Operationalizing Operational Logics: Semiotic Knowledge Representations for Interactive Systems

Joseph C. Osborn
University of California at Santa Cruz
jcosborn@soe.ucsc.edu

Abstract

All projects in AI begin by selecting or devising knowledge representations suitable for the project's functional requirements. Interactive systems (including games) have *semiotic* considerations on top of their functional requirements: they must be legible to users, players, and their own designers. AI working within or around interactive systems must acknowledge and support the concerns of human users. These concerns are generally phrased as inductive bias or domain knowledge and handled in an *ad hoc* way; I argue that it is possible and useful to represent them explicitly within a unifying approach. This work refines and extends *operational logics*, an interpretive framework describing how interactive systems communicate their own mechanisms to users. Making this move yields formal notations for interactive systems that are useful for humans and machines, with applications in modeling, verification, general game-playing, reverse-engineering, and automatic self-documentation.

1 Introduction

The first step in designing an AI system is to decide on the problem being solved. This may sound glib, but practitioners and researchers have long known that picking the right small world—the correct problem specification and knowledge representation—is vital for the system's success [Agre, 1997]. Typically, we approach this meta-problem by phrasing our domain in terms of well-understood *semantics* like state transition systems, first-order logic, or Bayesian statistics.

The focus of my work is interactive systems, including but not limited to games (the sort meant for humans, not economists or logicians). Games may include AI systems like virtual agents and story generators, but games may also feature AI in their design process (just as AI techniques play a major role in the design of integrated circuits). We want to perform many of the same analysis tasks on interactive systems and their AI components as we do for non-interactive applications: safety and progress checking [Smith *et al.*, 2010], parameter or controller synthesis, recovering specifications from concrete behaviors [Summerville *et al.*, 2017], and so on. Moreover, we also need to check non-functional requirements: Is necessary information presented to the user before its use is required (and does the game in fact require that information)? Is the game or interaction too difficult?

We could idealize the player and work strictly on functional requirements, but model checking and the other general problems are still challenging. The properties we care about in interactive systems generally concern complete interaction histories starting from an initial configuration; a game that runs at sixty frames per second will rapidly exhaust loop unrolling and bounded model checking. We usually get around this by *ad hoc* game- (or genre-) specific approaches and abstractions, but this means the wealth of tools available in AI and program analysis are off-limits to most practitioners.

Fortunately, interactive systems are not *just* state transition systems! Some inductive bias generalizes across large classes of interactive systems, and identifying that could lead to more useful knowledge representations. Besides functional requirements and the program code that implements them, interactive systems are characterized by *semiotic* considerations: they must be legible to users, players, and indeed their own designers. My project is to find a general approach to choosing and devising knowledge representations for interactive systems grounded both in semantics *and* semiotics.

2 Contributions

This work refines and operationalizes *operational logics* (OLs), an interpretive framework describing how interactive systems communicate their workings to their users (here, "logic" is used loosely as this theory comes from media studies; compare *formal logic*). An operational logic combines an abstract process (e.g. agents that change behaviors over time, resolving collisions between embodied agents, and discrete transactions of numeric resources) with a strategy for communicating that process's behavior to a user/player (respectively, visual feedback for the current agent state, projections of bounding boxes onto a 2D plane, or animated readouts of values and flows) [Mateas and Wardrip-Fruin, 2009]. Since these processes and strategies can be enumerated [Osborn *et al.*, 2017] and are often shared extensively across games, we can build AI that reifies and operates simultaneously at the level of underlying abstract model *and* what information is presented to users and how.

Concretely, I am enumerating and formalizing the most

Entity-state	Networks of finite state machines
Resource	Petri nets, multiset rewriting
Collision	Qualitative/quantiative spatial constraints
Physics	Switched piecewise affine systems

Table 1: Some operational and corresponding formal logics

common operational logics and communicative strategies[1]. Specifically, for each operational logic I find a formal logical theory suitable for compositional modeling and analysis (see Tab. 1). This reduction does not eliminate the human-relevant semiotic aspects of the original systems but carries them through to the specification and modeling language.

As an example, the agents in action games combine entity-state, physics, and collision logics. Hybrid automata [Alur *et al.*, 1995] likewise combine finite state machines with switched systems of differential equations and event generators. I have leveraged this in three main projects.

