Anthology: A Social Simulation Framework

Sasha Azad, Jennifer Wellnitz, Luis Garcia, Chris Martens

Department of Computer Science, North Carolina State University
{crmartens, sasha.azad, mawelli, lgarcia7} @ncsu.edu

Abstract

Social simulation research seeks to understand the dynamics of complex human behavior by simulating populations of individual decision-makers as multi-agent systems. However, prior work in games and entertainment fail to account for interactions between social behavior, geography, and relationships in a manner that allows researchers to easily reuse their frameworks and model social characters. We present Anthology, an extensible software framework for modeling human social systems, within the context of an ongoing research agenda to integrate AI techniques from social simulation games and computational social science to enable researchers to model and reason about the complex dynamics of human social behavior. Anthology comprises a motive-based agent decision making algorithm; a knowledge representation system for relationships; a flexible specification language for precondition-effect-style actions; a user interface to inspect and interact with the simulation as it runs in real-time; and an extensive user documentation and reference manual. We describe our participatory research design process used for the developing Anthology, the state of the current system, it's limitations and our future development directions.

Introduction

Computers are increasingly being used to simulate and analyze complex social phenomena to provide models that can help predict and explain human behavior. Simulating human societies allows more rapid and less disruptive social experimentation: one can change the parameters of a simulation and observe its effects without requiring decades-long longitudinal studies or causing any harm.

However, social systems are complex and contextual in ways that are not captured by current techniques. For instance, most social network simulations (McCoy et al. 2012) do not account for agent decision-making outside of the social network, including day-to-day activities such as going to work, discussing one's life with family members and friends, and leisure time activities, all of which can influence human social behavior (Azad and Martens 2019). Conversely, simulations of strictly geographical phenomena (such as transportation traffic or land-use change) do not account for the social relationships among humans, including family, work, school, and community relationships, that are inextricably tied to how people decide where to live and how to transport themselves between locations.

This paper presents a system design and ongoing research agenda to integrate social AI techniques from social simulation games (McCoy et al. 2012; Samuel et al. 2015; Evans and Short 2014) into an extensible framework for social system modeling that incorporates both geographical and relational factors. Our system, Anthology [1], comprises a motive-based agent decision-making algorithm; a knowledge representation system for relationships between agents and other world entities; a flexible specification language for precondition-effect-style actions that can both depend on and modify said motives and relationships; and a user interface to inspect and interact with the simulation as it runs in real-time.

Our goal is to provide a usable, open-source implementation, alongside clear documentation, examples, and instructional materials for running one's simulation, to enable better reproducibility and reuse, obviating the need for other research groups to reinvent the wheel. For this reason, a core contribution of this work is the presence of a thoughtfully-designed specification format for configuring and extending Anthology's basic functions. Users provide a world specification file in a standard format (Javascript Object Notation, or JSON) that defines agents, locations, and actions (akin to a PDDL file for planning systems). Our base simulation includes a set of motivations or needs that drive the character's decision-making algorithm. However, these can also be modified or added to if desired.

This paper describes our participatory research design process for designing Anthology's features, the state of the system, the results of an expert evaluation of Anthology's usability and usefulness for social simulation in practice, and the future directions for this project. In the long term, we aim for this project to enable reuse and reproducibility for social simulation research projects within and outside of our group, and to allow social simulation researchers to model and reason about the complex dynamics of human social behavior.

[1] Code repository, API documentation, and examples can be found: https://github.com/SashaZd/Anthology-Social-Simulation-Framework.git

Related Work

Anthology is inspired by several previous projects that aim to simulate social behavior in virtual agents and non-player characters.

Prior work includes *Comme il Faut* (CiF) (McCoy et al. 2010) and its successor project *Ensemble* (Samuel et al. 2015). CiF boasts the first implementation of a "social physics engine," simulating narrative scenarios using social rules. Anthology incorporates similar rules in the form of requirements for actions which describe social preconditions that must be met for the actions to be available to the agents. However, CiF was primarily employed for a single game development project, Prom Week (McCoy et al. 2012), has no formal definition outside of its implementation, and lacks comprehensive documentation for use in external projects, creating barriers to reproducibility and reuse, whereas those features are key priorities for Anthology.

