Encoding Socio-Historical Exegesis as Social Physics Predicates

Daniel DeKerlegand
University of New Orleans
dldekerl@uno.edu

Ben Samuel
University of New Orleans
bsamuel@cs.uno.edu

Jeffrey Leichman
University of New Orleans
jleichman@lsu.edu

ABSTRACT

This paper concerns developments in social physics rule and schema authoring made as part of the ongoing VESPACE project, a multinational research effort in developing a social physics driven virtual reality experience set in eighteenth-century France. Central to the theoretical application of social physics in this socio-historical context is the concept of *citation as AI*– that is, the historically-grounded reconstruction of human behavior represented by cited schemata, rules, and actions. In order to facilitate the authoring of domain-specific rules for the *Ensemble* social physics engine by history and cultural studies experts, we endeavored to develop a new collaborative authoring tool, which we tested in a series of workshops. The results of these workshops revealed various domain needs, as well as the shortcomings of representing data structures using one-to-one correspondences in UX/UI design.

KEYWORDS

artificial intelligence, narrative, socio-historical analysis, human-computer interaction, UX/UI

ACM Reference Format:
Daniel DeKerlegand, Ben Samuel, and Jeffrey Leichman. 2020. Encoding Socio-Historical Exegesis as Social Physics Predicates. In *International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3402942.3409608

1 INTRODUCTION

Games bringing social interaction to the forefront, such as *Façade* and *Prom Week*, have illustrated the expressive power of AI driven interactive narratives; these experiences are capable of producing exponentially more playthroughs and social interactions than other games having narrative choices but whose narrative threads are predetermined and handwritten. Games such as these which allow players the autonomy to construct narratives of their own have been described as *emergent.* Characters in emergent games have needs and desires, which can lead to conflicts and other rich narrative experiences. Player choices frequently result in emotional responses from NPCs, leading to further consequences [10]. Several blockbuster games, such as *The Sims* and its sequels, have already applied core principles of emergent narratives [1]. Furthermore, research has illustrated the possibility of using emergent narrative systems such as social physics engines to enhance NPC interaction in complex and highly successful AAA games like *Skyrim* [8][9].

Prom Week is notable in particular for its deep, procedurally-driven social play, where each player choice affects the social space, and actions can have rippling effects, determining the range of future actions [16]. The space of possible play traces in *Prom Week* is rich enough to allow for a completely new, unique play trace per player [21]. The wide possibility space of character interaction is made possible through the use of a *social physics* engine, the game itself being treated as a social physics puzzle with no predetermined narrative and having a multitude of possible outcomes [13]. The range of possibility of these actions and states is defined by a corpus of handwritten schemata, rules, actions, character traits, and character histories.

Until now, game developers applying social physics have focused on the creation of rich, dynamic fictional experiences, without significant emphasis placed on the *historical* verisimilitude of the experience. For *Prom Week*, the authoring of social schemata and rules was fairly informal, and limited particular importance was placed on grounding character behaviors in historical, sociological, or psychological sources beyond the sociological metaphors embedded in the social physics engine [20]. The game takes place at a high school in the present-day United States, so the setting is familiar both to the developers and a significant portion of the intended players, especially those raised in the United States or familiar with its media. While the authoring process of the game required research in the form of analyzing pop culture artifacts such as films and television shows involving high school students, the source materials only formed a small corpus: the two films *Twilight* and *Mean Girls*, along with a handful of other sources [17]. Furthermore, these sources did not represent the origin of all composed schemata and rules; in addition, the authors leveraged anecdotal experience and general knowledge.

The *Ensemble* engine is based on the earlier social physics engine *CiF* or *Comme il Faut*, which was used in *Prom Week.* Once it has been initialized by user-provided data, *Ensemble* can compute social interactions, possibilities, and histories. The authoring process consists of generating four core sets of user-provided data– schema, rules, actions, and characters– which are built upon first-order predicates. One of the important improvements of *Ensemble* over *CiF* is a further abstraction and flexibility of data representation, which makes it possible to construct social worlds both big and small, ranging from the hyperreal to the highly imaginative. By providing the necessary data, authors can tune the *Ensemble* engine to whichever historical, sociological, and psychological concepts and rules are pertinent to their simulation. Our current research suggests that given the right tools and basic training, authors can fairly quickly construct social worlds tailored to their own games.

