INTEGRATED DYNAMICAL INTELLIGENCE FOR INTERACTIVE EMBODIED AGENTS

Eric Aaron, Juan Pablo Mendoza

Department of Mathematics & Computer Science, Wesleyan University, Middletown, CT, U.S.A.

Henny Admoni

Department of Computer Science, Yale University, New Haven, CT, U.S.A.

Keywords: Interactive embodied agents, Reactive intelligence, Hybrid reactive-deliberative agents, BDI agent modeling, Hybrid dynamical systems.

Abstract: For embodied agents that interact with people in time-sensitive applications, such as robot assistants or autonomous characters in video games, effectiveness can depend on responsive and adaptive behavior in dynamic environments. To support such behavior, agents' cognitive and physical systems can be modeled in a single, shared language of dynamical systems, an integrated design that supports performance with mechanisms not readily available in other modeling approaches. In this paper, we discuss these general ideas and describe how *hybrid dynamical cognitive agents* (*HDCAs*) employ such integrated modeling, resulting in dynamically sensitive user interaction, task sequencing, and adaptive behavior. We also present results of the first user-interactive applications of HDCAs: As demonstrations of this integrated cognitive-physical intelligence, we implemented our HDCAs as autonomous players in an interactive animated Tag game; resulting HDCA behavior included dynamic task re-sequencing, interesting and sensible unscripted behavior, and learning of a multi-faceted user-specified strategy for improving game play.

1 INTRODUCTION

In interactive applications such as video games or personal robotics, embodied agents should be adaptive and responsive to users. In this paper, we present an intelligence modeling framework that supports these qualities: Influenced by *dynamicist* cognitive science —the study of mind as a dynamical system rather than a discrete, computational one (Port and van Gelder, 1995; Spivey, 2007)— our *dynamical* intelligence model integrates the physical and cognitive sub-systems of an agent in a shared language of differential equations, providing a unified, dynamically sensitive substrate for behavior. In particular, we describe how *hybrid dynamical cognitive agents* (*HDCAs*) (Aaron and Admoni, 2009; Aaron and Admoni, 2010) can reflect these ideas of *integrated dynamical intelligence*, and to illustrate these ideas, we present the first user-interactive applications of HDCAs.

The design of HDCAs' cognitive systems is influenced unconventionally by the *belief-desire-intention* (*BDI*) theory of intention (Bratman, 1987) and its implementations (e.g., (Georgeff and Lansky, 1987) and

successors), which established that BDI elements (beliefs, desires, intentions) are an effective foundation for goal-directed intelligence. Unlike typical BDI agents, HDCAs' cognitive models interconnect BDI elements in a continuously evolving system inspired by *spreading activation* frameworks (Maes, 1989). Each BDI element in an HDCA is represented by an activation value, indicating its salience and intensity "in mind" (e.g., intensity of commitment to an intention), and cognitive evolution is governed by differential equations, with activation values affecting rates of change of other activations. HDCAs employ these dynamical cognitive representations on both *reactive* and *deliberative* levels, distributing goal-directed intelligence over both levels. For example, HDCAs can re-order task sequences simply by evolution of *dynamical intentions*, without propositional deliberation (Aaron and Admoni, 2009).

The physical systems of HDCAs —comprising the elements pertinent for navigation, i.e., *xy*-location, velocity, and heading angle— are also modeled by differential equations; for this paper, HDCAs' navigation intelligence is based on (Goldenstein et al.,

2001). Thus, physical and cognitive systems in HD-CAs are unified in the common language of differential equations, which is critical to the HDCA learning demonstrations in section 3.3.

Along with the continuous evolutions of cognitive activations and physical variables, discrete changes —e.g., transitions from one task or behavior to another— can occur when activations reach pre-specified thresholds. Thus, we formally model an HDCA as a *hybrid dynamical system* (*hybrid system*, for short) (Alur et al., 2000). Discrete task transitions also illuminate the roles of intentions and *sequencing intentions* (see section 2.1) in our HDCAs: Conceptually, sequencing intentions have activations that represent intentions to perform tasks in a temporal relationship (e.g., to perform X before Y), as distinct from standard intentions associated with individual tasks. (In this paper, we consider only HDCAs with one intention for each task.) Sequencing intentions are critical elements of *guards* —threshold conditions for when discrete transitions occur— in agents' hybrid systems, so dynamical task re-sequencing emerges organically from continuous cognition.