The first is the modeling language and model checker HyPED. Defining game characters in terms of hybrid automata admits a variety of approaches: applying existing model checkers and controller synthesis algorithms; hybrid planning; and efficient discrete program synthesis.

Hybrid automata are also a useful representation for machines. Colleagues and I have work in publication for learning parameters (flows and velocity reset values) of automata with known structures: specifically, reverse-engineering characters' *jumping* behaviors automating the manual approach taken in [Fasterholdt *et al.*, 2016]. This works by dynamic analysis of visual features obtained from semi-automatic *experiments* on a system with controllable inputs and observable behavior (inspired by [Murphy, 2016]).

The less constrained case of learning both automaton structure and flows has been accepted to IJCAI. In CHARDA, we use similar visual features but do not assume a controllable system; we go from a timed sequence of exogenous inputs and observations to an automaton which covers the witnessed behavior. Instead of being fixed to one type of game rule, CHARDA works with a broader set of operators chosen from entity-state, collision, and physics logics, and uses statistical methods to account for the more open-ended domain. These projects in specification recovery used knowledge representations founded in operational logics to improve on the state of the art by considering both the role of human interaction and the behavior of the simulated objects.

Besides OLs, games also build up complex *playable models* of e.g. space, combat, cooking, and romance [Osborn *et al.*, 2015]. I have been involved in some preliminary work on making rhetorical arguments about games' rules starting from the standpoint of OLs [Martens *et al.*, 2016], but this needs a fuller account of cultural knowledge.

3 Future Work

Learning more about hybrid planning is my immediate goal. I have tried a few approaches to model checking HyPED games, and would like to learn about some continuous-time techniques that don't rely on non-linear programming. Capturing more of games' connotative visual meaning (e.g., spiky or burning objects are probably to be avoided) is another key milestone and something I have explored very little.

My research goal is compositional and modular game analysis. Compositional analysis and automated abstraction of games are open problems for me, and I would be very interested to learn how the planning community approaches them.

Generally, I aim to continue developing each of the projects and the agenda described in this abstract towards broader domains of games. My goal for the consortium, if accepted, is to become better-integrated with the broader AI community and to be pointed towards ways in which my work might be able to influence AI outside of games.

References

[Agre, 1997] Philip Agre. *Computation and Human Experience*. Cambridge University Press, 1997.

[Alur *et al.*, 1995] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A Henzinger, P-H Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 1995.

[Fasterholdt *et al.*, 2016] Martin Fasterholdt, Martin Pichlmair, and Christoffer Holmgård. You say jump, i say how high? operationalising the game feel of jumping. In *Proceedings of the First International Joint Conference of DiGRA and FDG*, 2016.

[Martens *et al.*, 2016] Chris Martens, Adam Summerville, Michael Mateas, Joseph Osborn, Sarah Harmon, Noah Wardrip-Fruin, and Arnav Jhala. Proceduralist readings, procedurally. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.

[Mateas and Wardrip-Fruin, 2009] Michael Mateas and Noah Wardrip-Fruin. Defining operational logics. In *Proceedings of the 2009 Digital Games Research Association Conference*, 2009.

[Murphy, 2016] Tom Murphy, VII. The glEnd() of Zelda. *SIGBOVIK*, April 2016.

[Osborn *et al.*, 2015] Joseph C Osborn, Dylan Lederle-Ensign, Noah Wardrip-Fruin, and Michael Mateas. Combat in games. In *Proceedings of the 10th International Conference on the Foundations of Digital Games*, 2015.

[Osborn *et al.*, 2017] Joseph C. Osborn, Noah Wardrip-Fruin, and Michael Mateas. Refining operational logics. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 2017.

[Smith *et al.*, 2010] Adam M. Smith, Mark J. Nelson, and Michael Mateas. Ludocore: A logical game engine for modeling videogames. In *Proceedings of IEEE Conference on Computational Intelligence and Games*, 2010.

[Summerville *et al.*, 2017] Adam Summerville, Morteza Behrooz, Michael Mateas, and Arnav Jhala. What does that ?-block do? learning latent causal affordances from mario play traces. *Proceedings of the first What's Next for AI in Games Workshop at AAAI 2017*, 2017.

[1] A partial catalog is available at `http://operational-logics.soe.ucsc.edu/`