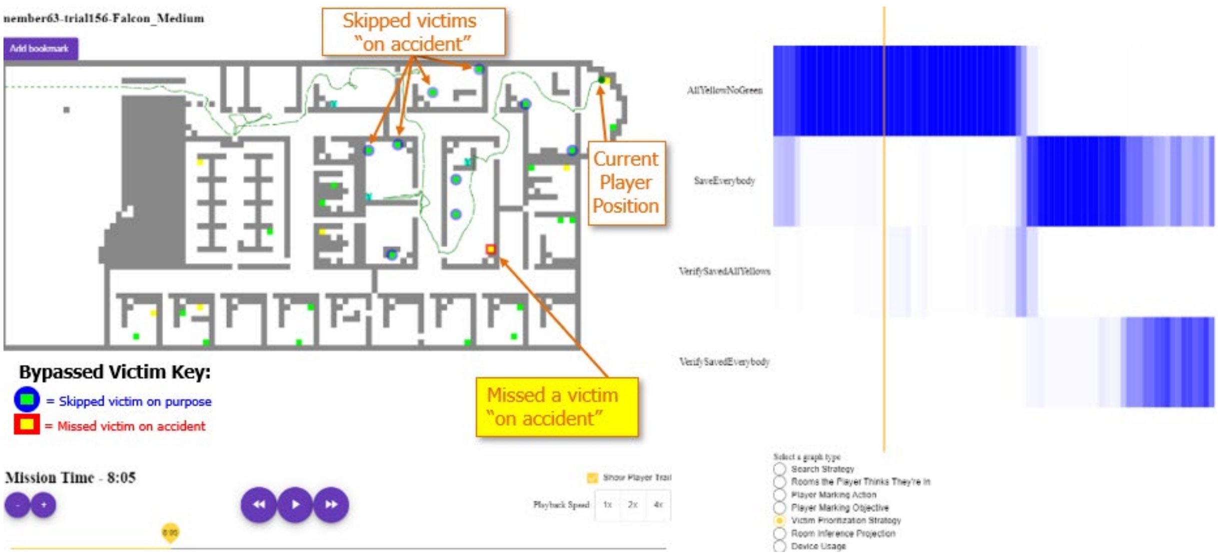


Figure 61: PSI-Coach inference of subject behavior

3.7.3.2 Capability B: Infer Gaps in Subject's Mental Map

B: Predict when a subject may be missing large portions of the building.

- Predict probability a subject may not be aware of an entire section of a building, even if empty it may have important passageways, in this case at 4:05mins on the mission countdown timer
- Map shows **missed room groups** (weighted blue highlight of room groups), while the subject is in the Terrace (far East side). Computer Farm is dark blue as most likely a missed area, lower West offices also missed but lower likelihood of actually being missed rooms
- Graph shows the running inference of **room inference projections**, which is used to infer likelihood subject plans to get to these rooms. Note: large re-trace of the explored parts of the building.

Figure 62: PSI-Coach inference of gaps in subject's mental map

3.7.3.3 Capability C: Infer Self-Correction of Navigation Gaps

C: Predict when a subject is missing a room and may stumble upon it before victims in the room expire.

- Predict probability a subject missed a room, but will likely return to the room before victims inside expire. In this case at 6:35mins countdown, subject missed rooms (with yellow victims that will expire soon), but subject can save in time
- Map shows **missed rooms likely to be revisited** (weighted blue room highlight). The subject doesn't show that they know about the rooms, but their search strategy shows a chance of visiting rooms before victims inside expire
- Graph shows the running inference of subject's **search strategy**. Area Planning means the subject will go from room-to-room (e.g., using holes) for shorter paths, while Hall Planning mean subject is more likely to anchor on hallways for orientation.

Figure 63: PSI-Coach inference of navigation gap self-correction

3.7.3.4 Capability D: Infer Self-Correction of Victim Bypasses

D: Predict when a subject bypassed a victim and may stumble upon them before the victim expires.

- Predict probability a subject missed a victim, but will likely return to the victim before they expire. In this case at 6:33mins countdown, subject bypassed 1 yellow and 19 green victims, yet subject has varying probability to return to each victim in time
- Map shows **bypassed victims** (weighted blue victim highlight). PSI-Coach predicts that the subject knows about the highlighted victims and assesses the likelihood of returning to save the victim before they expire
- Graph shows the running inference of subject's **search strategy**. This subject is predicted to have a preference for room-to-room Area Planning (e.g., using holes) for shorter paths – their search is likely to get to most but not all bypassed victims

Figure 64: PSI-Coach inference of victim bypass self-correction

3.7.3.5 Capability E: Infer Victim-Revisiting Navigation Plan

E: Predict when a subject bypassed a victim and when they will be on their path in the future.

- Predict probability a subject missed a victim, are far away now, but will likely be back in the area in the future. (indicating that there likely will be a better time to intervene) In this case at the 1:00min countdown mark, subject missed green (NOT highlighted, circled in orange), and not likely to have a navigation path that revisits this victim
- Map shows bypassed victims (weighted blue victim highlight). PSI-Coach predicts that the subject knows about the highlighted victims and assesses the likelihood of returning to save the victim
- Graph shows the running inference of subject's search strategy. At 1min, this subject has switched to prefer a Hall Planning navigation plan, making it less likely they will use the cut-through room and find the missed green in Herbalife conference room

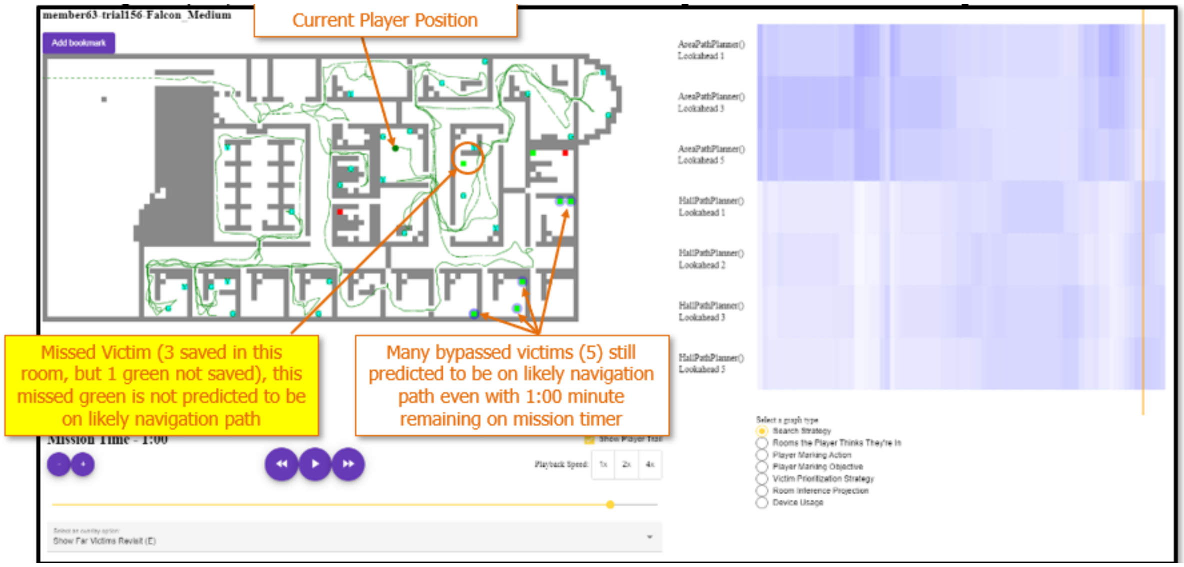


Figure 65: PSI-Coach inference of victim-revisiting

3.7.3.6 Capability F: Infer Bypassed Victims are Being Left Behind

F: Predict when a subject bypassed a victim, when the victim is nearby now, and when the subject is leaving the vicinity of the victim.

- Predict probability a subject missed a nearby victim, and they are leaving the vicinity of the victim, (indicating that it may be more timely to intervene now rather than later) – in this case with only 20 seconds remaining in the mission
- Map shows three missed previously-visited victims (weighted blue room highlight), as subject kept moving west passed the room with those victims
- Graph shows the running inference of subject's victim periodization strategy. subject is predicted to be using a SaveEverybody (with varying likelihood of VerifySaveEverybody), so PSI-Coach predicts that subject should be saving these nearby victims and they are leaving the vicinity.

Figure 66: PSI-Coach inference of victims left behind

3.7.3.7 Capability G: Infer Urgent Nearby, Bypassed Victims

G: Predict when a subject bypassed a victim, when the victim is nearby now, and when the victim is about to expire.

- Predict probability a subject missed a nearby victim, and the victim is about to expire, but there is still time to save it. In this case with 37 seconds left 4 victims are highlighted after subject comes out of a room where they saved two victims
- Map shows **missed three previously visited victims** (weighted blue room highlight), as subject kept moving west passed the room with those victims
- Graph shows the running inference of subject's **victim periodization strategy**. subject is predicted to be using a SaveEverybody (with varying likelihood of VerifySaveEverybody), so PSI-Coach predicts that subject should be saving these nearby victims and they are leaving the vicinity.