The VESPACE project, which seeks to replicate an eighteenth-century French theatre in virtual reality, will use *Ensemble* in order

FDG '20, September 15–18, 2020, Bugibba, Malta

ACM ISBN 978-1-4503-8807-8/20/09…$15.00
https://doi.org/10.1145/3402942.3409608

to recreate believable social interactions between player and NPCs. Due to its historical and pedagogical context, the project necessitates a close alignment of research and social physics authoring. Though the physical setting of the experience is small, the historical research necessary in order to create consistent, historically-grounded character interactions and social mores is quite vast.

Our goal in this particular project was to train a number of professional literary historians in social physics authoring, and then to collaborate within one week's time to construct a playable experience entirely sourced from scholarly material. In order to accomplish this, we needed a tool through which non-engineers could construct schemata, rules, actions, and characters. Thus, the tool needed to be user-friendly, abstracting away unnecessary jargon and facilitating the quick transformation of textual information into first-order logic. It also needed to be capable of recording and producing useful metadata including citation details, which would be used not only for citing data sources but also for altering and defining possible playthroughs and social experiences, a mechanism we call *citation as AI*.

2 RELATED WORK

Various authors have discussed the applicability of differing narrative types in games and in particular the use of emergent narratives [2][11]. One point of contention with emergent narrative games is the difficulty in achieving sufficiently detailed dialogue [5], and various dialogue authoring methodologies have been suggested [24][26]. Narrative visualization is posited as a useful technique for finding interactive narrative flaws such as dead ends [6].

Others have discussed the crucial role of authoring tools in the development of interactive narrative games [12], as well as methods for evaluating the expressive power or *authorial leverage* of these authoring tools [3]. Several meta-analyses and surveys of existing authoring tools have provided useful prospective taxonomies, as well as an overview of common issues [23][7]. Attempts have been made to produce a standard specification for interactive storytelling authoring tools, but these have not yet been widely adopted [25].

3 VESPACE

VESPACE is a multi-disciplinary, collaborative research effort between universities including Louisiana State University, Université de Nantes, and the University of New Orleans. The end goal of the project is the creation of a VR experience transporting users to the eighteenth-century Saint-Germain Fair theatre in Paris. The VESPACE project will be the first application of social physics in virtual reality. Ultimately, the project aims for the restitution of this long-gone social space, not only in terms of its physical atmosphere but also in its cultural and social context [4]. Sharing goals with both works of museography and with *serious games*, the VESPACE project is developing methods of engendering learning and study through immersion, as well as a dynamic setting for testing and challenging socio-historical hypotheses.

The physical space of the game's theatre, which is in particular a marionette theatre, has been exactingly recreated from a miniature image on a snuffbox. All elements of the space have been tailored to reflect the ambiance as it would have been, such as lighting the space with the exact lumens produced by candlelight fixtures of the period.

Despite the level of detail in the setting, the experience of VESPACE is not intended to be solely a passive one of observation. The Saint-Germain Fair theatre, being controversial during its time and frequently at odds with both wealthier theatre companies and the French government, represents a particularly dynamic space for social physics, as the rules of engagement would have been strictly governed by the social mores of a socially-stratified Enlightenment France [19]. In order to leverage these complexities, the player will perform the role of a spy seeking to gain insider information of the theatre, which can only be accomplished by interacting with a variety of different NPCs, each deeply modeled in terms of class, social status, gender, and personality.

At the root of the VESPACE project is a set of questions concerning epistemology and the reconstruction of history in illusionistic computer-mediated environments while remaining transparent about sources and the limits of historical certainty. The project also broaches the question of what role the use of technologies such as virtual reality or video game engineering will play in academic scholarship, specifically in the humanities, and how these technologies can be used as pedagogical tools.

Due to the necessary research, crafting schemata and rules for the VESPACE project is the purview of French historical, literary, and theatre scholars, rather than AI engineers and developers. While software engineers may be familiar with themes of teenage romance, they are likely less familiar with other domain-specific social knowledge, such as the mores of French aristocratic society during the Enlightenment. One important aspect of the project is in training scholars, who may be new to designing computer-assisted social experiences, in composing the predicates that make social interactions possible. These authors must work to synthesize a large corpus of historical texts and citations– ranging from period fiction to modern historical analyses– into social rules which are computed by the social physics engine.