As a motivating example for the ideas in this paper, consider a child playing Tag, avoiding a player designated as *It* as well as other obstacles. She wants to accomplish many things before the game ends: actively protect a friend in the game for a while; actively try to become It and tag an adversary; and reach several *bases*, locations where she cannot be tagged by It. She can only do one of these tasks at a time, so she begins with an intended task sequence that achieves her goals, but as the game proceeds, she re-sequences tasks in response to her environment. Moreover, while playing, her behavior shows sensible subtleties, such as making decisions a bit more quickly when she's comfortably near her goal.

In demonstrations for this paper, such a player is implemented as an HDCA in an interactive, animated Tag game, illuminating the capabilities of integrated dynamical intelligence. In conventional agents, for example, task re-sequencing such as the child performs is deliberative, but in HDCAs, it arises from sub-deliberative cognitive evolution. Moreover, our integrated modeling enables moment-by-moment interactions among elements considered cognitive and those considered physical, which evoke micro-scale effects that can cascade into observable effects (see section 3.2). We also demonstrate how an HDCA can learn a multi-faceted Tag strategy, perhaps based on user input during a game, extending previous HDCA learning methods to this application domain.

2 HYBRID DYNAMICAL COGNITIVE AGENTS

HDCAs can be viewed as having *physical* and *cognitive* sub-systems, composed of the differential equations and variables describing the behavior conventionally considered physical or cognitive, respectively; BDI elements are thus considered cognitive, while *xy*-location and heading angle ϕ are physical. HDCAs are implemented by augmenting physical systems with cognitive BDI elements and their activation values. For this paper, cognitive activations are within $[-10, 10]$, where near-zero values indicate low salience and greater magnitudes indicate greater intensity of associated concepts—e.g., more active intentions represent more commitment to the related tasks. Negative values indicate salience of the opposing concept, so, e.g., a moderate desire to not cycle the bases and strong commitment to protect a friend could be encoded by value -3 on a desire for `runBases` and value 9 on an intention for `protect`.

Our HDCAs' cognitive activations are interconnected in differential equations. A partial cognitive system —with many equations omitted and terms elided in equations shown— is in equation 1, in which beliefs, desires, and intentions are represented by variables beginning with *b*, *d*, and *i*, and time derivative variables are on the left in each equation:

$$\dot{dRun} = -c_1 \cdot bAmIt + c_3 \cdot iRun + \dots \qquad (1)$$
$$\dot{iTag} = d_1 \cdot bAmIt - d_3 \cdot dRun + d_4 \cdot iTag + \dots$$
$$\dot{iRun} = -e_1 \cdot bAmIt - e_2 \cdot dTag + e_5 \cdot iRun + \dots$$

This illustrates interconnectedness: Elements have *excitatory* or *inhibitory* influence on activations by increasing or decreasing derivatives. In equation 1, variables stand for activations of cognitive elements such as the desire to run around the bases (*dRun*) and the belief that the agent is It (*bAmIt*); coefficients represent the impacts of the connections between elements.

2.1 Our HDCA Implementation

Because HDCA behavior consists of switching among multiple, continuous behaviors, our HDCA implementation is based on a *hybrid automaton* (Alur et al., 2000), a state-transition model of hybrid systems. Each hybrid automaton has discrete *modes* representing individual behaviors or tasks, each having differential equations that govern variables' evolution in that mode, and *guard* constraints describing when mode transitions occur (see Figure 2). We straightforwardly implemented and simulated our HDCAs as hybrid systems in MATLAB, with modes as functions containing guards for mode transitions and dynamical

systems for agent evolution. Within the hybrid automaton structure, our HDCAs also include the structures described below for dynamical intelligence.

2.1.1 Task Sequencing

In addition to standard intentions, our HDCAs have *sequencing intentions* for dynamic task sequencing. In this paper, we implement sequencing intentions as pairs; the activation of sequencing intention (A, B) is the difference in activations of corresponding intentions, $iA - iB$, representing the commitment to performing action A before action B. To determine task sequence in an HDCA with actions $\alpha_1 \ldots \alpha_k$, for each action α_i, we sum activations on the k sequencing intentions with α_i in the first position; the descending order of these associated sums induces a sequence on the actions. Sequencing intentions could in principle encode other concepts, but this suffices to illustrate integrated intelligence in HDCAs.