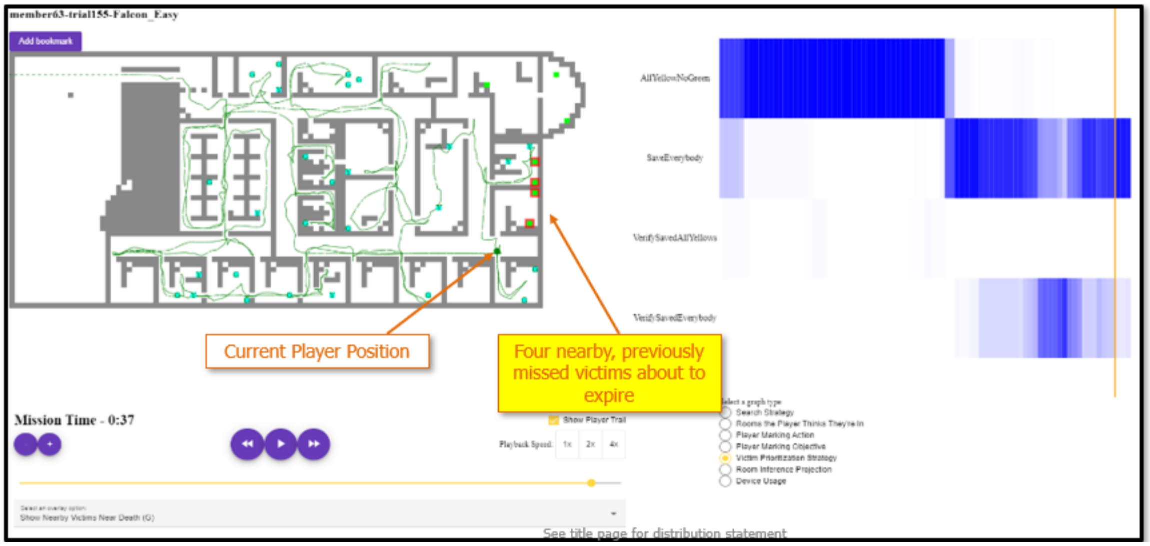


Figure 67: PSI-Coach inference of nearby, bypassed victims

3.8 PSI-Coach Integration and Testing

The goal of this task was to integrate PSI-Coach into a standalone system and test it in the TA3 testbed environment.

We actively supported ASIST's **Testbed Working Group** and established a development infrastructure and process. We created an initial **architectural design** and implemented initial versions of key components in the pipeline critical for the next experimental evaluation. The three main PSI-Coach components (the Data Abstractor, Cognition Inverter, and Strategic Coach) interact via shared memory, reading from a MQTT bus fed by Malmo Data Collector.

The pipeline supports both online and offline agent execution to assess performance with and without real-time constraints.

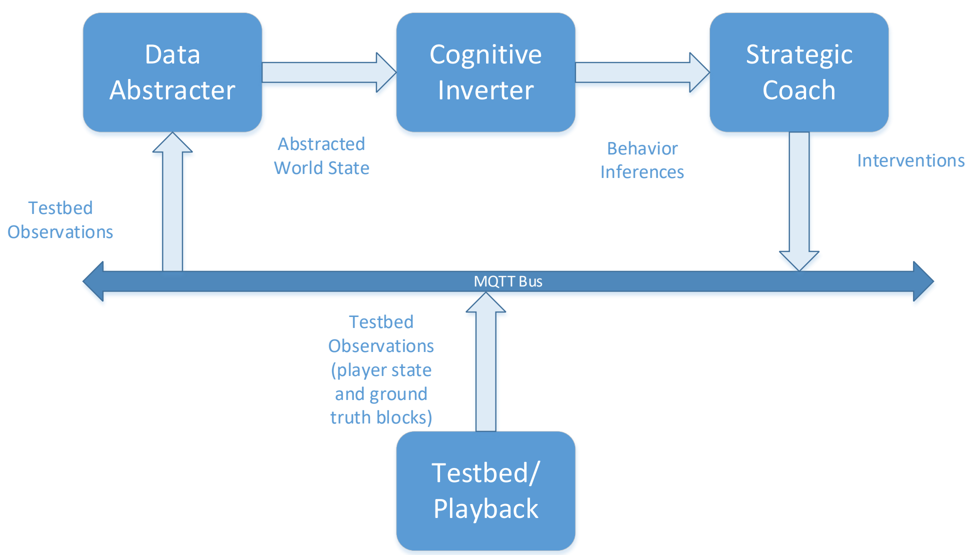


Figure 68: PSI-Coach architecture uses proxy components to accelerate testbed integration

We completed a fully functional pipeline from data to visualization. We designed a physical layer representation of the world and a **conceptual space hierarchy** that builds on it. The physical layer is represented as a 2D block grid to simplify field of view (FOV) calculations and navigation inferences. Spaces are connected in a graph. Agents can reason about space attributes (e.g., victim presence, proximity to other spaces) in multiple levels of the hierarchy. We differentiated between the ground truth state and the observed state. The observed state is a subset of ground truth based on observations that are inferred from player state (i.e., position and orientation in the physical layer).

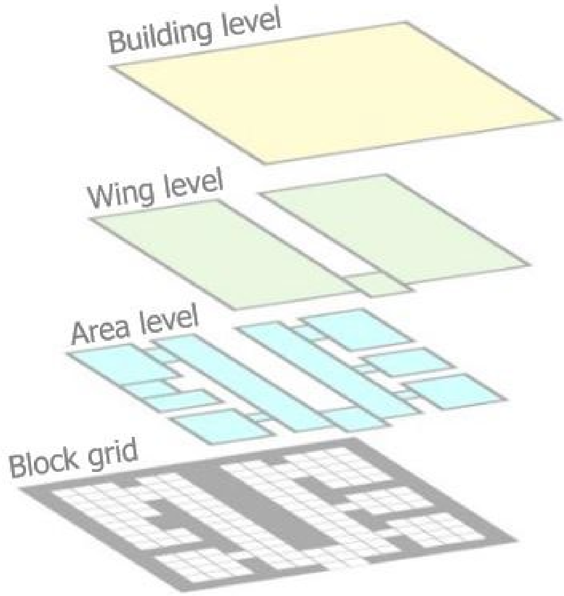


Figure 69: PSI-Coach uses a cognitive hierarchy of spatial locations, which can be re-indexed based on participants' beliefs, goals, and confusion

We implemented an event handling framework to help manage the asynchronous information flow. We provided additional support for all relevant primary events generated by the testbed, including the following: Beep, BlockageList, Door, Lever, MissionStatus, Observation, Triage, VictimList, and VictimExpired

We defined a set of secondary events derived from the primary testbed events that represent data at the appropriate level of abstraction for inference, prediction, and intervention. These events included the following: IgnoredBeep, RoomDeparted, SkippedSave, SkippedTriage, SpaceTransition, VictimAppears, and GroundTruthGraphUpdated

We defined a CognitiveInverterStateUpdate event to pass derived information from the Cognition Inverter to the Strategic Coach when there are updates to the inferences. We also defined a PlaybackStatus event to provide resilience to malformed experimental metadata executed by the Playback script.

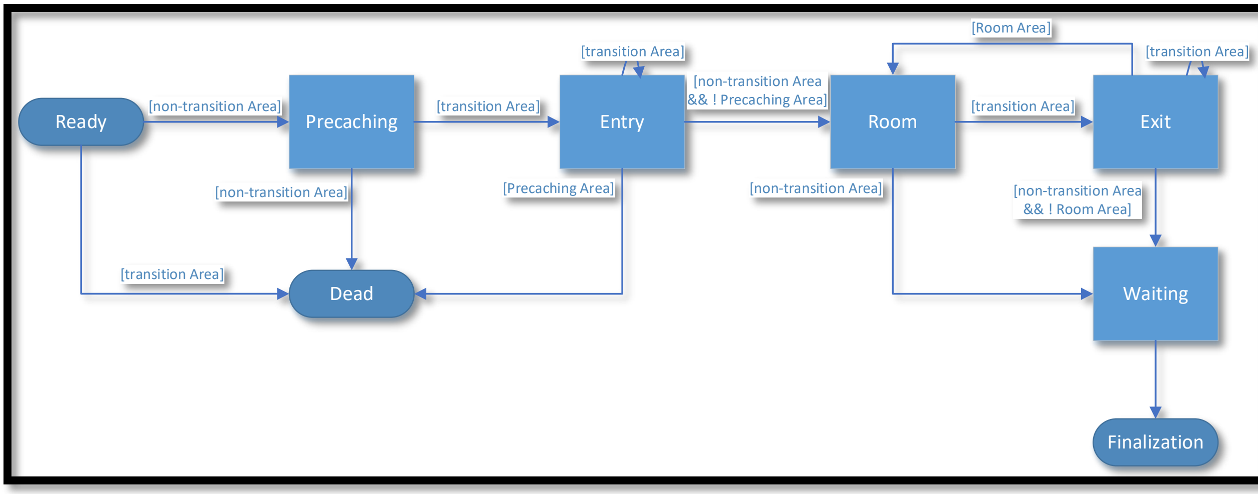


Figure 70: Gathering data for the RoomDeparted event requires accumulation of observational data over several states with state transitions triggered by SpaceTransition events

We integrated with TA3 testbed releases. We generated semantic maps of the spatial hierarchy for each of the experimental maps and created a Playback script to replay experimental trial data and a batch script to run Playback over many trials. We ingested engineering and HSR trial data and analyzed results. We completed the integration of all PSI-Coach capabilities for processing TA3 **Study 1 experimentation data**, which included processing 200+ testbed message bus data sets (.metadata files), displaying video files for Charles River's annotation tool (for 52 trials of the TriageSignal-StaticMap condition), and ingesting all testbed map complexity levels. We developed a metric calculator to capture metrics on distance traveled and percentage of steps spent retracing previous steps. We shared an algorithm used to identify a "can see" FOV sensor, which indicate what is visible given the position and orientation of the player and the presence of objects (e.g., walls) blocking the line of sight.