The work of social physics authoring in VESPACE is two-fold. First, scholars must perform a *socio-historical exegesis* of their source material, that is, interpreting texts in order to derive evidence and meaning such as behaviors, beliefs, and desires as revealed by individuals in these sources or by the original authors themselves. Secondly, social physics authors must take these parcels of knowledge derived through analysis and encode them in the form of first-order predicates comprehensible to an AI system. The intended schema and rule authors of the VESPACE project– experts in literary history and other humanities fields– are perhaps less likely than engineers to be familiar with concepts such as first-order predicate logic, Boolean values, or sets. Thus, our goal in this project was to develop an authoring tool playing to the strengths and catering to the needs of non-engineer authors.

Traditional literary-historical scholarship has developed protocols of textual analysis and exegesis that have proven for centuries to be effective methods for the creation and dissemination of knowledge. As we increasingly rely on interactive, computationally expressive forms of data representation– whether in web pages, phone apps, or VR games– we run the risk of losing much of this knowledge. Therefore, we believe that it is important to begin the work of bridging these two forms of knowledge representation, making it possible to extend this scholarship to the medium of AI-driven games. Furthermore, as social physics continues to find new fields of application, it stands to reason that we will need to continue to develop and refine the authoring process, so that different forms of data and metadata can be recorded and reflected in these experiences. With the right tools, we can aid scholars in transforming enormous, preexisting sources of knowledge into data capable of producing rich AI experiences.

4 DATA REPRESENTATION IN *ENSEMBLE*

In order to conceptualize how an authoring tool might facilitate the transformation of historical data into social physics predicates, it is first important to understand the general structure of data representation in *Ensemble*. At the top level, authoring social physics in *Ensemble* consists of four primary categories– schema, rules, actions, and characters– with schemata being the most abstract and defining the range of expression for the other three categories.

4.1 Schema

A schema in *Ensemble* is an abstract grouping of *categories* and *types* which define possible states within the system, making possible the predicate logic of larger components such as rules. The *Ensemble* schemata system is an improvement upon the previous *Comme il Faut* system, adding the flexibility of user-authored categories; previously, *CiF* provided users with pre-determined categories of state such as traits and statuses, whereas *Ensemble* gives authors the option of creating their own by specifying properties like directionality, value type, and duration [22]. However, as has been observed in recent workshops, the data structures used in schema authoring, whose properties are derived from concepts in computer science, are a new form of encoding for authors as yet unfamiliar with the creation of computer-assisted social experiences. These two main data structures of the schema, *categories* and *types*, are here described:

4.1.1 Category. A category is a structure of state representation within *Ensemble* that is defined by a set of properties, and each category contains a set of types. For instance, a category of "network" might represent interconnections between characters. This category could be visually represented as a directed graph, reflecting a weighted social network of individuals.

The value for the category could be numerically represented, such that each edge of the network is weighted. Importantly, the "network" category might be directed but not reciprocal (i.e. bidirected), so that each individual's network bond to another is distinct and not co-dependent. A minimum value, maximum value, and default value can be specified, though the typical range would be from 1 to 100, with a default value of 50.

In total, the following properties together comprise and define a particular category:

- **isBoolean** - category is Boolean as opposed to numeric or textual
- **directionType** - direction of edges (undirected, directed, or reciprocal)
- **defaultValue** - if numeric or textual, the default value
- **minValue/maxValue** - if numeric or textual, the minimum and maximum values
- **duration** - how many "steps" this category's state will last, these steps being a measurement of time within the *Ensemble* engine comprising one turn of play and predicate recalculation
- **actionable** - if the category can be part of character intent formation

4.1.2 Type. Types represent single instances within a category. In the case of the "network" category, a network type specifies a specific kind of social relationship, each based on a different social bond, such as friendship, family ties, or business relations. For example, "network" might include such types as "business," "artistic," "affinity," or "familial." Internally, these types are represented as an array of strings which are stored as part of the category object; however, in our requirements gathering for the new *Ensemble* authoring tool, we determined that for authors-in-training, it was easier to think in terms of types rather than categories. Therefore, we designed the schema authoring component of the tool such that users first create types and then either add these types to an existing category or build a new category up from the new type.