Activations on intentions and sequencing intentions evolve over time, so at any time, a new action α_i might attain maximum priority and re-sequence tasks. When a task is finished, intentions and sequencing intentions are altered to reflect that, and the agent continues in the new maximal-priority action.

2.1.2 Cognitive-physical Integration

Because of integrated intelligence in HDCAs, any variable, cognitive or physical, could affect any other variable. To illustrate how any physical element in HDCAs could subtly affect any aspect of cognitive state, we demonstrate an extreme case: physical elements considered "involuntary" affecting cognitive elements considered "subconscious." In particular, we encode that cognitive dynamics, as specified by differential equations governing activation evolutions, should accelerate slightly when the agent is more "relaxed," i.e., near a target location and not turning rapidly. To do this, we construct a physical-cognitive multiplier *pcm* so that physical values can affect activations of BDI elements: values of *pcm* range from 1 to $1 + p$, where p is a designer-specified parameter, and intensify cognitive evolution by multiplication with time derivatives, e.g., $iTag = pcm \cdot i\dot{T}ag \cdot timeStep + \ldots$, instead of $iTag = i\dot{T}ag \cdot timeStep + \ldots$.

The *pcm* function in our demonstrations begins with function $e^{-k_1(|\dot{\phi}|+d)}$ of angular velocity $\dot{\phi}$ and current distance d from the target, so that when $(|\dot{\phi}| + d)$ is close to 0, the function value is close to 1, and as $(|\dot{\phi}| + d)$ gets larger, the function value gets closer to 0. Designer-chosen constant $k_1 > 0$ controls the rate at which values approach 0 as $(|\dot{\phi}| + d)$ grows. Building upon this, to get our desired effect, we chose:

$$pcm = 1 + p \left(\frac{2}{\pi} \sin^{-1} (e^{-k_1(|\dot{\phi}|+d)}) \right)^{k_2} \quad (2)$$

This enables a boost as agents near targets and stop turning, with much less effect outside of the desired range for $(|\dot{\phi}| + d)$; it could be changed for different effects. (Parameter values for our demonstrations are available at (Aaron et al., 2011).)

3 EXPERIMENTS

As a demonstration domain for our HDCAs, we implemented animated interactive Tag games containing a *user* player and two kinds of autonomous players: *simple Tag* agents, HDCAs with limited intelligence; and *cognitive Tag* agents, with more extensive dynamical intelligence. Agents intuitively interact with other agents and the user: Each It agent pursues some non-It player; each non-It agent avoids It players and views non-It players as safe. To make the game more adversarial, agents also have slight anti-user biases, and two players at a time are It. The field of play (Figure 1) is a square with *bases* near the corners, obstacles between bases, and other players. Players touching base cannot become It, but they cannot stay on base too long before moving away. Players are penalized for touching an obstacle.

3.1 Autonomous Tag Players

A risk-averse non-It player could simply run clockwise from base to base, hoping not to be forced into a position to get tagged. A simple Tag agent (*STA*) executes that strategy. When an STA *S* becomes It, it chooses from two possible *It-actions*: chasing the user; or chasing an agent. If another It player is chasing the user, *S* joins the chase; if not, *S* tries to tag the closest non-It agent. In addition, if *S* engages in one of these It-actions for a long time, "boredom" sets in, represented by attenuation on the corresponding intention activation, so *S* will eventually switch to the other It-action. Unlike a cognitive Tag agent (*CTA*), an STA's cognitive structure is a very simple dynamical intention-based system, straightforwardly supporting only the design and behavior noted above.