We shared scripts for starting and stopping the testbed in a Windows environment, facilitating experiments with Commander reports. We contributed to the development of the Semantic Map design specifications. We shared playback script that replays experimental data in real time (at the same speed as the original experiment). We contributed to demonstrations and source code development for map and video annotation tools for encoding human observers' inferences player behavior used to compare AI agent performance. We contributed to establishing the feasibility and benefits of a map-based intervention for the then-anticipated March 2021 study.

We updated the Data Abstractor to support our continued experimentation. Our efforts included adding support for new events from the testbed, including RoleSelected, RubbleDestroyed, ToolDepleted, and ToolUsed. We added entity tracking for victims that are moved and logic to update graph connectivity as rubble is removed. We also added a new secondary event, PlayerAppears, that is triggered whenever players appear in another player's VOF. We added support for gravity-sensitive stacked rubble and victims hidden under the rubble. We updated the block occupiability logic to account for new experimental constraints on players, such as no jumping. Our testing support efforts included enabling dockerization of PSI-Coach and deploying it on the TA3 testbed. We updated the local testbed with enhancements from the TA3 development testbed.

We updated the agent to support various iterations of Saturn map data published before, during, and after the Hackathon. The Data Abstractor was updated to support the latest details of the next experiment. The Nearbyness of the logical spaces is now automatically calculated instead of manually defined. The spatial view of the map supports temporal queries (e.g., were these two spaces connected at a specific time in the past?). We added support for new events from the testbed: MarkerPlaced, ScoreboardEvent, VictimList, Triage, ProximityVictimInteraction, VictimPickedUp, VictimPlaced, MarkerAppears, and PlayerFrozenStateChangeEvents

We also added support for Player objects tracking which players are observing it. We added TeammateMarkerProximityEvents to trigger inference of the next predicted action. Agent deployment and the testing pipeline were updated to reflect the latest changes to the experiment. We installed local instance of the updated testbed and refined the build script to integrate Julia-based Strategic Coach.

The Data Abstractor was updated as we began the transition to Study 3. We updated the TeammateMarkerProximity event to ignore markers labeled 3. We reduced the number of MarkerPlaced events processed for inference by ignoring similar events within close proximity. We added new events for trial marking and AgentPrediction. We switched the agent to use participant ID exclusively and no longer depend on the player name, which is intended to be deprecated. We made agent deployment and testing pipeline updates to support Study 2 PI meeting and Study 3 preparation. We added processing of trial information to support conversions between map player name, participant ID, and call sign in support of program metric M7. We updated the program metrics messages because of changes to the message specification requiring an exact match of the prediction time. We generated baseline results for PSI-Coach agent the metrics. We generated simple intervention logic for message generation to test the intervention pipeline, as well as to provide support for intervention messages. We deployed the Docker container to the TA3 registry and ran the agent on the TA3 testbed, receiving feedback.

4. Technical Results

4.1 Results

Our efforts have concluded that PSI-Coach successfully inferred human intents and strategies: PSI-Coach identification of variation in subjects' strategy is closer to human coach identification versus baseline coach (significant at $p < 0.05$). Baseline coach (i.e., first 5mins yellow-only, green-only thereafter) is correct in the majority of cases (~80%); both PSI-Coach and human coaches identify novel variations missed by the baseline coach.

PSI-Coach was particularly strong at inferring victim triage prioritization strategy (plan), skipped/missed victim bypasses (intent), and navigation strategy including rooms likely to be missed. PSI-Coach was often delayed in recognizing strategy switches; the team worked on extending device usage inferences.

Charles River Analytics collaborated across the program, sharing annotation tool source code, leading Hypothesis working group, and working with TA2s.

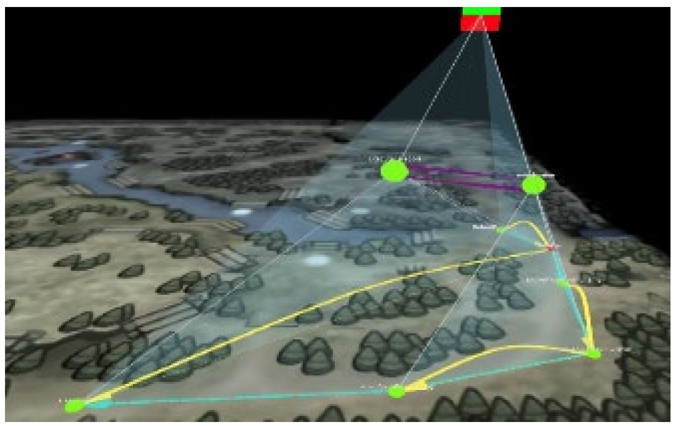


Figure 71: PSI-Coach infers goals and mental states of human subjects and maintains multiple hypotheses to predict good, socially-intelligent intervention points

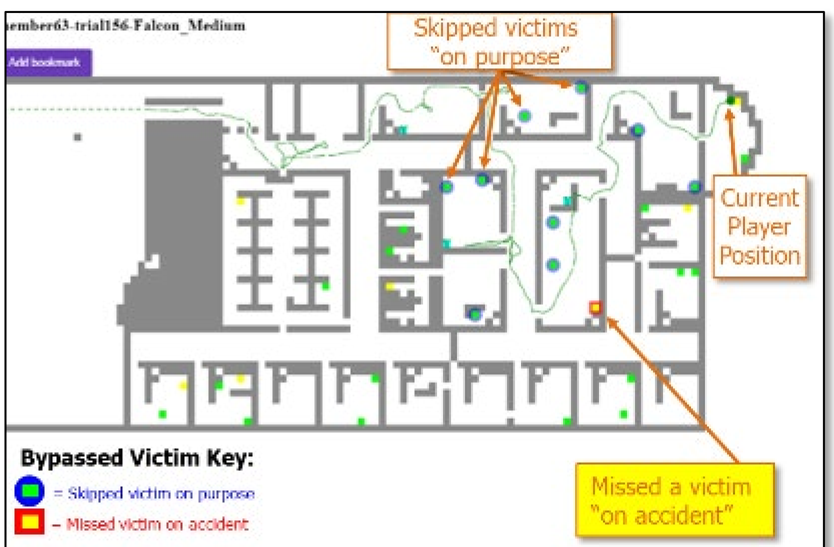


Figure 72: PSI-Coach infers if player intentionally skips a victim or missed a victim on accident

Compared to the Baseline Coach:

1. ***Strategy Inference:*** PSI-Coach agrees more with Human Coaches in triage strategy prediction
(PSI-Coach 82.5%, Baseline 75.0%, p<0.001)
2. ***Mental Model Inference of victims skipped-on-purpose vs. accidentally-missed:*** PSI-Coach agrees more with Human Coaches missed vs. skipped inferences for subjects with strategy variations
(PSI-Coach 91.5%, Baseline 60.9%, p<0.05)
3. ***Intervention Inference:*** PSI-Coach agrees more with Human Coaches in combined consider-intervening-but-choosing-not-to-now and intervene-now judgments
(PSI-Coach 27%, Baseline 18%, p<0.05)
4. ***Intervention Inference****:* PSI-Coach acts more often when Human Coaches do not
(PSI-Coach 39.2%, Baseline 20.6%, p<0.05)
5. ***Intervention Inference****:* PSI-Coach misses fewer Human Coaches' actions
(PSI-Coach 55.4%, Baseline 77.2%, p<0.05)
6. ***Intervention Inference****:* PSI-Coach agrees more with intervene-now judgements by Human Coaches (PSI-Coach 25.1%, Baseline 17.2%, p<0.05)
7. ***Intervention Inference****:* PSI-Coach misses fewer Human Coaches' intervene-now judgements
(PSI-Coach 38.7%, Baseline 66.4%, p<0.05) [Slide 44]