4.2 Rules

Rules in *Ensemble* govern both a) changes to the world state based on predicates which are calculated to be true at a given step in the social record and b) character desires and behaviors. The former are called trigger rules, while the latter are called volition rules. Both types of rules contain a right hand side and a left hand side, which are themselves composed of predicates. The structure of the left hand side is identical for both types of rules, being composed of any arbitrary number of precondition predicates which must evaluate to true in order for the right hand side to execute. However, though the right hand side of a rule always contains its effects, trigger rules and volition rules feature differently structured effect predicates.

4.2.1 Trigger Rules. Trigger rules are rules which may change the state of the world when the conditions of the rule are met. At each step in *Ensemble*'s step system, all trigger rules in the rule database are evaluated for their truthfulness. If the world state has changed– or remained– in such a way that the conditions for the trigger rule evaluate to true, the effects of that trigger rule will then become true as well.

Table 1: Trigger Rule - "Noble falls in love with beautiful charming charismatic woman"

Conditions	Effects
A has more than 75 attractiveness	B is gobsmacked
A is charming	B has exactly 100 affinity for A
A is beautiful	
B is male	
A is female	
B is noble	
B has more than 50 sensitiveness	

Table 1 shows an example trigger rule which was composed by an historical scholar during our week-long remote workshop and which cites the 1731 novel *Manon Lescaut* by Abbé Prévost. This rule, entitled "noble falls in love with beautiful charming charismatic woman," defines the manner by which a nobleman might become enamored of a beautiful bourgeois woman, capturing experiences indicative of class and gender norms in eighteenth-century France. The rule requires that a woman– character A– has an attribute of attractiveness with a value greater than 75, a trait of charming, and a trait of beautiful. It also requires that a man– character B– has an attribute of sensitiveness with a value greater than 50, as well as a trait of noble. If these are true for characters A and B, then the conditions of the rule will evaluate to true, and the effects will become true as well– that is, the man will gain the trait of gobsmacked, and he will gain a network affinity for A with a value of 100. Thus the rule is, more or less, the rule of "love at first sight."

4.2.2 Volition Rules. Rather than determining changes to the world state, volition rules govern character tendencies and desires. Like trigger rules, volition rules will be evaluated on the left hand side, so as to determine if the rule is applicable at a given step. If the left hand side evaluates to true, then the right hand side will execute, affecting characters' volitions, such as by incrementing or decrementing a character's volition to increase or decrease some value associated with that character. An activated volition rule thereby increases or decreases a character's likelihood to perform some action, depending upon the effects of that action. For this reason, unlike trigger rules, the right hand side of a volition rule must be composed of actionable predicates.

Table 2: Volition Rule - "Provincial bourgeois males married to noble women have less affinity with provincial bourgeois"

Conditions	Effects
A is bourgeois	A has +3 volition to decrease affinity for C
B is noble	A has +3 volition to decrease emulation for C
A is male	C has +3 volition to decrease emulation for A
B is female	
A is disdainful	
A is provincial	
A is married to B	
C is bourgeois	
C is provincial	
C has more than 65 affinity for A	

One example of an actionable predicate category could be a category entitled "network," which could be represented by a weighted, complete directed graph whose weights can reflect, for example, a character's affinity, curiosity, or animosity toward another character. As with other types in *Ensemble*, these are arbitrary and may be defined by schema authors.

Table II shows an example volition rule, another rule developed by an historical scholar during our week-long remote workshop. This rule "provincial bourgeois males married to noble women have less affinity with provincial bourgeois" similarly encodes behaviors indicative of class and gender norms in eighteenth-century France, but rather than encoding an event or experience that might *happen to* a character as in a trigger rule, this volition rule defines what might *be done by* a character. According to this rule, a disdainful bourgeois man who is married to a noble woman will possess a volition to decrease affinity for a bourgeois woman– thus, it manages to define, in predicate logic, class snobbery gained through marriage. One gameplay effect will be an increased likelihood that any character fitting description A will perform some action to decrease affinity for any character fitting description C.

