Biologically Inspired Cognitive Architectures xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Biologically Inspired Cognitive Architectures

journal homepage: www.elsevier.com/locate/bica

Research article

The performance comparison problem: Universal task access for cross-framework evaluation, Turing tests, grand challenges, and cognitive decathlons

Vladislav D. Veksler [a,*], Norbou Buchler [b], Christian Lebiere [c], Don Morrison [c], Troy Kelley [b]

[a] DCS Corp, HRED, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, USA
[b] HRED, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, USA
[c] Department of Psychology, Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 9 June 2016
Revised 29 September 2016
Accepted 2 October 2016
Available online xxxx

Keywords:
Grand challenge
Cognitive decathlon
Turing test
Performance comparison
Simulation
API
Standards

ABSTRACT

A driver for achieving human-level AI and high-fidelity cognitive architectures is the ability to easily test and compare the performance and behavior of computational agents/models to humans and to one another. One major difficulty in setting up and getting participation in large-scale cognitive decathlon and grand challenge competitions, or even smaller scale cross-framework evaluation and Turing testing, is that there is no standard interface protocol that enables and facilitates human and computational agent "plug-and-play" participation across various tasks. We identify three major issues. First, human-readable task interfaces aren't often translated into machine-readable form. Second, in the cases where a task interface is made available in a machine-readable protocol, the protocol is often task-specific, and differs from other task protocols. Finally, where both human and machine-readable versions of the task interface exist, the two versions often differ in content. This makes the bar of entry extremely high for comparison of humans and multiple computational frameworks across multiple tasks. This paper proposes a standard approach to task design where all task interactions adhere to a standard API. We provide examples of how this method can be employed to gather human and computational simulation data in text-and-button tasks, visual and animated tasks, and in real-time robotics tasks.

Introduction

Performance comparison is a major focus in the fields of Artificial Intelligence and Cognitive Science. Grand challenges in the fields of computational cognition focus on contrasting agent/model performance within a given task environment, or across a set of tasks (i.e. a cognitive decathlon). Oftentimes human-like performance is the yardstick of evaluation for machines, either in the context of the Turing test,[1] or for better predictions of human behavior. Whether the focus is on benchmarking contrasting cognitive systems against one another or in relation to human performance, the seemingly minor practical nuisance of dealing with varied and often idiosyncratic task interfaces has become a major limiting factor for theoretical progress in the field.

It is often impossible for humans to participate in simulation tasks designed for computational agents, and it is difficult to connect a computational system to task software designed for human use. Moreover, it is often difficult to connect a computational system to a software task environment designed for use by another computational system. Henceforth, we refer to this as the *performance comparison problem*.

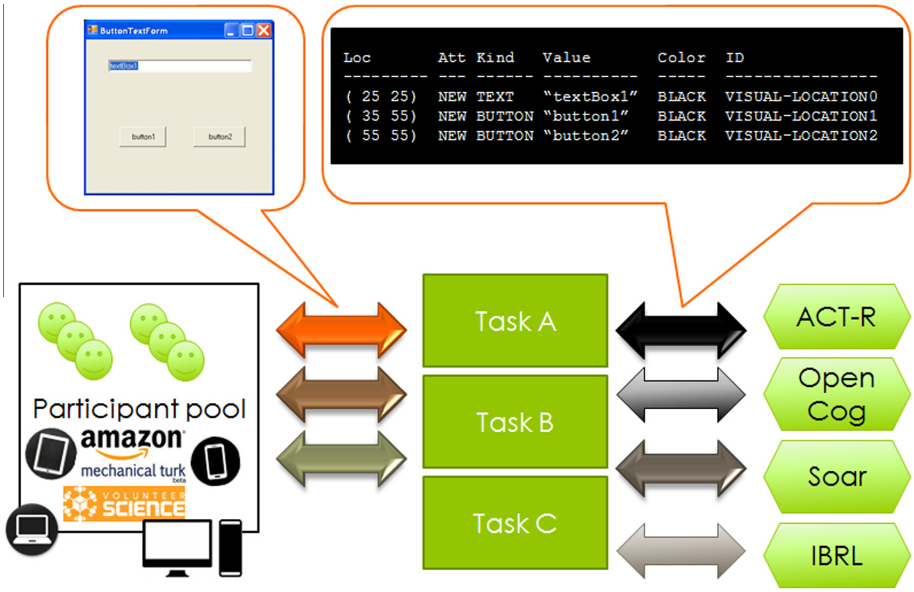
Ideally, we would like to reuse the same task software and collect data in the same format from computational participants, regardless of computational framework and theory (e.g. symbolic/-subsymbolic), and from human participants, regardless of operating system and display type (e.g. mobile/desktop). Such task reuse and cross-user, cross-agent task access would make it easier to run computational simulations and behavioral studies using the same task software, compare human and computational participant performance, and develop task environments where human and computer agents can interact interchangeably (e.g. human-AI teaming, recommender systems, multi-agent training scenarios).