The interactive narrative authoring tool *Versu* (Evans and Short 2014) has agents that use utility-based decision making, considering both social norms and their own goals. In Anthology, agents also use utility-based decision making, seeking the actions that maximize their motives. Unlike Versu, which does not have a built-in locative component, Anthology agents also consider geographical context, constraining actions to specific locations, considering agents present, and travel time when computing utility.

More recently, attention has turned towards the need for authoring tools and languages, such as *Kismet* (Summerville and Samuel 2020), for social simulation. Kismet is billed as a lightweight social simulation specification language for facilitating the creation of small-scale scenarios, such as tabletop role-playing games. Kismet was designed to be accessible to lay people (e.g., non-experts in social simulation), yet expressive enough to cover a large range of possible scenarios. Likewise, Anthology users can edit a single JSON file, which can also be written in any text editor. Anthology's goals appear to diverge from Kismet's mostly in audience and intended use: we want to offer a flexible range of utility-calculating algorithms for agent decision-making (whereas Kismet currently supports only a simple "proclivity" model that determines the locations that agents travel to) and support realistic simulations of human populations at the scale of neighborhoods or cities. However, since both projects are at an early stage, there may still be fruitful paths to explore to integrate or better differentiate these efforts.

Research Goals

Anthology is intended to enable researchers, game developers, and social scientists to model and reason about virtual agent behaviors. Our primary design goals are usability and expressivity, defined as follows.

Usability: Users should be able to create a working simulation, from scratch, with 2-5 agents and 5-10 actions. It should take less than an hour, and they will not require supervised training from the research team nor need to read any source code.

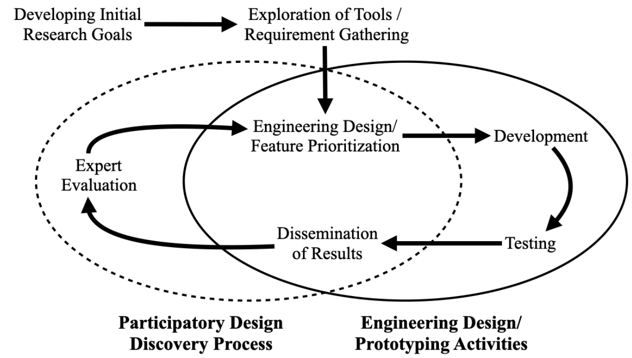

Figure 1: Participatory Design Methodology Overview

Expressivity: Users should be able to conceptualize multiple different use cases for the tool (once they understand its scope) across any genre of simulation, and successfully implement these scenarios.

These goals are not new: agent modeling systems, language specifications, and frameworks for social simulation have been generated in previous work (McCoy et al. 2010; Evans and Short 2014; Summerville and Samuel 2020). However, in practice, these systems do not see use outside of their original research team. A recent survey showed that existing social simulation systems tend to invent new project-specific terminology and do not build on existing conceptual frameworks for social systems, despite implementing very similar sets of social concepts (Azad and Martens 2021). This disparity among approaches impairs the reuse of these systems, with users opting to reinvent the wheel, building their simulations from scratch to test a single decision-making algorithm (Marsella, Pynadath, and Read 2004; Azad and Martens 2018). Two key elements of our approach support these goals: *participatory design research methodology* and our *motive-based decision making* approach. These have been detailed below.

Participatory Design Research Methodology

To ensure that Anthology meets the needs of our intended user base, we adopted a participatory design research methodology and development process, involving prospective users in a co-design and co-production process (Spinuzzi 2005; P. Carvalho et al. 2021; Mirel 1998). That is, rather than conduct human-subjects studies to evaluate the framework only at the end of development, we involve users in its design from its inception, allowing their goals and needs to shape Anthology's architecture and feature set.