4.3 Action and Characters

Actions in *Ensemble* begin with *intents*, which represent intentions to change the world state. Specific actions correspond to these intents; for example, if a character possesses an intent to "increase animosity" with another character, one available action might be "yell at." One enhancement of *Ensemble* over *CiF* is the ability for actions to point at further specific actions. In selecting a character action, if *Ensemble* determines that a user meets the preconditions for a more specific *instantiation* of that action, *Ensemble* can choose that more specific action. For example, the instantiation "yell at a close friend" would be different from "yell at an enemy." This feature of hierarchical actions in *Ensemble* makes possible a richer, more detailed and appropriate set of actions.

Closely related to actions are the game characters. These characters can be described using any number of predicates, which will be loaded at the start of the game. In this way, authors can build up a cast of characters with initial beliefs, relationships, and social histories.

5 CITATION AS AI

The flexibility of the *Ensemble* engine and the abstractions afforded to authors can be used to produce reconstitutions of different societies, cultures, and eras; however, it could similarly be used to produce different interpretations or social renderings of the *same* society, culture, and era. While the VESPACE project does not claim to produce an historical reconstruction to be considered "authentic" or an "exact" representation of an eighteenth-century French theatre, the project does seek to produce a set of reconstructions which are consistent and are supported by period source materials. This is the idea underlying the concept that we call *citation as AI*.

Humanities scholars typically aggregate data and facts at the level of the sentence or paragraph by way of citation, where a sentence or paragraph's validity may be bolstered by a footnote referencing a cited work. Analogously, in software engineering we typically organize data and facts at the level of the predicate or object, for example, a row in a database, or an object containing an array of objects. A notable improvement to the *Ensemble* authoring process introduced as part of the VESPACE project has been the inclusion of metadata, notably citation data, which may be attached to each item produced during the authoring process; in order to approximate the application of citation in the humanities, we determined that it should be possible to allow the inclusion of citations at the level of category/type, rule, action, and character. The addition

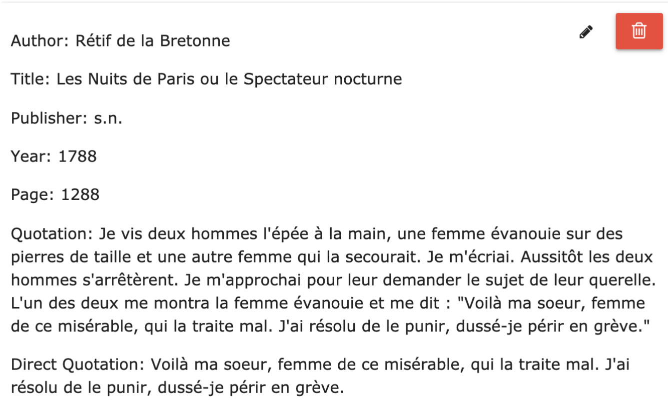

Figure 1: VESPACE Citation

of citation as part of the *Ensemble* authoring process has been a necessary step in moving from the production of anecdotal social experiences to experiences and reconstructions based in evidence and study.

As illustrated in Figure 1, VESPACE citations contain such expected fields as author, title, publisher, and year. In addition to these, they may also contain the fields *quotation* and *direct quotation*: These fields are important, because the former provides the necessary evidence to show that a category/type, rule, action, or character was authored based on historical scholarship. The latter, on the other hand, will be useful in the VESPACE game for lending verisimilitude in the form of actual, documented language from the time period.

In addition to evidence and verisimilitude, citations will provide a mechanism for manipulating the social experience itself. For instance, a player could elect to experience an historical setting, such as the theatre of VESPACE, through the lens of a particular author, historical worldview, or analytical framework; by increasing the weight of the effects bound to particular authors by citation, the social physics engine could thus present a virtual social experience "through the eyes" of those writers, whether they be from the historical era or from the present day. This foregrounding of perspective is an important part of the VESPACE pedagogical model, but it could make for interesting narrative gameplay in other contexts as well, as it provides a model by which a social state can be shifted or morphed according to a perception or understanding of reality.