4.1.1 Research Theme #1

- Seek to make predictions that tease out effective versus annoying interventions:
  - In Phase II → Guide Interventions that
    - Build credibility
    - Avoid losing credibility by having interventions perceived as not useful (even if they are)
    - Reduce number of unneeded interventions
    - Choose timing to maximize adoption & perception of value (as well as actual value)
    - In Phase I, this Aim informs:
  - The distinctions in ToM we target
  - The aggregations of ToM we create
  - The range of projections of futures we seek to infer
  - Key Element:
  - Not targeting a single prediction or small set of predictions
  - Rather: a *continuously updated time series* of predictions to inform potential interventions decisions over time

Capability Statements Follow This Research Theme:

A. Recognize when user may be skipping victims, and whether they are skipping them accidentally vs. skipping them on purpose
B. Predict when a subject may be missing large portions of the building
C. Predict when a subject is missing a room and may stumble upon it before victims in the room expire
D. Predict when a subject is missing a victim and may stumble upon them before the victim expires
E. Predict when a subject is missing a victim and it will be on their path in the future
F. Predict when a subject is missing a victim, it is nearby now, and the subject is leaving the vicinity of the victim
G. Predict when a subject is missing a victim, it is nearby now, and the victim is about to expire
H. Predicting human working memory workload

4.1.2 Research Theme #2

- Predictions of many important individual elements are hard to do with high accuracy
    - Even for humans
    - We expect many of our important predictions to have this property as well
- However experienced human coaches are still able to coach effectively
- A key research question is how can we provide effective coaching interventions with predictions that are low probability
- An approach we are exploring in our research is
    - Aggregations of collections of low-probability predictions
    - With in-the-moment, situated reasoning
    - To provide high-probability, efficacious interventions from combinations of low-probability inferences

4.1.3 Summary ASI Strategy and Results

The PSI-Coach ASI Strategy can be summarized as follows. Recognize what the team is trying to do, and help them do it better with strategically timed interventions to improve their team processes:

- Understand what the team is trying to do (as individuals and as a team), and how they are using diverse team processes (effectively or ineffectively) by inverting cognitive models through probabilistic program inference and combining with Analytic Component reasoning
- Reason about possible targeted interventions now and in the future that can help to improve their processes as individuals and as a group; optimize using situated reasoning, interactive-narrative principles, and Monte-Carlo projection of tens of thousands of

possible futures each second to target both selection and timing for both effectiveness and understandability, actionability, and uptake/adoption

The results we generated, specifically for the final Study, measure effects from the PSI-Coach ASI compared to No-Advisor trials when PSI-Coach would have intervened. Specifically, we compare:

- Places in the trails where PSI-Coach chose to intervene to help raise team process
- Against places where it would have intervened in the no-advisor trials

We measure if the interventions made a difference in improving team process showing that the PSI-Coach just-the-right-thing at just-the-right-time approach has measurable and observable effects, such as targeting places where the team is trying to do something, and the team process was suboptimal for that given period of time. This is the process that we used to measure effects:

1. Document when PSI-Coach intervened and when it would have intervened
    a. Note (actual) intervention points from PSI-Coach trials
    b. Identify (predicted) intervention points from No Advisor trials
2. Measure the change in team process measures over an interval after each intervention point allowing time for responsive actions to affect measures
3. Compare change in team process measures between PSI-Coach trials and No Advisor trials by intervention type

Figure 73 shows the notable results.

Comparing 56 Teams: PSI-Coach vs. No-Advisor
PSI-Coach Improves Team Processes within ~60 seconds *

Intervention Type	Team Process Improvement (p=0.09) within 60 seconds
Aligned Team Priorities: Prioritize Critical Victim	**60%**
Coordinated Communication: Victim Marker Following	**13%**

Figure 73: PSI-Coach improves team processes with priority-based and coordinated communications-oriented interventions. * = Comparison of when PSI-Coach intervened vs. No Advisor trial where PSI-Coach would have intervened

This demonstrates PSI-Coach's ability to automatically recognize what the team is doing, identify team process problems, and tailored interventions and timing of the interventions to improve team processes. Experiments demonstrated ability to rapidly (within 60 seconds) improve team processes across diversity of teams: 28 advisor versus 28 no advisor trials.

- Improved Aligned-Team-Priorities by 60% (p=0.09)
- Improved Coordinated Team Communication by 13% (p=0.09)

Also, these interventions are based on Analytic Components (AC) that implement 4 Team performance theories, specifically:

1. CMU's Appropriate Use of Skill

2. IHMC’s Joint Activity
3. UCF’s Knowledge in the World
4. Charles River Analytics’ Team Alignment

To illustrate these effects see Figure 74, which shows an experimental segment from Study 3’s USAR task where PSI-Coach Recognizes the needs of individual teams and tailors interventions and timing to their situation.

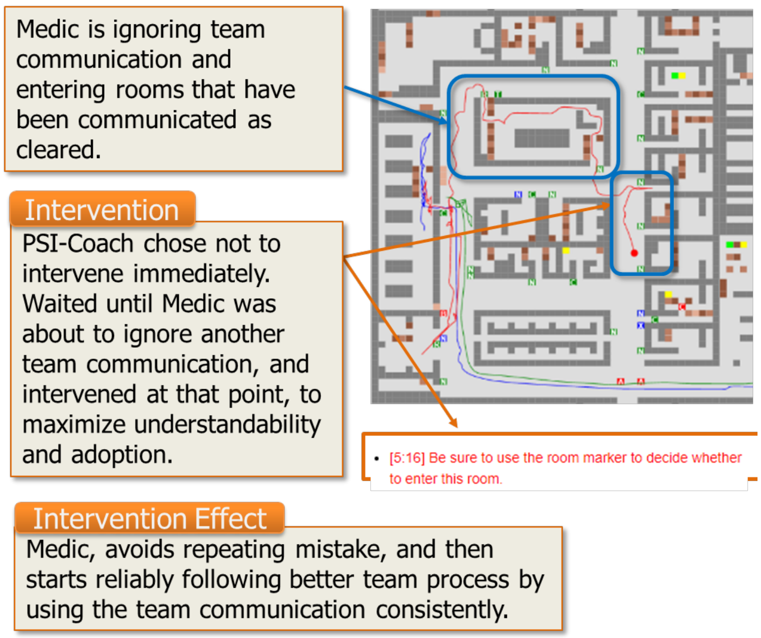


Figure 74: Right before the red medic participant was going to make the repeated mistake, PSI-Coach intervened at the appropriate time to point out team communication the medic was not acting upon.

These results support several TA1 hypotheses, including:

- TA1 CRA: ASIs can recognize team process deficiencies in individual teams
- TA1 CRA: ASIs can create tailored interventions and time them to the team situation to improve team processes
- TA1 CRA: ASIs can robustly perform these targeted interventions when needed across a diversity of teams

4.2 Discussion

Much program-wide discussion was aimed at the challenges of the USAR task to study team performance. Here are a couple of recommendations to address the challenges.

Challenge: Task performance is dominated by Minecraft skill, making inferring and intervening on ToM elements less effective.

- ***Recommendation****:* Modify the task to have more strategies (e.g., marking strategies, verification passes) that use knowledge more than Minecraft skill (e.g., fix bugs in the device and make the use have a greater performance benefit).

Challenge: The Triage Strategy has little effect on performance, which makes intervention with regards to ToM on this strategy of little value to the player (undermining the ASIST "Usefulness" metric); Deterministic time when yellow victims expire further exacerbates this.

- ***Recommendation:*** Make time of yellow expiration unknown and random; reduce number of victims to allow for richer verification strategies; give points for remaining time; consider other small changes to multiple strategies can produce significant, measurable improvement in performance (e.g., marking strategies that contribute to score, increase penalty for dead victims in concert with reduced number of victims to enable verification/completion strategy to differentiate; longer term—add knowledge-based performance elements).

5. Important Findings and Conclusions

5.1 Conclusions

PSI-Coach's artificial social intelligence (ASI) agent employs methods to decide whether to intervene immediately or wait until a more appropriate time. The method is based on automatically inferring team Theory of Mind (TToM) and Machine Theory of Teams (MToT), and was demonstrated with four diverse theories of team processes. The four theories included Collective Intelligence (CMU TED and BEARD), Joint Activity (IHMC), Shared Knowledge in the Head and World (UCF), and PSI-Coach team alignment (CRA). CRA demonstrated increased agreement with inferences made by human coaches as well their choices and timing to intervene. In addition to comparison with human advisors, the CRA agent outperformed a non-social-intelligent agent. This comparison refers to how much PSI-Coach is able to match human coaches' judgments, compared to how much human coaches' match a non-socially-intelligent agent (i.e., heuristics and full ground truth to make the inferences, and a greedy strategy for when to intervene). Experiments showed a 35% increase ($p<0.05$) across two MToT real-time inferences (i.e., ability to automatically infer team alignment/misalignment in the believed meaning and use of marker blocks for communication, and in the role assignment strategies). CRA also showed a 42% and 68% increase in agreement with human coaches on intervention points ($p<0.05$). CRA demonstrated the ability to automatically identify team process problems unique to each team and their situation dynamics, and provided tailored intervention content with optimal timing that improved team processes within 60 seconds. Experiments showed, compared to all of the no-advisor teams, a 60% improvement in Aligned-Team-Priorities ($p=0.09$) and 13% improvement in Coordinated-Communication ($p=0.09$) compared to baseline trials with no interventions.