Participatory design is iterative: we alternate phases of development and (formative) evaluation with prospective users. For each of our participatory phases, we recruited and interviewed experts familiar with social simulation. We performed an expert evaluation on testable versions of our prototypes after each development phase. We observed how experts used Anthology intending to understand their existing thought processes and workflows, as well as their *knowledge* *by doing*, the tacit ways in which they demonstrate knowledge by performing activities (Spinuzzi 2005) to meet their narrative and simulation goals. We noted when and why Anthology frustrated them or helped them accomplish goals. We used this feedback to iterate on each prototype and build the next one, which would again be followed by user feedback, for the entire lifetime of the project. This approach is illustrated visually in Figure 1. This process is still ongoing, but we have reached a stable enough point with our prototype that we believe it will serve the broader research community to share it at this stage.

We began by developing initial research goals, identifying the user base, and defining usability and flexibility goals for Anthology (as stated above). Next, we performed a literature review and gathered requirements based on a recent survey of social simulation and agent-based modeling tools (Azad and Martens 2021) During this process, we consulted with a paid expert on agent-based modeling and computational social science who is familiar with building frameworks for both researchers and industry practitioners who want to use social simulation tools to analyze real-world human behavior. These steps allowed us to define the minimal viable product (MVP) features of Anthology required to be developed before any evaluation or co-design phases could begin. We generated and prioritized a list of features to serve as a roadmap for a sequence of Anthology prototypes. Once the MVP was ready, we created comprehensive documentation to support self-directed learning and the use of Anthology.

In our next iteration, we recruited two additional experts in games and social simulation research. They were asked to perform a series of tasks with the tool, aimed at evaluating the usability and flexibility of Anthology. We took into account their feedback, and *iterated on the design* of the tool following the software engineering cycle (i.e. design, development, and testing) followed by an updating of our documentation for a detailed *dissemination of results*. Once this process was complete, we iterated a 3rd time, recruiting two new experts (from social simulation research and an industry practitioner) and repeating the process above.

Approach

We believe our decision to use Motive-Based Decision Making as our approach supports our design goals of usability and expressivity. On the AI architecture side, we decided to base Anthology's approach to agent decision-making on one of the simplest and most widely-familiar ideas in-game AI: motive-based utility. This idea is familiar to many people because of the popularity of the Sims franchise (Maxis 2003) where each action in the world (e.g. watching television or cooking a meal) can fill and/or deplete any subset of these motives. Our terminology and formalization of this idea comes from Millington and Funge's textbook *Artificial Intelligence for Games*, 2nd. ed. (Millington and Funge 2018).

As Millington and Funge note, motive-based decision-making algorithms fall under a more general class of approaches to goal-oriented agent behavior (Millington and Funge 2018). Starting with a naive approach that evaluates an action based only on its immediate net effects, one can

Listing 1: `Default Decision-Making Algorithm`

```
function get_next_action(agent):
  best_act = wait;
  best_u = 0;
  for each action of world:
    find locations where action is possible
    compute travel times for each location
    get nearest such location l
    u = utility(agent, action)
    u /= action.time + travel_time(l)
    if (u > best_u || (u == best_u && withProb(0.5))):
      best_act = action
      best_u = u
    agent.action_queue.push(best_act)
```

gradually dial up complexity to handle more complex situations, up to and including general-purpose planning algorithms that generate multi-step action sequences. This unification of approaches gives us a nice way to structure the system modularly, open to any decision-making algorithm that calculates numeric utility for an agent and an action. In our initial prototype, we use a simple utility-sum-based to select actions as can be seen in Figure 1. For each action that is possible for an agent, we consider the sum of its effects on all motives and divide it by a metric to account for the travel time required for that action.

Anthology: System Description

At a high level, Anthology follows a Model-View-Controller software design pattern and architecture. Anthology is currently able to simulate virtual worlds that can be described using *models* of agents, actions, and locations. In Figure 2 we depict how the interactions between the agent, action, and location *controllers* are mediated by a central execution engine. The models are populated by the user by the input of a single JSON file. The input from the JSON file sets the initial state of the system and begins the simulation. The user can *view* the output of the system on our web interface, which depicts a world map, the locations, agents (with their current motivations), and any upcoming events in the system. The javascript console can be pulled up for a detailed log of historical events in the simulation.