6 AUTHORING TOOL DEVELOPMENT

The development of our new authoring tool consisted of a number of phases: research and development, prototyping, implementation, alpha and beta releases, refactoring, and release. In a more granular sense, we went through these phases for each major authoring component of the tool: schema, rules, and actions. Throughout each phase of development and culminating in a week-long synchronous, remote authoring workshop, we solicited feedback, bug reports, and feature requests as we continued to refine the authoring tool.

We began our work with a process of requirements gathering, during which we held a series of meetings with our co-author and "client," a scholar of French literary history and head of the VESPACE project. In these meetings, we proposed a number of possible user interfaces for the tool via white-boarding and received feedback from the client. From these sessions, we derived a set of requirements for the tool that distinguished it from previous authoring tools. From there, we conducted a series of meetings in which we white-boarded a more fully-realized design, which we subsequently built into a prototype schema authoring application. This phase ended with a presentation of the prototype during a guest lecture to literature graduate students and faculty, after which we gathered feedback, questions, and suggestions.

After the phases of requirements gathering, prototyping, and preliminary feedback, we refactored our prototype into an alpha release, and at the end of each weekly sprint we again solicited user feedback, which was used for new feature design, refactoring, and bug fixes. This development phase included a major refactor of a previous rule authoring application written in JavaScript; we restructured and expanded it from a local application for individual authors to a deployed, isomorphic and reactive JavaScript web application capable of handling the work of many authors across many schemas. This development phase included a three-week period of once weekly workshops, during which French studies graduate students used a beta release of the tool to encode historical information into social physics schema predicates and rules. After completing the schema and rule authoring portions of the tool, we then moved on to action and character authoring.

Finally, once the tool had become a means of authoring all four major components of *Ensemble*, we held a week-long synchronous, remote workshop from May 17th, 2020 to May 22nd 2020, which included two professors and seven graduate students from Louisiana State University, University of New Orleans, Tulane, Yale, University of Maryland, and Université de Nantes. Starting with a day of schema authoring, we quickly moved to rule authoring, which occupied the majority of the workshop. Then, the last two days included both action and character authoring. The end result of the workshop was a completely realized schema for VESPACE, as well as many rules, each one sourced from relevant literature or historical sources.

Throughout our development process, we learned many things that both informed the development of our tool and are likely generalizable to similar efforts to empower non-engineers to author in computationally parseable ways. These insights are as follows:

(1) VESPACE authors frequently begin their authoring process by reading and then aggregating specific textual material at the level of the citation, which can then be transformed into computable data. Therefore, we found that during the schema authoring process, it was to some degree more natural for authors to start at the specific level of a type and then further abstract from there in order to develop its enclosing category.
(2) Because schema authoring leverages computer science concepts such as network theory, directionality, first-order predicate logic, and Boolean versus scalar values, it represents an initial hurdle for new authors developing these encoding skills.

(3) At first, the numeric ranges used in defining categories in *Ensemble* presented a challenge when attempting to correspond these values to aggregated textual information from literary-historical sources. Therefore, the easiest solution was to set these ranges by default to 1 - 100. During the action and character authoring phase, the utility of these ranges became more clear, at which point they could possibly be further refined to fit cited information.

(4) Although raw numeric ranges are not immediately useful to new authors, numeric ranges represented by textual sets *are* useful, and the tool should allow users to define custom textual ranges resembling enums, such as the following:
  - $\{Worst, Bad, Neutral, Good, Best\}$
  - $\{Serf, Merchant, Noble, Royal\}$

  Though the end user sees these ranges as textual representations of values or default values, the *Ensemble* engine treats them the same as any other numeric set.

(5) Every authored item– category/type, rule, action, character– can be associated with many citations, by which they can be filtered, sorted, and aggregated. Since researchers must frequently reuse citations– in fact, these citations being as important as the items themselves– they should be searchable and autofillable both partially and complete.

7 NEW AUTHORING TOOL

The new *Ensemble* authoring tool consists primarily of four authoring units– schema, rules, actions, and characters– which we will now discuss individually.