Testing initial PSI-Coach system on hundreds of human trials, 54 trials comparing to human coach annotations, showed PSI-Coach to be:

- Strong at inferring subjects' victim triage prioritization strategy (plan), skipped/missed victim bypasses (intent), and navigation strategy including missed rooms (task activity), and field of view (open world observations)
- Often delayed in recognizing strategy switches
- Incomplete in the inference implementation of the victim-detection device usage; work is in progress on extending

Creating an efficient experiment that has strong relevance to all TA1 & TA2 performers and can be scaled appropriately continues to be a key challenge for ASIST. For example:

- Study 1 required very little use of strategy, a simple rule-based baseline was 80% correct; Minecraft player travel speed was most important factor in task performance, eclipsing other ToM variations concerning plans, intent, device usage, etc.
- Study 2 current draft introduces many features for team adaption (re-charging tools, rubble-removal) but is more individual performance rather than specifically requiring ToM for optimal team performance resulting from good social intelligence
- Recommend TA3 add a game design expert to include ASI and team ToM features, drawing from game-balancing principles

Establishing a common framework and language is critical for communication across both TAs and technical disciplines; the program and research community will continue to benefit from establishing such commonality.

Cognition Inverter Conclusions. Cognitive Architectures combined with Probabilistic Programming to implement "inverse cognition" can be used to represent important player mental model variations, including player intent, task strategy, task execution, and resource usage and knowledge, among others. Cognition Inverter algorithms can infer dynamically-changing player mental models (e.g., unexpected strategies, triage intent change) from observations of player behavior and predict future execution of strategies to enable well-timed intervention points, similar to human coaches.

Strategic Coach Conclusions. Task and team ToM predictions (captured by Capability statements A–G) that are highly relevant to interventions can be constructed from Cognition Inverter ToM inferences. Interventions generated by initial strategic coach correlate with human annotator judgements concerning when interventions are helpful and necessary, e.g., diagram to right shows inferences that lead to a well-timed intervention, in this case, suggesting the player saves a nearby, previously missed, high-priority victim.

Integration and Test Conclusions. Definition of a data abstraction layer insulates the modeling and inference algorithms from testbed specifics and increases the generalizability of the approach. Separation of the processing pipeline into multiple discrete components (i.e., Data Abstracter, Cognition Inverter, Strategic Coach) with clearly defined inputs/outputs increases experimentation flexibility and enables richer composability, e.g., enabling TA2 analytic agent integration.

5.1.1 Large-Scale Team Study and Machine Theory of Teams (MToT) Reasoning

PSI-Coach extends probabilistic programming, cognitive models, and interactive narrative research to implement a Machine Theory of Teams and improve team processes.

- Accomplishments
    - Completed openly shareable, massive team data collection <nn teams, mmm trails, nn hours video, xx MB machine understandable team activity data repository; 3-participant USAR teams in Minecraft testbed
    - Quantified predictive characteristics of good/bad teams: coordination, prioritization, competency categories
    - Recognized and intervened on real-time measures of good/bad team performance using 6 different Artificial Social Intelligence (ASI) advisors
- Why is it better?
    - Post-facto. Predictive processes of good/bad teams in line with human-expert assessments (which takes hours)
    - Real-time. Measurable team process improvement after ASI advisor interventions, compared to similar situations in no-advisor baseline
- How does this help teams?
    - Automate AARs using good/bad team performance characteristics

- Improve team processes at the right time to have a measurable benefit to mission effectiveness

5.2 Significant Hardware Development

There was no significant hardware development under this contract.

5.3 Special Comments

5.3.1 Future ASI Capabilities

Future work to build additional and improved ASI capabilities would be valuable. Within our demonstrations, we noted the following instances where a fully productized version of a PSI-Coach ASI would have intervened but research PSI-Coach did not:

Scenario: Team accidentally misses critical tasks, and it will be harder to do them later.

- ***Potential Intervention:*** Check in on critical tasks and prioritization. ASI enables relevant reminders that are on strategy. Choosing when to intervene requires additional theory of mind and avoids alert fatigue from pervasive simple criteria.

Scenario: Team underprioritizes supportive actions, waits 45 seconds to start providing support, wasting team time.

- ***Potential Intervention:*** Provide supportive action prioritization advice. ASI enables automated analysis of the benefits of team processes and makes the intervention where it makes sense.

6. Implications for Further Research

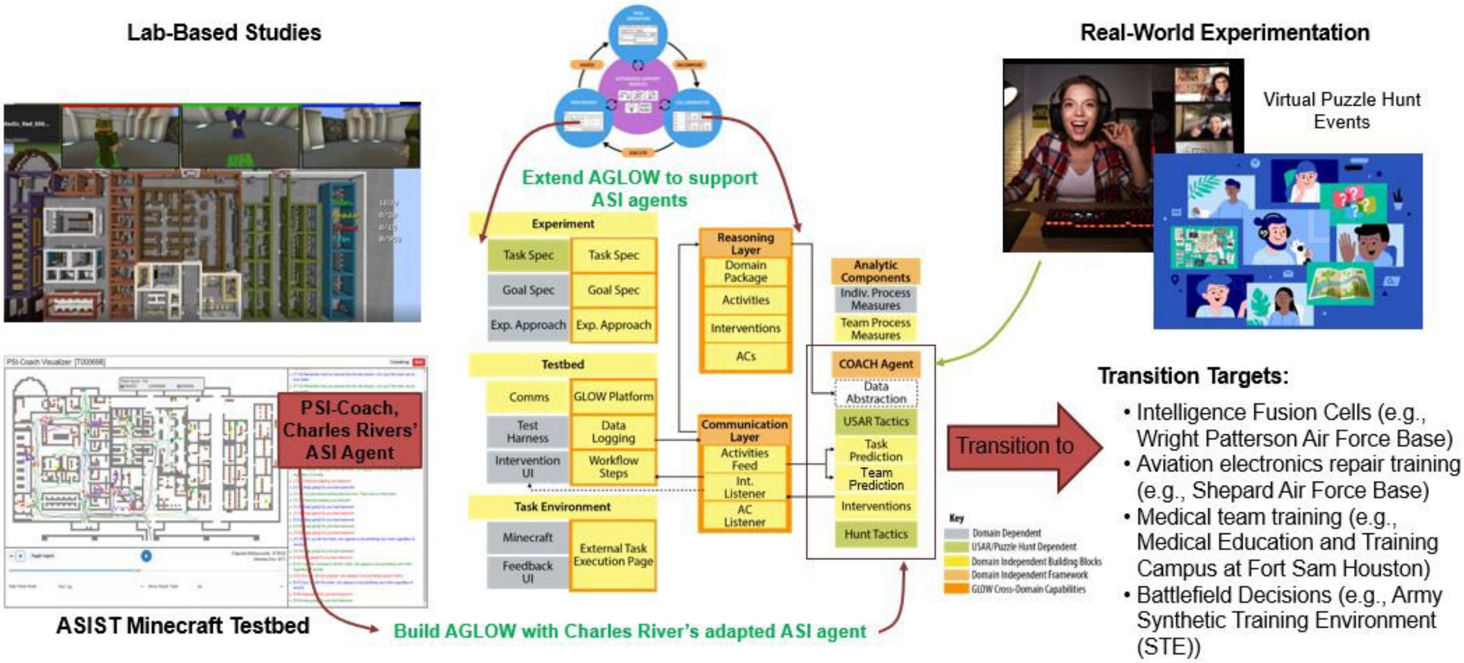
Our extensive efforts on PSI-Coach have broad research applications with technological transitions already underway. See below for descriptions on Charles River's AGLOW and CADE applications of the Cognition Inverter as well as how MARLIN-SPIKE is utilizing the Strategic Coach method.

6.1 ASI for Group Learning and Optimization of Collaborative Workflows (AGLOW) Summary

The DARPA ASIST program demonstrated early, promising results from ASI implementations (known as ASI agents) for teamwork support. In controlled lab experimentation, these ASI agents have performed at near-human levels at some tasks and significantly below human performance in other tasks (Freeman et al., 2021). These results, though preliminary, suggest ASI interventions have the potential to scale teams by supporting teamwork functions, previously limited to human facilitators or team members. However, this work to date has been limited to a single domain—Urban Search and Rescue (USAR)—and has only been studied in university experimental settings (e.g., a curated participant population, in a synthetic (Minecraft) environment, with pre-established information resources for the team participants to access). Though ASIST research plans include experimentation with different task domains, there are no plans to include real-world studies with externally administered events, featuring open-ended tasks or resources. By conducting teamwork-support research with real-world ASI deployments, the DoD can gain unique insights for building new ASI features for a rapidly expanding commercial market, bolstered by increased remote-work demands in recent years.