A cognitive Tag agent more fully demonstrates dynamical intelligence and cognitive-physical integration; see Figure 2 for its mode-level architecture and BDI elements. When a CTA *C* is not It, it will try to accomplish all of the following actions before the game ends: runBases, cycling the bases (as STAs do); getMitten, retrieving its mitten (which the poor agent drops in every game); protect, spending time protecting a friend from being tagged; and
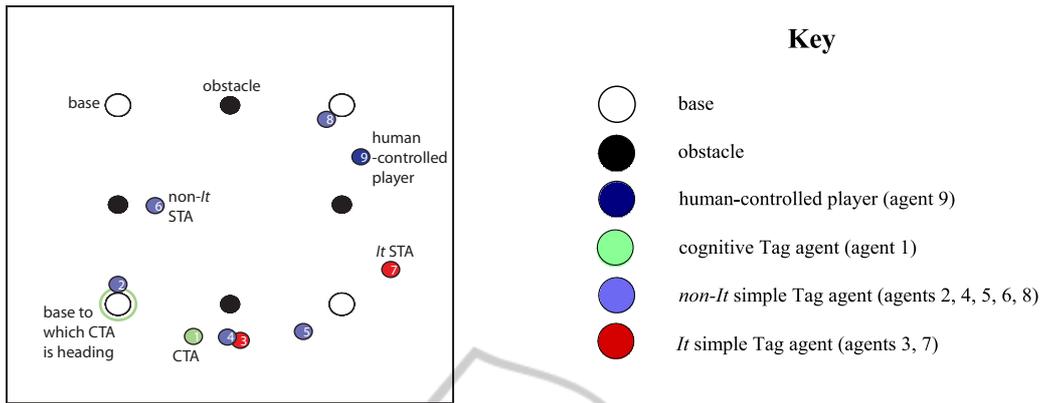
Figure 1: An annotated screen shot of our Tag game, illustrating field layout and players on the field, including simple Tag agents (*STAs*) and cognitive Tag agents (*CTAs*). Color variations distinguish entities, as does the convention that players, human-controlled or automated, are numbered, while bases and obstacles are not. The program indicates the base to which the cognitive Tag agent is heading by drawing a light circle around that base. See supplementary website (Aaron et al., 2011) for further details of color, notation, and function of elements in our animated Tag games.

Figure 2: The mode-level architecture and BDI elements of a cognitive Tag agent. Each mode also has self-transitions, omitted by convention to avoid visual clutter.

`readyToTag`, trying to become It and tag an adversary. The `getMitten` action is implemented by selecting a time when, wherever *C* is, its mitten drops; soon after, *C* finds the mitten's location, and activations on BDI elements evolve until, in general, mitten-retrieval becomes *C*'s highest priority. To enable `protect` and `readyToTag`, *C* has beliefs of affinities for each player in the game, and *C* will protect a non-It player with maximal affinity during `protect`, and pursue a non-It player with minimal affinity during `readyToTag`. These non-It actions are dynamically re-sequenced, based on time pressure, affinities, and proximity to locations (e.g., a base, an adversary).

When a CTA is It, it either follows through on a `readyToTag` action or selects between pursuing the user or an automated player, exactly as an STA would.

3.2 Experiments

We performed various demonstrations of HDCA intelligence. Some were proofs that our ideas work as expected in sensible environments. One showed that CTAs can follow instructions, e.g., that cognitive evolution need not prevent them from completing tasks in accord with initial intention activations. Another illustrated physical-cognitive multiplier *pcm*: In a contrived situation, two cognitively identical CTAs were equally near a target, one facing the target, the other facing away; the CTAs then changed heading angle as usual, but not position. As a result, the CTA facing the target had higher *pcm* values and changed task-modes faster than the other CTA. These are all expected demonstrations of proper performance; for more details, see (Aaron et al., 2011).

In other experiments, specific values were varied in controlled environments, to investigate particular effects. As examples, we simulated many game segments with identical CTA C; initially, C's intentions implied task order [readyToTag, runBases, protect, getMitten]. Across simulations, two factors varied —when C dropped its mitten; and when C was tagged by the user (which was automated, for replicability)— to illuminate dynamicist effects in game play. As *mitten-drop* grew later with *get-tagged* held constant, for example, the time at which C moved from readyToTag into runBases was not affected, but the time at which C then entered protect tended to get *earlier*. In addition, for particular values of *mitten-drop* and *get-tagged*, C entered protect mode —in which movement is not obstacle-avoidant— at an inopportune moment and ran straight into an It player. This sequence of events and ensuing cascade of effects illustrates how engaging, unscripted behavior can emerge from the continuous-level variations supported by our HDCA framework.

For additional details and other results, including natural game play, see (Aaron et al., 2011).

3.3 Learning

To support learning from unpredictable users, cognitive-physical integration is maximally flexible: All physical and cognitive variables can be interconnected, and any connection can be modified by learning. For our demonstrations, HDCAs are trained by reinforcement learning similar to that in (Aaron and Admoni, 2010), which requires heuristics selecting which connections to modify during learning and criteria for when learning is complete. Learning occurs without interrupting interactive applications.