7.1 Schema Authoring

Because it is the foundation of all predicate authoring in *Ensemble*, the schema is also the most abstract and jargon-heavy level of authoring. Due to the structural incongruity between rigid data structures and the freeform prose of source material– in this case period fiction and works of literary history– it made sense to obscure to some degree any one-to-one correspondences with the data structures and computer science terminologies used in schema definition. Thus, rather than having authors build schemata via a single form as in previous authoring tools where authors could, for example, toggle a category as being either "Boolean" or "scalar," we found that the schema editor should instead be presented as a questionnaire. Due to its relationship with ethnographic coding, this format seemed to be a natural fit for social physics coding.

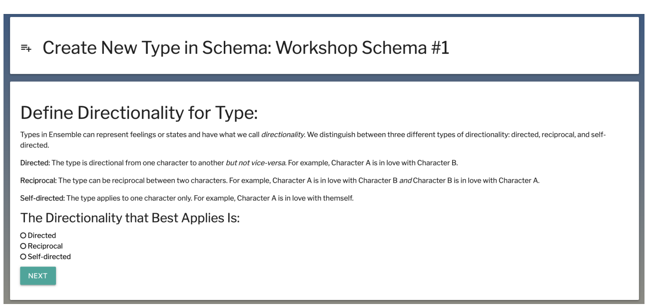

Figure 2: Schema Editor

The first page of the editor presents users with a list of sets of types representing each existing category. The questionnaire prompts the user to read through each set of types and determine if their new type belongs in any of the existing sets; if the type does fit in one, it will be added to the category, and the authoring is done. The tool was designed in this way so as focus on the specific first, in order to leverage the specificity and detail inherent in textual exegesis and citation; its secondary positive effect is to deter users from creating a proliferation of unnecessary new categories.

However, in the event that an author *does* have legitimate need of a new category, each subsequent page of the questionnaire presents a question representing a field in the category datatype, such as *isBoolean* or *directionType*. Authors are presented with a set of plain English examples explaining the concept and asked to pick the closest to the type they have in mind. By completing the questionnaire, the authors are accomplishing two goals, the first being that they are building out the schema, and the second being that they are learning the structures of data representation in *Ensemble*.

Through our series of workshops, the goal of which was to build a working schema, rule set, and action set for the VESPACE game, we determined that a nearly complete schema, unlike a rule set, could be developed in a relatively quick amount of time, which could then be ready for any subsequent workshops. However, we *also* determined that this process was a necessary ritual at the beginning of each workshop, even if our schema was already complete. We have found that working through at least some amount of schema authoring is an important step for authors to understand subsequent rule and action authoring.

7.2 Rule Authoring

The largest holdover from the previous *Ensemble* authoring tool is the rule authoring feature, albeit with a number of important changes and additions. As can be seen in Figure 3, the rule authoring screen consists of two columns. On the right, in a smaller column, is the current active schema; each category is displayed along with its most important fields, as well as its list of types. Actionable types are highlighted in yellow. The left hand column contains the rule editor itself, where users can edit a rule's title, conditions, effects, as well as use a Test Rule feature, where users can execute a rule in a sample environment in order to see its behavior.

The rule editor is split into two major sections. The yellow section at the top of the rule contains the rule's conditions, which can be added, removed, and edited, and which are displayed to users in plain English. In each condition, a dropdown menu allows users to scroll through all available types, which are organized by category. When a user selects a type, the condition automatically updates to reflect its plain English structure, including Boolean toggles for choosing between "is" and "is not." For instance, because the type "jealous of" is a "DirectedStatus," which is a directed category, the plain English condition is displayed as "other is jealous of third," where "other" and "third" are character variables. Using the provided inputs in each condition, authors can craft character relationships for their rules. Importantly, because these character names are variables, which are bound at the time of rule execution, they can apply to any characters meeting the conditions of the rule.