We are extending an existing Charles River–developed task and communication management platform—Group Learning and Optimization of Collaborative Workflows (GLOW)—to include an ASI agent that can administer teamwork interventions in a real-world environment for a team puzzle-solving activity, such as a puzzle hunt event. GLOW's demonstrated support for cross-domain teamwork makes it a good platform to study ASI agent generalizability. GLOW has been used to collect data on collective function in domains such as intelligence analysis and innovation challenges, including methods to capture shared mental models and guide structured analytic techniques. GLOW uses modular task cards and team-process reasoning to help teams decompose their team tasks, set team goals, externalize their knowledge, assign tasks to appropriately skilled team members, support task execution, and assess team performance.

Our teamwork-support framework ASI GLOW, or AGLOW for short, achieves the following goals: (1) integrate the Charles River ASI agent from our DARPA ASIST work into the GLOW framework; (2) extend GLOW with new domain-general processing components and reasoning algorithms; (3) adapt our ASIST ASI agent for a new domain; and (4) deploy the ASI agent via AGLOW to a real-world puzzle-hunt environment. The puzzle-hunt environment is similar to teamwork performed by Intelligence Fusion Cells, with which we have significant expertise; both include a high reliance on the disparate expertise of team members, high uncertainties with respect to task complexity and team members' actions, the need to coordinate team efforts efficiently, access to open-ended information resources, and self-organized teams with both internal and external motivation to complete the task at hand. We also have in-house expertise in both performing in and running puzzle contest teams, such as MIT Mystery Hunts, spanning 20 years of participation. This is critical to building and training ASI agents for a new domain. We have access to decades of concrete data on past strategies and team performance required for our proposed domain adaptation modeling. AGLOW helps the generalization of ASI agents to address operational problems that arise from real-world deployment with higher uncertainties and greater complexities of teamwork.



Our technical approach focused on adding new team-scaling innovations to the GLOW platform. These innovations include abstracted layers for reasoning, communications, and environmental sensing to enable a repeated methodology for ASI integration for increased cross-domain applicability. Our approach also included select generalizations to our PSI-Coach ASI agent.

To address the problems inherent in team-scaling and problem solving, we identified four technical challenges to guide the AGLOW research: (1) ensure *agent domain-portability generalizability;* (2) develop and evaluate *domain-spanning metric categories,* and apply *ASI interventions in a new, relevant domain;* (3) apply *ASI interventions in a new, relevant domain*; and (4) ensure *generalizability of interventions*.

To ensure *agent domain-portability generalizability*, AGLOW provides an **Analytic Reasoning Layer**, which enables rapid translation of activities, analytic components, and interventions across different domains. Activities must also be captured by domain-specific interfaces. Developing the Analytic Reasoning Layer requires three distinct representations, which will be defined using a typed ontology based on Score™: (1) a definition of Domain Packages (the cards and workflows that are active in each domain); (2) an Activity representation that captures the meaning of different interface or algorithmic actions; and (3) an Intervention Representation that captures the content, form, and target of an intervention. The entire knowledge representation enables portable queries into the data collected from different domains.

To develop and evaluate *domain-spanning metric categories*, AGLOW provides a **Communications Layer**, which ensures that the components of GLOW can send and receive appropriate data from other components and from supported external interfaces. Developing a lightweight Communications Layer requires three distinct types of data pipelines: (1) Activity Feeds that generate and distribute representations of the abstract Activities that occur in real-time; (2) Techniques, which are activity patterns that correspond to individual or team tactics or strategies; and (3) Intervention Listeners that receive messages initiating specific Interventions, sent by ASI agents.

To *apply ASI interventions in a new, relevant domain*, AGLOW enhances our existing GLOW platform with domain-specific refinements represented as new workflows and cards in an **ASI Testbed**. These domain-specific refinements implement the data harness aspects of the Reasoning Layer, using Cards to represent Tasks, Team Members, and Objects, and using Workflows to implement Techniques and deliver Interventions. We are using domain-specific models, such as Petri nets of puzzle challenges (e.g., those addressed in the puzzle-hunt events) to enable AGLOW to support data collection and interventions during domain-specific events. Workflows represent the possible actions of team members that can build the Petri net or create markings that enable transitions through (high-level, team-relevant) domain states.

To ensure *generalizability of interventions*, we are conducting a series of evaluations of AGLOW in puzzle-hunt events, including scaled practice sessions, throughout each year of the effort. These range from engineering tests and practice sessions to field studies.

6.2 Cyber Adversary Discovery Engine (CADE) Summary

Cyber warfare is a rapidly expanding, critical battlefield for the US Navy. Attacks on infrastructure, ship systems, and sailors themselves can significantly reduce operational readiness and deployment time and can be very costly. To prepare for and successfully defend this rapidly evolving battlefield, defensive cyberspace operations (DCOs) must effectively analyze and forensically investigate attacks, both cyber-technical (e.g., attacks on computer networks and IT infrastructure) and cyber-social (e.g., social media influence operations). When executed by an attacker, or analyzed forensically, today's sophisticated attacks include complexity in multiple dimensions that creates challenges for analysis. Individual elements of an attack often have multiple parts that the attackers execute in parallel, and interleave dynamically, based on the details they see as they interact with the targets. The execution of these elements is also reactive and adaptive, with the attackers changing actions, details, or even entire elements in response to opportunities and contingencies that they encounter as they execute a portion of the attack. Further, modern attacks are complex and hierarchical, with multiple tiers of goals and behaviors that exhibit these dynamic complexities at all levels. The analysis is made more difficult because low-level data are typically presented in event logs or other displays that do not provide any insight into the connections of those actions.

Currently, few tools exist that support such complex analysis. Skilled analysts must manually perform forensic analyses of attacker behaviors, which requires significant investments of staff, time, and money, and limits the breadth and depth of the possible analyses. To understand these complex tactics, techniques, and procedures (TTPs) and behaviors, Navy forensic cyber analysts need: *(1) a way to represent models of complex, multi-tiered, parallel and adaptive adversary behaviors and TTPs; (2) a mechanism to automatically recognize and interpret complex* *hierarchical, adaptive attacker behaviors and TTPs in the data; and (3) tools that can partner with analysts, enabling them to rapidly find attacks in the data, create and test their hypotheses, and dramatically increase their effectiveness while reducing their effort.*

To address these challenges, Charles River Analytics designed and demonstrated the feasibility of the Cyber Adversary Discovery Engine (CADE). The CADE system represents the sophisticated TTPs and behaviors of today's cyber attackers; automatically finds attacks in forensic data; and collaborates with human analysts to identify goals, behaviors, TTPs, and changes in TTPs that are critical for today's DCOs. The CADE effort advances the technology used to perform forensic cyber analysis for cyber-social and cyber-technical attacks, as well as technology for understanding human behavior more broadly. It provides a collaborative AI-based partner to enable analysts to deeply understand the behaviors, goals, and TTPs of attackers. CADE also enables analysis of the personas involved in the attacks, and how the TTPs of individual attackers change over time. Understanding these behaviors will support a wide range of defensive strategies that are key to our national security, and will help us create simulation environments to bolster our nation's defense against subtle and pervasive attacks on social media platforms and our nation's computer systems.

To *represent models of complex, multi-tiered, parallel and adaptive adversary behaviors and TTPs* that are needed to build scalable models of cyberspace threat actors, CADE incorporates a **Hap Cyber Behavior Modeling System.** This system is built on Hap (Sliva et al, 2018; Loyall 1997), which is a behavior modeling system specifically designed to capture the complexity of the types of behaviors exhibited by cyber attackers. Earlier versions of Hap have demonstrated the ability to model complex cyber attacker behaviors. Hap enables the expression of hierarchical behaviors with multiple, interleaved and parallel behaviors at all levels. It includes mechanisms to directly model reactivity and adaptability in the models at all levels, as well as individual differences that could be exhibited by different attackers.

To *automatically recognize and interpret complex hierarchical, adaptive attacker behaviors and TTPs,* CADE incorporates a **Hap Behavior Inversion System,** or Inverse Hap, which uses machine learning to automatically recognize behaviors, goals and TTPs from forensic data. Inverse Hap, which combines Hap with probabilistic programming (PP) language technology (Pfeffer, 2016, Wood, van de Meent, & Mansinghka, 2014), has proven success on other efforts, such as DARPA's Probabilistic Programming for Advanced Machine Learning (PPAML) program. Unlike deep learning, which requires massive amounts of labeled data and produces unexplainable models, PP can learn from small amounts of data, even single instances of data, in the way that people can, and produces intuitive, explainable models. As a PP version of Hap, Inverse Hap enables the learning of explainable behavior models with the behavioral expressiveness of Hap.