As preparation, we first determined *control condition* behavior by letting a game play extensively (for more than 8000 simulated seconds), with an automated user for replicability. In this game setup, when a cognitive Tag agent C_{ctrl} became It, C_{ctrl} would almost always tag some other player in less than 25 simulated seconds (average: 12.85 seconds). In addition, the value a_{ctrl} of the average number of bases reached per execution of the runBases behavior, over the full game, was $a_{ctrl} = 4.01$ (see Figure 3).

Based on this, we demonstrated a CTA C learning from a simulated user request to change one aspect of game play without affecting another; it exemplifies an arbitrary user choice, unrelated to agent design and substantively changing control behavior. The goal had two components: *speed change*, requiring *speed-only* learning; and *base-running maintenance*, requiring *speed-and-bases (SB)* learning.

- *Speed change*: After becoming It, C should optimally tag some other player between 25 and 45 seconds later. Speed-only training (and thus partial SB training, see below) occurs when C transitions out of chase mode. If the time C was It is outside of the desired range (25–45 seconds), C is trained to become slower or faster, as appropriate, by a factor that depends on exactly how far outside of the desired range C was It.

- *Base-running maintenance*: Despite the effects of speed-only learning, C should only minimally change the value a_C of the average number of bases reached during each runBases behavior. SB training occurs when C transitions out of runBases mode: a_C is updated, and coefficients in cognitive differential equations are altered to train C to approach the desired, control value of 4.01 in the future. As a partial example, if $a_C < 4.01$, coefficients in the differential equation governing *iRun* are altered so that C tends to remain longer in runBases, encouraging greater a_C in the future. The amounts altered depend on values such as the velocity of C when training occurs, exemplifying cognitive-physical integration: Values of physical variables affect cognitive adjustments.

To focus our demonstrations, the connections modified during training were pre-selected, though the adjustments were autonomous. More details on the learning process are available at supplementary website (Aaron et al., 2011).

Our tests demonstrate C successfully learning integrated cognitive-physical behavior during game play: C slowed to spend more time as It before tagging another player (average time: 32.62) while also maintaining a bases average of $a_C = 4.21$, very close to 4.01. Figure 3 illustrates the effects of SB learning on a_C and base-running performance. Additionally, Figure 3 shows that speed-only learning without full SB learning resulted in a value of $a_C = 2.19$ in otherwise identical game play, suggesting the importance of integrated learning for the desired goal.

4 CONCLUSIONS

This paper describes the integrated cognitive-physical intelligence underlying our HDCAs, and it presents the first applications of HDCAs in interactive scenarios. Agent cognition in our HDCAs is based on continuously evolving activations of BDI-based cognitive elements, enabling a model that unites cognitive and physical 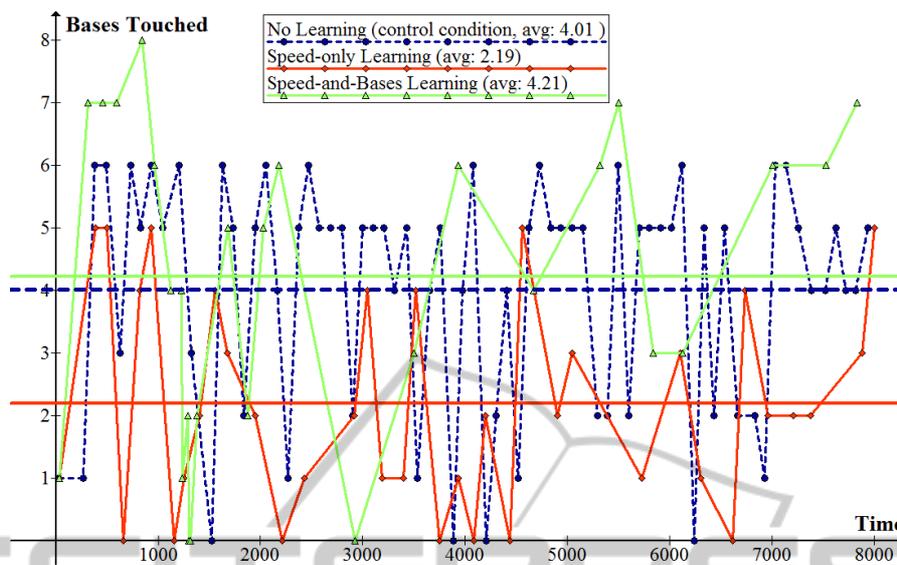intelligences in a single system; as a result,