We adopt a running example we have devised called the College Roommates Scenario to explain each component of the system. In this scenario, *Norma*, a math student, and *Quentin*, a physics student, live in the same dorm. The campus on which they reside includes the dormitory, lecture halls, a computer lab, a dining hall, and a greenway (outdoor path). Our modeling goal is to simulate and observe how college students might balance different life activities, such as attending class, doing homework, eating, relaxing, and socializing, by providing simple models of how these actions affect them.

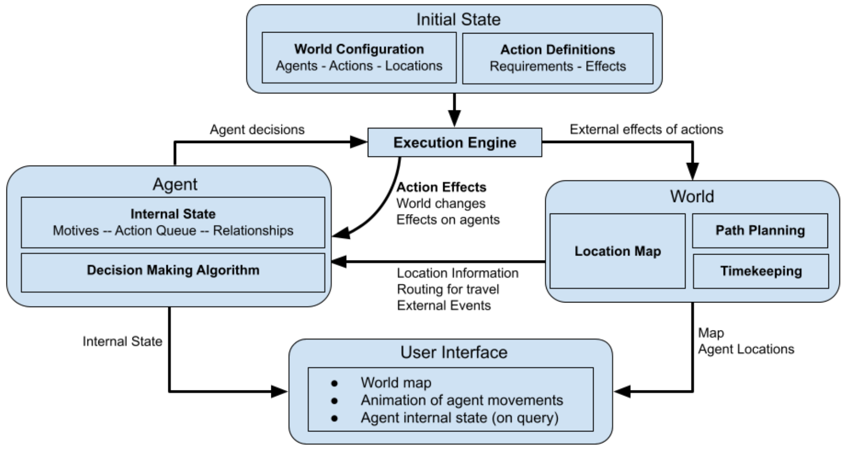

Figure 2: Anthology's system architecture.

Listing 2: Agent Example: `Norma, a math student`

```
{   name: "Norma",
      motive: {
          accomplishment: 2,
          social: 2,
          physical: 4,
          emotional: 3,
          financial: 5 },
    relationships: [
        { type: "friend",
          with: "Quentin",
          valence: 3 },
        { type: "student-of",
          with: "MathProf",
          valence:1 }],
    currentLocation: {
        xPos: 0,
        yPos: 0 },
    occupiedCounter: 0,
    currentAction: "wait_action",
    destination: null
}
```

Listing 3: Location Example: `Dining Hall Location`

```
{
  name: "Dining Hall",
  xPos: 5, yPos: 5,
  tags: ["food"]
}
```

Agents

Agents represent the virtual characters in Anthology simulations. The Agent type includes a unique name, a set of (mutable) motive values, and (mutable) relationships with other agents.

Motives represent the needs of the agents. By default, we supply 5 motive types in the system: Physical, Emotional, Social, Financial, and Accomplishment, based loosely on Maslow's classic theory of human motivation (Maslow 1943). We chose these motives to correspond to real-world human motives as follows:

- *Physical*: The need for an agent to maintain their body through actions such as eating, sleeping, and exercising.
- *Emotional*: The need for leisure time, play, and mental rest, addressed by actions relating to rest or recreation.
- *Social*: The need to interact positively with other agents, met by actions that involve multiple agents.
- *Financial*: The need for financial stability; addressed by working.
- *Accomplishment*: The need to achieve something, addressed by having hobbies or earning rewards.

Each motive is represented by a number on a scale from 1 to 5, where lower numbers indicate a lower level of satisfaction. Motives change over time, both in response to the actions an agent takes and as a product of motive decay on a fixed interval.

Agents move the simulation forward by undertaking actions. Each turn, they either make progress towards the next action in their queue, or they choose a new action to add to their queue. See Figure 2 for an example of how we could model our college student Norma.

Locations

Locations represent geographic points of interest on the Anthology World Map. Anthology uses the locations listed in the JSON file to setup the world, check whether actions can be performed at a specific location, and track agents moving around the map.