To create tools that can partner with analysts, allowing them to rapidly find attacks in the data, create and test their hypotheses, and dramatically increase their effectiveness while reducing their effort, CADE incorporates an **Interactive Cyber Visualization & Exploration (ICV&E) System**. We based the ICV&E System on the Stratmapper (Canossa, Nguyen, & Seif El-Nasr, 2016) and Glyph (Nguyen, Seif El-Nasr, & Canossa, 2015) visualization tools. These tools were created by our partner, Professor Magy Seif El-Nasr, and have demonstrated the ability to uncover tactics and behaviors that were not possible with machine learning alone (Nguyen et al., 2015; Nguyen, Subramanian, Seif El-Nasr, & Canossa, 2014). This allows analysts to immediately see recognized tactics and behaviors, as well as discover additional patterns, tactics, and behaviors.

6.3 Mentoring and Responsive Learning through Intelligent Nautical Skill-modeling, Prompting, Intervention, and Feedback during Independent Exercises (MARLINSPIKE) Summary

The safety and operational success of the US Navy's fleet depends on the expert navigation, seamanship, and ship handling skills of its Surface Warfare Officers (SWOs). To address these challenges, Charles River Analytics and the University of Nevada Reno (UNR) are developing and evaluating a full-scope prototype Mentoring and Responsive Learning through Intelligent Nautical Skill-modeling, Prompting, Intervention, and Feedback during Independent Exercises (MARLINSPIKE) system. MARLINSPIKE features (1) a scenario generation and director agent that uses search-based experience management to project potential training outcomes and plan training scenario elements from high-level, quick-to-author specifications; (2) Charles River's Hap reactive planning architecture to make realistic and reactive high-level decisions for each ship; (3) UNR's ship movement controller using rules of the road and potential fields to navigate ships through traffic and around obstacles to meet the high-level decisions; and (4) UNR's Rules of the Road ship handling simulation to enable ease of integration and rapid iteration on development prototypes, transitioning to Navy Mariner Skills Suite (MSS) and Navigation Seamanship Shiphandling Trainer (NSST) simulations for transition efforts.

To *automatically translate desired training elements into adaptive training scenarios*, MARLINSPIKE features a scenario generation and director agent that uses artificial intelligence (AI) search, planning, and adaptive execution (Niehaus & Riedl, 2009; Weyhrauch, 1997). During scenario generation, training staff identify the scenario specifications in a high-level, quick-to-author configuration file. For example, the training staff would list the desired world location, ships or ship classes, weather and environmental conditions, and the training challenges to be encountered (such as a rapidly approaching speed boat). The director agent translates these specifications into constraints for the scenarios and then searches the space of possible scenarios to find a complete configuration that meets those constraints and default constraints, setting the world location and specific weather parameters, propagating each ship and nationality, and setting the objectives for the speed boat and other ship AIs. During execution, the director agent manages the scenario to ensure that each of the ship handling AIs are not interfering with the main scenario events and that they maintain their realism. The director agent accomplishes this by search-based experience management, projecting possible futures using a Monte-Carlo tree search (MCTS) and intervening to change scenario elements or ship AI goals when beneficial to the overall arc of the scenario. With this approach, scenarios adapt dynamically to the trainee's actions to ensure that training challenges are encountered in a realistic manner. We have previously used this approach to create scenarios for inter-cultural communication, commercial interactive entertainment applications (Zoesis), battle plans for interactive wargaming (Deep Green), and terrain for Marine Corps adaptive decision-making training (SCALE/ATLAS).

MARLINSPIKE will enable more and more effective use of virtual ship handling training, by improving the automation supporting scenario creation, configuration, and execution. The safety and operational success of the (USN's) fleet depends on the expert navigation, seamanship, and ship handling skills of its SWOs. As the Navy is continually tasked to do more with less, virtual training, in concert with other training practices, has the potential to significantly improve training. The Navy has a critical need for rigorous and comprehensive ship handling training. They have begun to address this with multiple virtual environments—COVE, NSST, and MSS—that provide realistic virtual exposure to common ship handling evolutions in limited scenarios. However, training scenario creation and execution still heavily rely on master mariners performing detailed, labor-intensive planning, configuration, and management. Realistic training scenarios enable trainees to navigate crowded, limited space waterways such as harbors and canals with many other ships. Scenario developers and training staff must first select, place, and configure all non-trainee ships in the scenario, set waypoints and navigation controls for the ships to follow during scenario execution automatically and blindly, and then personally navigate the key ships to provide realistic and challenging training experiences. Creating automated capabilities for ship handling training scenario development and execution will enable:

1. More training scenarios (at a reduced cost) due to reducing complex, time-consuming, and error-prone manual scenario creation and debugging
2. Reduced requirements for costly expert training personnel
3. Increased availability to run training scenarios because of reduced training staff requirements
4. More realistic and nuanced automated ship behaviors, freeing human operators for other tasks
5. Adaptive training scenarios to better meet individual trainee needs

And these benefits will enable more effective virtual ship handling training experiences and more “reps and sets” of these training experiences.

7. References

Abbasi, Y. D., Ben-Asher, N., Gonzalez, C., Kar, D., Morrison, D., Sintov, N., & Tambe, M. (n.d.). *Know Your Adversary: Insights for a Better Adversarial Behavioral Model*. CogSci. 2016.

Arora, N. S., Russell, S., & Sudderth, E. (2013). NET-VISA: Network Processing Vertically Integrated Seismic Analysis. *Bulletin of the Seismological Society of America*, *103*(2A), 709–729. http://doi.org/10.1785/0120120107

Bates, C., Battaglia, P., Yildirim, I., & Tenenbaum, J. B. (2015). *Humans predict liquid dynamics using probabilistic simulation.* Presented at the CogSci.

Bates, J., Loyall, A. B., & Reilly, W. S. (1992). Integrating Reactivity, Goals, and Emotion in a Broad Agent. *In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*.

Bohus, D., Raux, A., Harris, T., Eskenazi, M., & Rudnicky, A. (2007). Olympus: an open-source framework for conversational spoken language interface research. *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, 32–39. Retrieved from https://www.aclweb.org/anthology/W07-0305

Bracken, B. K., Alexander, V., Zak, P. J., Romero, V., & Barraza, J. A. (2014). Physiological synchronization is associated with narrative emotionality and subsequent behavioral response. *International Conference on Augmented Cognition*, 3–13. Springer.

Chikersal, P., Tomprou, M., Kim, Y. J., Woolley, A. W., & Dabbish, L. (2017). Deep structures of collaboration: physiological correlates of collective intelligence and group satisfaction. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 873–888. ACM.

Coyne, J., & Sibley, C. (2016). Investigating the use of two low cost eye tracking systems for detecting pupillary response to changes in mental workload. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *60*, 37–41. Sage Publications Sage CA: Los Angeles, CA.

Cranford, E. A., Gonzalez, C., Cooney, S., & Tambe, M. (n.d.). *Learning about Cyber Deception through Simulations: Predictions of Human Decision Making with Deceptive Signals in Stackelberg Security Games*. 6.

Dechter, R. (1998). Bucket Elimination: A Unifying Framework for Probabilistic Inference. In M. I. Jordan (Ed.), *Learning in Graphical Models* (pp. 75–104). https://doi.org/10.1007/978-94-011-5014-9_4

Dechter, Rina. (1998). Bucket elimination: A unifying framework for probabilistic inference. In *Learning in graphical models* (pp. 75–104). Springer.

Doucet, A., de Freitas, N., & Gordon, N. (2001). An Introduction to Sequential Monte Carlo Methods. In A. Doucet, N. de Freitas, & N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice* (pp. 3–14). https://doi.org/10.1007/978-1-4757-3437-9_1

Dutt, V., Ahn, Y.-S., & Gonzalez, C. (2013). Cyber Situation Awareness: Modeling Detection of Cyber Attacks With Instance-Based Learning Theory. *Human Factors*, *55*(3), 605–618. https://doi.org/10.1177/0018720812464045

Elkin-Frankston, S., Bracken, B. K., Irvin, S., & Jenkins, M. (2017). Are Behavioral Measures Useful for Detecting Cognitive Workload During Human-Computer Interaction? In T. Z. Ahram & W. Karwowski (Eds.), *Advances in The Human Side of Service Engineering* (Vol. 494, pp. 127–137). https://doi.org/10.1007/978-3-319-41947-3_13

Engel, D., Woolley, A. W., Aggarwal, I., Chabris, C. F., Takahashi, M., Nemoto, K., … Malone, T. W. (2015). Collective Intelligence in Computer-Mediated Collaboration Emerges in Different Contexts and Cultures. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 3769–3778. https://doi.org/10.1145/2702123.2702259

Freeman, J., Huang, L., Wood, M., & Cauffman, S. J. (2022). Evaluating Artificial Social Intelligence in an Urban Search and Rescue Task Environment. In N. Gurney & G. Sukthankar (Eds.), *Computational Theory of Mind for Human-Machine Teams* (Vol. 13775, pp. 72–84). Cham: Springer Nature Switzerland. http://doi.org/10.1007/978-3-031-21671-8_5

Geman, S., & Geman, D. (1987). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *Readings in computer vision* (pp. 564–584). Elsevier.

