Attention Guidance Agents for the MATBII cockpit task

Szonya Durant[1], Benedict Wilkins[2], Callum Woods[1], Emanuele Uliana[2], and Kostas Stathis[2]

[1] Department of Psychology
[2] Department of Computer Science
Royal Holloway, University of London, Egham TW20 0EX, UK
{szonya.durant,benedict.wilkins, callum.woods, emanuele.uliana,kostas.staths}@rhul.ac.uk

Abstract. We show how an implementation of the MATBII cockpit simulator embedded in an agent environment that supports eye tracking, can act as an experimental platform that will be beneficial for future attention guidance experiments in complex multi-task interfaces. We also explore how such a system may be useful for improving task performance, by simulating users with agents to demonstrate how the system might work in practice with more complex examples of user behaviour.

Keywords: agent environment, eye tracking, interface, workload.

1 Introduction

We demonstrate a framework for attention guidance in multi-task interfaces based on cognitive agent environments [1]. With this approach, interactions between agents and the objects of the display are formulated as events that change the agent environment representing the deployment of a practical application [2]. Our objective is to demonstrate how by monitoring the state of the various interface components and incoming user input data (including eye tracking data) construed as observations, the agents can process these to guide a user, while taking into account important considerations relating to user attention [3].

We exemplify the framework by developing methods for attention guidance in MATBII [9], a computer-based task battery designed to facilitate research in human multiple-task performance with consideration for the effects of automation. As shown in Fig.1, the tool includes four component tasks: system monitoring, tracking, communications monitoring, and resource management, while a scheduling window provides preview of anticipated workload, and component tasks can be automated or manual. The system has been used in numerous experiments of human attention, especially in the domains of aviation and aerospace.

The contribution of our system is that it makes use of gaze location (a proxy for spatial attention), allowing agents to use this information to help users allocate their limited cognitive resources. To this end we reproduce and improve some aspects of MATBII with our own simple simulation of a cockpit-based task

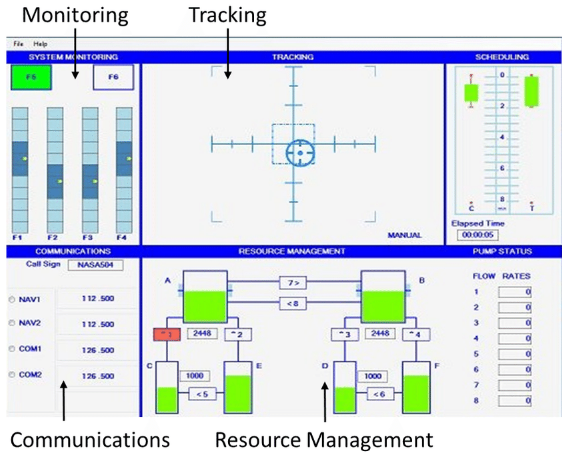

Fig. 1. The MATBII system with sub-tasks labelled.

space. We then embed this functionality in an agent environment and use cognitive agents to guide user monitoring. Our agent system, is an instance of a principled multi-agent framework that can provide attention guidance in other domains, independent of MATBII.

2 Integrated Cognitive User assistance with Agents

Although an open source implementation of MATBII with eye-tracking options is available [4], we found it better suited to our purpose of combining the interface with an agent environment to develop our own. More specifically, we have implemented a subset of the tasks from the original MATBII but with some functional improvements that are essential for experimentation. The resulting system, which we refer to as ICU (Integrated Cognitive User) [5], allows for display changes and eye movements to be monitored externally via a bi-directional API. This API is event-based and makes ICU suitable to be easily embedded in an agent environment, but also for experimental settings that do not require agents.

To support eye tracking, ICU provides a wrapper around the PsychoPy library [8], which enables any eye tracker supported by the library to be used with ICU(we assume that the eye tracker is already calibrated). The system was tested using a USB screen based X2-30 Tobii eye tracker, sampling at 30Hz on average. Raw gaze coordinates are filtered using an I-VT filter with standard moving average as specified in [7], coordinates are classed as fixation (eyes are stationary) or saccade (eyes are moving and thus unable to take in information).

ICUa is ICU extended with agents implemented in PyStarWorlds, an agent environment that supports Python agent applications, see Fig. 2 for a reference

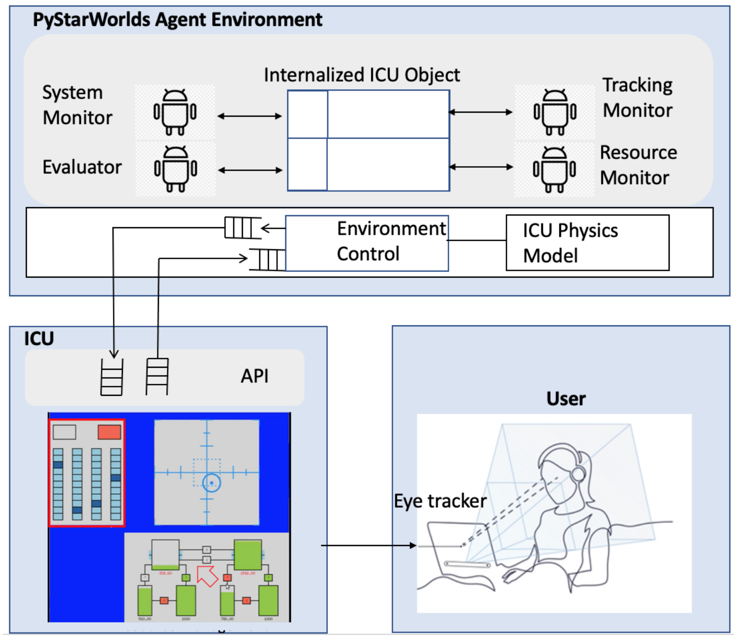

Fig. 2. The ICU system with agents.