Locations can be added by adding geographic coordinates, an optional name, and an optional list of associated tags to the JSON. The list of tags represents what actions are possible at a given location. These tags are fully user-defined and are compared against location requirements when agents consider which actions to take. For instance, Norma would

Listing 4: Action example: `Attending Class`

```
{   name: "attend_class",
    requirements: [
      { reqType: "location",
        hasAllOf:["classroom"]},
      { reqType: "people",
        relationshipsPresent: ["student-of"],
      minNumPeople: 2 }],
    effects: [
      { motive: "accomplishment",
        delta: 1 }],
    time_min: 75
}
```

evaluate the Dining Hall location (with the "food" tag) as represented in Figure 3 and depicted on the top right corner of the map in Figure 3 as a candidate location for dinner.

Actions

In Anthology, actions are the main mechanism by which the agents advance the simulation and change their motive values. While there can be different types of actions within the simulation, all actions include a unique name, an associated time that it occupies, and a set of requirements.

All actions have an associated time which represents how many turns the action takes to be completed (if not interrupted). Agents only choose a new action if they are not already executing an action. An agent may hold a list of several actions at any given time forming a queue of actions to take where the front of the queue is the action currently being performed. This can be seen in Figure 3 where Norma's action queue includes traveling to a restaurant to join a friend who is currently eating their lunch there. In general, there are three types of requirements actions can have:

- *People*: This requirement enforces which agents are present or absent for an action to be performed. Authors can specify the minimum and the maximum number of agents attending, list out specific agents that must be present or absent, or require the presence or absence of an agent by specifying a relationship type.
- *Locations*: Location requirements determine the eligibility of a location for an action to be performed. Authors can specify a set of tags that must all be present in the location's tag list, a set that one or more of must be present, and a set that all of must be absent from the tag list.
- *Motives*: Motive requirements allow authors to compare agents' motive values to threshold values to determine whether or not an agent is capable of taking the action.

The action, `attend_class`, shown in Figure 4 depicts an agent, Norma, attending a lecture. In this example, its location requirement states that the agent must be in a location tagged `classroom` (remote learning is frowned upon in our virtual university), and the people requirement states that Norma must have a `student-of` relationship with an attendee, i.e. her professor must be present. This action increases the `accomplishment` motive by 1 and will occupy Norma for 75 minutes (or iterations) of simulation time. There are two types of actions in the Anthology system: *primary actions*, and *scheduled actions*.

Primary Actions Primary actions are atomic actions that are executed directly by a single agent. Specifically, their effects only change the motives of the agent that performs them. In Anthology, action effects are a tuple of a motive type and an integer value. When a primary action is executed, the specified motive of the agent who executed it will be adjusted by the provided integer value. Figure 4 shows an example of a primary action, `attend_class` that increases the `accomplishment` motive for any student executing it.

Scheduled Actions Scheduled actions, by contrast, are social actions involving plans undertaken by multiple agents (so they need to be "scheduled" for a time when both agents are unoccupied), and they consist of multiple primary actions. Scheduled actions affect not only the agent *instigating* the action but also *targets* other agents around them. In its current version, Anthology deems any agent in the same location as a target of a scheduled action. In lieu of effects, schedule actions have instigator and target actions which are added to the action queue for the participating agents. For instance, in a scheduled `hug_romantically` action an instigator may attempt to hug a target agent. In this case, the scheduler adds an instigating action `attempt_hug` to the instigator, and `accept_hug` or `reject_hug` onto the action queue of the target agents.

Finally, an action may be `hidden`, to represent actions that can only ever be added to the queue of an agent in response to a scheduled action. For instance, an agent may never select the action `reject_hug` without another agent attempting to hug them.

Execution Engine

The execution engine has two primary responsibilities: periodically decaying agent motives and triggering each agent's turn. Every time step, the execution engine calls for each agent to either make progress on an action or choose a new action to perform every turn.