Giannakakis, G., Pediaditis, M., Manousos, D., Kazantzaki, E., Chiarugi, F., Simos, P. G., … Tsiknakis, M. (2017). Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control*, *31*, 89–101.

Glikson, E., Woolley, A. W., Gupta, P., & Kim, Y. J. (2019). Visualized Automatic Feedback in Virtual Teams. *Frontiers in Psychology*, *10*. https://doi.org/10.3389/fpsyg.2019.00814

Gonzalez, C., Ben-Asher, N., & Morrison, D. (2017). Dynamics of Decision Making in Cyber Defense: Using Multi-agent Cognitive Modeling to Understand CyberWar. In P. Liu, S. Jajodia, & C. Wang (Eds.), *Theory and Models for Cyber Situation Awareness* (pp. 113–127). https://doi.org/10.1007/978-3-319-61152-5_5

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591–635. https://doi.org/10.1207/s15516709cog2704_2

Gupta, P, Kim, Y. J., Glikson, E., & Woolley, A. (2019). Digitally nudging team processes to enhance collective intelligence. *Proceedings of Collective Intelligence 2019*.

Gupta, Pranav, & Woolley, A. (2018). Productivity in an Era of Multi-Teaming: The Role of Information Dashboards and Shared Cognition in Team Performance. *Proceedings of the ACM on Human-Computer Interaction*, *2*, 1–18. https://doi.org/10.1145/3274331

Hackman, J. R., & Wageman, R. (2005). A Theory of Team Coaching. *Academy of Management Review*, *30*(2), 269–287. https://doi.org/10.5465/amr.2005.16387885

Hasson, U., Furman, O., Clark, D., Dudai, Y., & Davachi, L. (2008). Enhanced intersubject correlations during movie viewing correlate with successful episodic encoding. *Neuron*, *57*(3), 452–462.

Kim, Y. J., Engel, D., Woolley, A. W., Lin, J., McArthur, N., & Malone, T. W. (2016). What Makes a Strong Team? Collective Intelligence Predicts Team Performance in League of Legends. *Academy of Management Proceedings*, *2016*(1), 13564. https://doi.org/10.5465/ambpp.2016.13564abstract

Kulkarni, T. D., Kohli, P., Tenenbaum, J. B., & Mansinghka, V. (2015). *Picture: A probabilistic programming language for scene perception*. 4390–4399.

Lejarraga, T., Lejarraga, J., & Gonzalez, C. (2014). Decisions from experience: How groups and individuals adapt to change. *Memory & Cognition*, *42*(8), 1384–1397. https://doi.org/10.3758/s13421-014-0445-7

Li, R., Curhan, J., & Hoque, M. E. (2015). Predicting video-conferencing conversation outcomes based on modeling facial expression synchronization. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, *1*, 1–6. https://doi.org/10.1109/FG.2015.7163102

Lim, Y. M., Ayesh, A., & Stacey, M. (2015). Using Mouse and Keyboard Dynamics to Detect Cognitive Stress During Mental Arithmetic. In K. Arai, S. Kapoor, & R. Bhatia (Eds.), *Intelligent Systems in Science and Information 2014* (pp. 335–350). Springer International Publishing.

Loyall, A. Bryan, Reilly, W. S. N., Bates, J., & Weyhrauch, P. (2004). System for Authoring Highly Interactive, Personality-rich Interactive Characters. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 59–68. https://doi.org/10.1145/1028523.1028532

Loyall, Aaron B. (1997). *Believable Agents: Building Interactive Personalities.* (No. CMU-CS-97-123). Retrieved from CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE website: https://apps.dtic.mil/docs/citations/ADA327862

Maaoui, C., Bousefsaf, F., & Pruski, A. (2015). Automatic human stress detection based on webcam photoplethysmographic signals. *Journal of Mechanics in Medicine and Biology*, *16*(04), 1650039. https://doi.org/10.1142/S0219519416500391

Mannaru, P., Balasingam, B., Pattipati, K., Sibley, C., & Coyne, J. (2016). Cognitive context detection in UAS operators using eye-gaze patterns on computer screens. *Next-Generation Analyst IV*, *9851*, 98510F. International Society for Optics and Photonics.

Mateas, M., & Stern, A. (2004). A Behavior Language: Joint Action and Behavioral Idioms. In H. Prendinger & M. Ishizuka (Eds.), *Life-Like Characters: Tools, Affective Functions, and Applications* (pp. 135–161). https://doi.org/10.1007/978-3-662-08373-4_7

Mathieu, J., Maynard, M. T., Rapp, T., & Gilson, L. (2008). Team Effectiveness 1997-2007: A Review of Recent Advancements and a Glimpse Into the Future. *Journal of Management*, *34*(3), 410–476. https://doi.org/10.1177/0149206308316061

McEliece, R. J., MacKay, D. J. C., & Cheng, J.-F. (1998). Turbo decoding as an instance of Pearl's" belief propagation" algorithm. *IEEE Journal on Selected Areas in Communications*, *16*(2), 140–152.

McGrath, J. E. (n.d.). *Groups: Interaction and Performance* (Vol. 14). Retrieved from https://www.pearson.com/us/higher-education/program/Mc-Grath-Groups-Interaction-and-Performance/PGM102430.html

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Pentland, A. (2010). *Honest Signals: How They Shape Our World*. MIT Press.

Pfeffer, A. (2005). Functional specification of probabilistic process models. *AAAI*, 663–669.

Pfeffer, A. (2016). Practical Probabilistic Programming (Version 1st) (1st ed.). Greenwich, CT, USA: Manning Publications Co.

Pfeffer, A., Ruttenberg, B., & Kretschmer, W. (2016). Structured Factored Inference: A Framework for Automated Reasoning in Probabilistic Programming Languages. *ArXiv Preprint ArXiv:1606.03298*.

Pfeffer, A., Ruttenberg, B., Kretschmer, W., & OConnor, A. (2018). *Structured factored inference for probabilistic programming*. 1224–1232.

Pfeffer, A., Ruttenberg, B., Sliva, A., Howard, M., & Takata, G. (2015a). Lazy Factored Inference for Functional Probabilistic Programming. *ArXiv Preprint ArXiv:1509.03564*.

Pfeffer, A., Ruttenberg, B., Sliva, A., Howard, M., & Takata, G. (2015b). Lazy Factored Inference for Functional Probabilistic Programming. *ArXiv:1509.03564 [Cs]*. Retrieved from http://arxiv.org/abs/1509.03564

Qi, Y., Reynolds, C., & Picard, R. W. (2001). Machine for computer-user frustration detection via pressuremouse. *Proceedings of the 2001 Workshop on Perceptive User Interfaces.*, 1–5. ACM.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., … Lanctot, M. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489.

Sliva, A. L., Gorman, J., Voshell, M., Tittle, J., & Bowman, C. (2016). Combining cognitive engineering and information fusion architectures to build effective joint systems. In SPIE Defense+ Security (pp. 98310M-98310M). International Society for Optics and Photonics.

Soleymani, M., Asghari-Esfeden, S., Fu, Y., & Pantic, M. (2016). Analysis of EEG signals and facial expressions for continuous emotion detection. *IEEE Transactions on Affective Computing*, *7*(1), 17–28.

Tomprou, M., Kim, Y. J., Chikersal, P., Woolley, A. W., & Dabbish, L. A. (2019). Visual cues disrupt prosodic synchrony and collective intelligence in distributed collaboration. *Proceedings of Collective Intelligence 2019*.

Vidulich, M. A., Stratton, M., Crabtree, M., & Wilson, G. (1994). Performance-based and physiological measures of situational awareness. *Aviation, Space, and Environmental Medicine*.

Weyhrauch, P. W. (1997). *Guiding Interactive Drama* (PhD Thesis). Carnegie Mellon University, Pittsburgh, PA, USA.

Wilson, G. F. (2000). Strategies for psychophysiological assessment of situation awareness. *Situation Awareness Analysis and Measurement*, 175–188.

Woolley, A. W., Aggarwal, I., & Malone, T. W. (2015). Collective Intelligence and Group Performance. *Current Directions in Psychological Science*, *24*(6), 420–424. https://doi.org/10.1177/0963721415599543

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., & Malone, T. W. (2010). Evidence for a Collective Intelligence Factor in the Performance of Human Groups. *Science*, *330*(6004), 686–688. https://doi.org/10.1126/science.1193147

Xu, J., Slagle, J. M., Banerjee, A., Bracken, B., & Weinger, M. B. (2019). Use of a Portable Functional Near-Infrared Spectroscopy (fNIRS) System to Examine Team Experience During Crisis Event Management in Clinical Simulations. *Frontiers in Human Neuroscience*, *13*, 85.

Yun, K., Watanabe, K., & Shimojo, S. (2012). Interpersonal body and neural synchronization as a marker of implicit social interaction. *Scientific Reports*, *2*, 959.