architecture. PyStarWorlds is a specialised version of the GOLEM framework described in [6], and implemented as an event-processing system under a publish/subscribe model [7]. PyStarWorlds also allows the development of agents, whose behaviours are specified in Python using condition-action rules following a teleo-reactive execution model [6]. In this context, ICU is internalised as an object in the environment by the API it exposes, so that its state can be perceived and acted upon by agents. A group of four agents are deployed to guide the user's attention. We assign one agent to each of the first three application simulator tasks: system monitoring, resource management and tracking. Each of these agents subscribes to task specific events enabling them to perceive relevant information about the application simulator's current state. These agents further subscribe to user input, including saccade or fixation location, and to receive communications from the other agents. Communication allows agents to coordinate their feedback so as to be most helpful to the user. The effect of agents' actions is to modify the application simulator interface i.e. to draw an overlay. We currently consider actions with two kinds of feedback (a) highlighting a particular sub-task and (b) draw an arrow at the current gaze location that points in the direction of a component that needs urgent attention. The fourth agent, acting as an evaluator, monitors user performance using specific performance metrics.

3 Flexible experimentation

Our system is available to use (see [5]). The work provides a basis for future researchers wishing to perform controlled experiments, including synthesising and testing various types of complex human behaviour. For this purpose, ICU supports event schedules to be specified in a number of ways, including probabilistically, allowing for easy randomization, and various display options and system behaviour to be configurable. Also, by default, ICU provides an output file containing all events, including gaze positions. Similarly, ICUa supports 'user' agents. Such agents directly observe events from the ICU system and, according to the way they are programmed, can emulate user input e.g. mouse clicks, key input and eye movement. We will demonstrate how such behaviours are evaluated against specific experimental metrics, such as time that a main fuel tank is out of the acceptable range in the resource management task, or time that warning lights are in an incorrect state in the system monitoring task. In this way, ICUa provides insights into how the system might perform before deploying it in actual user experiments.

4 Conclusions

Our intentions with this work is primarily to emphasize the flexibility of multi-agent environments for experimenting with eye-tracking and multi-display interfaces. In this setting, the eye-tracker and the displays are embedded as environment objects that can be monitored by agents that provide advice when a user's attention deviates from monitoring norms. This demonstrator shows that our platform brings new scope for experiments in human factors research owing to more flexible manipulation of the task space, the ability to collect eye tracking data easily and in real time. We found that the modularity of an agent-based approach eases the process of experimentation and provides some unique benefits for creating a system that is extensible, reproducible and transparent. Ultimately we hope this serves as a test-bed for the future development of agent-based user attention guidance.

Acknowledgements

The work for developing this demonstrator was supported by a Human-Like Computing EPSRC Network+ Kickstart grant. Thanks to Suzy Broadbent and Lisa Boyce for discussions and support.

References

1. Bromuri, S., Stathis, K.: Situating Cognitive Agents in GOLEM. In: Engineering Environment-Mediated Multi-Agent Systems. pp. 115–134. Springer Berlin Heidelberg (2008). https://doi.org/10.1007/978-3-540-85029-8_9

2. Bromuri, S., Stathis, K.: Distributed Agent Environments in the Ambient Event Calculus. In: Gokhale, A.S., Schmidt, D.C. (eds.) Proceedings of the Third ACM International Conference on Distributed Event-Based Systems, DEBS 2009, Nashville, Tennessee, USA, July 6-9, 2009. ACM (2009)
3. Carrasco, M.: Visual attention: The past 25 years. Vision research **51**(13), 1484–1525 (2011). https://doi.org/10.1016/j.visres.2011.04.012
4. Cegarra, J., Valéry, B., Avril, E., Calmettes, C., Navarro, J.: OpenMATB: A Multi-Attribute Task Battery promoting task customization, software extensibility and experiment replicability. Behavior Research Methods pp. 1–11 (2020). https://doi.org/10.3758/s13428-020-01364-w
5. Durant, S., Wilkins, B., Woods, C., Uliana, E., Stathis, K.: The ICU web-site. https://dicelab-rhul.github.io/ICU/ (Dec 2020)
6. Nilsson, N.J.: Teleo-reactive programs for agent control. J. Artif. Int. Res. **1**(1), 139–158 (Jan 1994). https://doi.org/10.1613/jair.30
7. Olsen, A.: The Tobii IVT Fixation Filter Algorithm description (2012)
8. Peirce, J., MacAskill, M.: Building experiments in PsychoPy. Sage (2018)
9. Santiago-Espada, Y., Myer, R.R., Latorella, K.A., Comstock Jr, J.R.: The Multi-Attribute Task Battery II (MATB-II) software for human performance and workload research: A user's guide (2011)