When an agent is called upon to take their turn, if they have no actions remaining in their queue, they must select the next action to take. Agents are incentivized to select actions that improve their motives, so long as the requirements of an action can be fulfilled. The overall gain in motives is calculated and weighted to take into account the time it would take to complete the action and optionally any travel time required. This computed value is the utility of the action. Action utilities are then compared and the highest utility action is chosen by the agent, deferring to random chance when deciding between actions with equivalent utilities as can be seen in Fig. 1.

User Interface

Currently, the user interface for the simulation is a configurable simple grid-based map (see Figure 3). Each cell in the grid can contain any number of locations and agents.
Agents are denoted by capital letters representing the first letter of their name, and locations are denoted by lower case

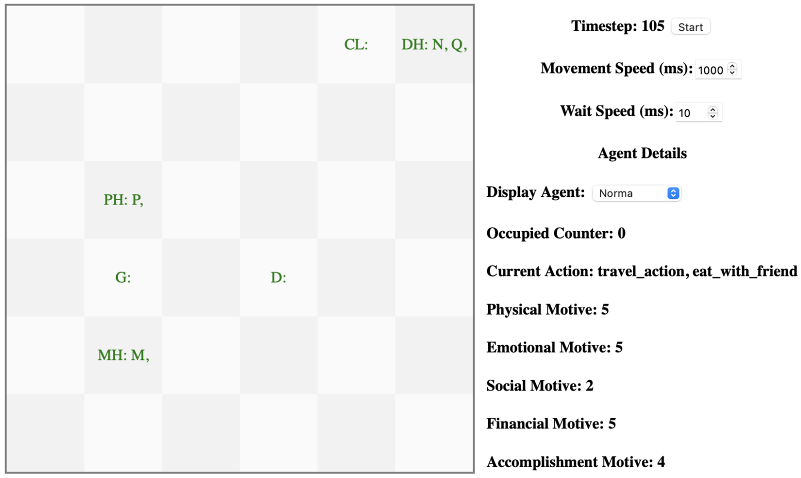

Figure 3: User Interface depicting our University example

letters of the first letter of their names. There are two different running speeds for the simulation: one for when all agents are staying in the same location, presumably working on completing an action, and another for when any agent is moving about the map. Both these speeds can be modified by users in real-time on the side panel of the user interface.

Further, users can select any agent to view real-time information about them. The side panel will display their current motive values, their action queue, and their occupied counter. This can be used to keep track of how a particular agent addresses their motives throughout the course of the simulation. To get a more fine-grain view of all agents, we also log a full action trace for the simulation in the developer console of the browser.

API Documentation

The Anthology system comes with a complete set of user documentation (see Figure 4). This documentation is auto-generated from the typescript documentation of our code base and is separated into modules for convenience.

Our documentation is detailed. We provide both a general description of every type and function, as well as what it's used for, and a description of each input and output the same.

Scalability

To measure the computational scalability of the system, data was collected on the average time of a single turn (one tick of the simulation where all agents advance their current actions or choose a new action), the total time of a complete simulation, and the total number of turns in a complete simulation. Each of these metrics was measured as the total number of agents in the simulation was varied, and each configuration of the simulation (i.e. each trial number of agents) was tested five times and the results aggregated into a single data point for each metric. The simulation was tested with configurations of the agents ranging from 2 to 10000.