More to the point, the ability to develop such *universal access* software would promote a growing library of plug-and-play tasks. Individual research efforts would benefit greatly from the

* Corresponding author.
E-mail address: vdv718@gmail.com (V.D. Veksler).

[1] We use the term Turing test to mean any test where computational agent performance is evaluated with the goal of being indistinguishable from that of a human. This is a more generalized definition than the original version of the Turing test, which focused specifically on computational agent performance in a verbal chat session.

http://dx.doi.org/10.1016/j.bica.2016.10.003
2212-683X/© 2016 Elsevier B.V. All rights reserved.



```
Loc        Att Kind   Value       Color  ID
---------  --- ------ ----------  -----  ----------------
( 25 25)   NEW TEXT   “textBox1”  BLACK  VISUAL-LOCATION0
( 35 55)   NEW BUTTON “button1”   BLACK  VISUAL-LOCATION1
( 55 55)   NEW BUTTON “button2”   BLACK  VISUAL-LOCATION2
```

Fig. 1. Status quo in task interfacing. Each human user-type (mobile/desktop) and each computational agent/model type requires a separate effort in interface development. Human participants are rarely able to parse simulation-targeted API's, and computational agent/model frameworks are rarely able to parse GUI's intended for human use, or API's intended for use by dissimilar compuational frameworks.

availability of standard-access tasks suited for their respective research questions, and prior data from human and computational participants for these tasks. Larger-scale computational cognition grand challenges would be easier and more affordable to set up as incremental collections of off-the-shelf tested tasks, sometimes with readily available human performance data. Perhaps more importantly, standard access task software would encourage more research groups to participate in grand challenges and cognitive decathlons by lowering the barrier to entry.

There are two major hurdles on the path to this vision. First, tasks designed for human participants often include many task-irrelevant features, used mostly for visual appeal. Computational cognition researchers often require separation between task-essential and non-essential interface information. Such separation does not exist in most task software. Second, there is no standard API[2] dictating how computational agents can interact with task software. The bar to entry for connecting a number of different computational systems to a number of different task environments, all with different APIs, is very high. To put it plainly, it is often the case that each human device type and each computational agent/model framework requires its own specialized interface to each individual task, exponentially ratcheting up the effort required for progress in the field (see Fig. 1). More fundamentally, handcrafted interfaces introduce countless degrees of freedom, making it all but impossible to compare computational frameworks on an equal footing.

In this paper we propose a standard approach to task design that focuses on task functionality (i.e. affordances) rather than non-task-essential visualization choices. In this approach, task interactions adhere to a standard API. Computational agents/models interact with the API directly, and human participants interact with the task via the same API, employing customizable visualization templates to make the task visually appealing. We propose a minimal, web-friendly, JSON-compatible API called Simple Task-Actor Protocol (STAP). We provide examples of STAP use in text-and-button tasks, visual and animated tasks, and in real-time robotics tasks. We argue that this approach to task development does not have to become universal to provide a boost for growth in the field of performance comparison. Each task adhering to the proposed methodology will add to a growing collection of off-the-shelf plug-and-play software that may be employed in grand challenges, cognitive decathlons, and individual simulation efforts.

Grand challenges and cognitive decathlons

Competitions and grand challenges for computational cognition systems and artificial general intelligence have been a primary means to motivate and galvanize the research community to solve ambitious scientific and engineering challenges or hard problems. A major criterion for grand challenges in the field is that they should include a range of non-gameable problems and test a decomposition of functional capabilities (Brachman, 2006). A popular term describing a range of problems requiring varying general cognitive capabilities is a *cognitive decathlon*.

The Biologically Inspired Cognitive Architectures international research community, which currently holds an annual conference and publishes a journal with Elsevier, was originally galvanized via a prospective DARPA grand challenge and follow-up cognitive decathlon proposals (Mueller, 2010; Mueller, Jones, Minnery, & Hiland, 2007; Samsonovich, 2012). Grand challenges in robotics aside,[3] there have been numerous ideas for proposals and successful competitions in the field of computational cognition, e.g. Gluck et al. (2014), Lebiere, Gonzalez, and Warwick (2010), Lebiere and Bothell (2004) and Pew, Gluck, and Deutsch (2005). However, it is difficult to get researcher buy-in for each new competition, as each proposed task requires a complete retooling of the interface between the model/agent framework and the task API.

There is no library of off-the-shelf task environments adhering to a standard API. Thus, each competition proposal is riddled with difficulties in choosing the best task(s). Achieving consensus among organizers on the type of task proves to be a major

[2] API (application program interface) is a protocol that specifies how software components should interact.

[3] Robotics competitions tend to capture the public imagination, but often limit the progress in general computational cognition due to overly high focus on hardware, sensors, and locomotion. We argue this point at length in the *What's wrong with the physical world* section below.