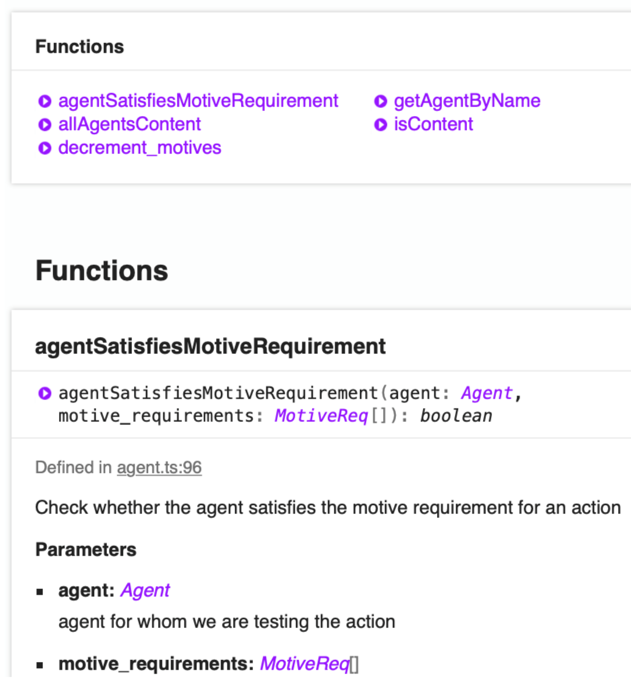

Figure 4: Documentation for Anthology's Agent Module

A graph detailing the average total time scaling for the simulation can be seen in in Fig. 5, with the average turn time following a largely similar pattern. For both metrics, notable slowdowns did not occur until 5000 agents were run simultaneously in the simulation. While turn times were all under 1ms previously, and total times ranged from 22000ms to 32000ms, at the 5000 agent mark, turn times increased to several ms and total times more than doubled. However, the number of turns remained constant regardless of the number of agents simulated.

Discussion and Future Work

With our running example, in Fig. 6, we demonstrate the varied behaviors that can emerge out of Anthology by only changing the input JSON file. The ease with which the simulation can be refined, as tested during our formative evaluations in our participatory design phase, shows that Anthology's design is conducive to rapid prototyping and iteration. However, JSON files have syntax requirements that can prove to be burdensome in larger projects. These limitations may be rectified by the inclusion of a domain-specific language for Anthology akin to Kismet (Summerville and Samuel 2020).

Using the Participatory Research Design methods described above allowed us to iteratively design and develop the emerging design. Expert user feedback phases helped us to envision and shape the system alongside participants who will eventually form a part of our user base for the tool (Spinuzzi 2005; Mirel 1998). We believe that this methodology has contributed to the project making sustainable progress and that in the long term it will improve the practicability and long-term reach of the project. We will continue to iterate on the Anthology system following the same methodology and protocols. Our roadmap for future prototypes of Anthology includes, but is not limited to:

Agent-specific motives: Currently all agents have the same motives that decay at a set periodicity and increase by a set value based on the actions they are undertaking. A future version of Anthology would support agents with distinct or weighted motives. Additionally, we would like to change the factor by which the agent's motives are modified. This would allow us to model, for instance, an introverted agent that is more socially fulfilled (i.e. receives greater motive increments) by performing easier social actions (i.e. attending lab meetings), and who is easily overwhelmed by larger social actions (e.g. a crowded frat party).

Scheduled joint actions and interrupts: Currently, our implementation of joint actions only supports immediate interactions between agents. A more robust implementation would allow agents to schedule joint actions for a future time. Additionally, in the current version, agents receive the entire effects (i.e. motive changes) of their actions at the

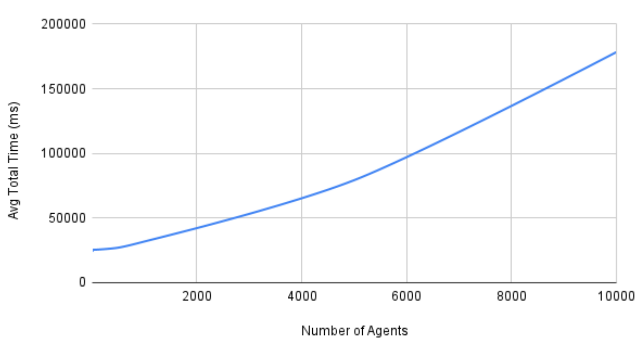

Figure 5: Chart depicting the average run-time when scaling from 2 to 10000 Agents

```
time: 0 | Norma: Started travel_action; Destination: Greenway
time: 0 | Quentin: Started travel_action; Destination: Dorm
time: 4 | Norma: Finished travel_action
time: 4 | Norma: Started go_for_walk
time: 6 | Quentin: Finished travel_action
time: 6 | Quentin: Started do_homework
time: 35 | Norma: Finished go_for_walk
time: 67 | Quentin: Finished do_homework
time: 67 | Quentin: Started travel_action; Destination: Dining Hall
time: 73 | Quentin: Finished travel_action
time: 73 | Quentin: Started eat_alone
time: 99 | Norma: Started travel_action; Destination: Dining Hall
time: 105 | Norma: Finished travel_action
time: 105 | Norma: Started eat_with_friend
time: 134 | Quentin: Finished eat_alone
time: 134 | Quentin: Started eat_with_friend
```

Figure 6: Simulation output from our university example

end of the action duration period. This combination of features has led to hilarious (but unfortunate) narrative scenarios where an agent about to complete their 8-hr workday was interrupted by another agent that yelled at them, and consequently wasn't paid (i.e. received no financial or accomplishment motive gain) for the entire day.

Conclusion

In conclusion, we have presented Anthology's system design and its role in our ongoing research agenda to integrate social AI techniques from digital games into a usable, expressive framework for modeling and understanding human behavior. We defined our design goals from a human-centered software development perspective, explained Anthology's technical underpinnings in an implementation-independent way, and demonstrated a modeling interaction in which a scenario is constructed, run, and iterated upon to reveal simulation insights and prompt social science research questions. In the long term, we expect this project to enable reuse and reproducibility for social simulation research projects within and outside of our group, and to allow social simulation researchers to model and reason about the complex dynamics of human social behavior.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research, award number FA9550-20-1-0355.

References

Azad, S.; and Martens, C. 2018. Addressing the Elephant in the Room: Opinionated Virtual Characters. In *AIIDE Workshops*.

Azad, S.; and Martens, C. 2019. Lyra: Simulating believable opinionated virtual characters. In *Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment*, volume 15, 108–115.

Azad, S.; and Martens, C. 2021. Little Computer People: A Survey and Taxonomy of Simulated Models of Social Interaction. *Proceedings of the ACM on Human-Computer Interaction*, 5(CHI PLAY): 1–30.

Evans, R.; and Short, E. 2014. Versu - A simulationist storytelling system. *IEEE Transactions on Computational Intelligence and AI in Games*, 6: 113–130.

Marsella, S. C.; Pynadath, D. V.; and Read, S. J. 2004. PsychSim: Agent-based modeling of social interactions and influence. In *Proceedings of the international conference on cognitive modeling*, volume 36, 243–248.

Maslow, A. H. 1943. A theory of human motivation. *Psychological review*, 50(4): 370.

Maxis, E. 2003. The Sims.

McCoy, J.; Treanor, M.; Samuel, B.; Reed, A. A.; Wardrip-Fruin, N.; and Mateas, M. 2012. Prom week. In *Proceedings of the International Conference on the Foundations of Digital Games*, 235–237.

McCoy, J.; Treanor, M.; Samuel, B.; Tearse, B.; Mateas, M.; and Wardrip-Fruin, N. 2010. Authoring game-based interactive narrative using social games and comme il faut. In *Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate*, volume 50. Citeseer.

Millington, I.; and Funge, J. 2018. *Artificial intelligence for games*. CRC Press.

Mirel, B. 1998. “Applied constructivism” for user documentation: Alternatives to conventional task orientation. *Journal of business and technical communication*, 12(1): 7–49.

P. Carvalho, L.; Jackson, D.; Guerreiro, T.; Guan, Y.; and Montague, K. 2021. Participatory Action Research and Open Source Hardware Appropriation for Large Scale In-The-Wild Studies. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, 511–513.

Samuel, B.; Reed, A. A.; Maddaloni, P.; Mateas, M.; and Wardrip-Fruin, N. 2015. The ensemble engine: Next-generation social physics. In *Proceedings of the Tenth International Conference on the Foundations of Digital Games (FDG 2015)*, 22–25.

Spinuzzi, C. 2005. The methodology of participatory design. *Technical communication*, 52(2): 163–174.

Summerville, A.; and Samuel, B. 2020. Kismet: a small social simulation language. In *Summerville, A., Samuel, B.(2020, September). Kismet: a Small Social Simulation Language. In: 2020 International Conference on Computational Creativity (ICCC).(Casual Creator Workshop). ACC*.