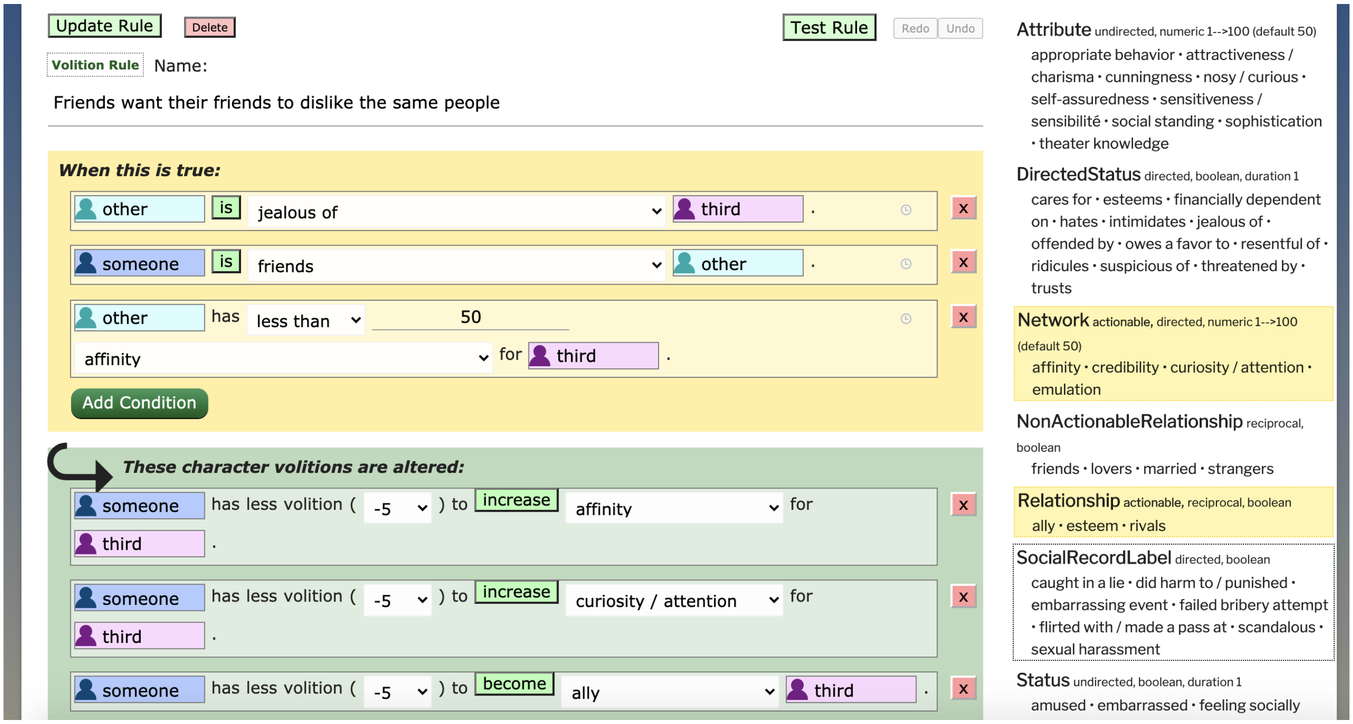

Figure 3: Rules Editor

The green section at the bottom of the rule contains the rule's effects. In the case of a trigger rule, this section is virtually identical to the condition section. For volition rules, however, available types are limited to only those categories which are *actionable*. Figure 3 shows a rule with three effects. In each effect, users can employ a numeric dropdown to set the value by which an effect will increase or decrease volition. In this case, all three effects of the rule cause a -5 decrease in volition, albeit for different intents.

Compared to its predecessor, rule authoring in the new *Ensemble* authoring tool includes a number of important changes and additions. For one, the previous tool was an installable application which would run locally on a user's machine. Persistence of data in the tool was achieved through the use of JSON files; users could load a JSON file to initialize a schema, and any rules they wrote would be saved to a file for future use. As the new tool is a collaborative web application– a necessity in order to facilitate the coordination of work during remotely held workshops across seven time zones– we added user accounts and permissions. Normal user accounts are able to create, edit, and delete their own rules, as well as view the rules of others. All rules are stored in a database, so users can access their work from any machine, anywhere, although they can also be exported to a JSON file if they so choose.

7.3 Action Authoring

Another major addition to the new *Ensemble* authoring tool is the action authoring feature, which can be seen in Figure 4. By default, each actionable type in a schema corresponds to two *intents*, which reflect a change that a character wishes to make to the social state [18]. If the actionable type is numeric, then it will have both an *up* intent and a *down* intent, indicating that the type's value will be either increased or decreased. If, however, the type is Boolean, it will instead have an *on* and an *off* intent, indicating that the type's Boolean value will be toggled. These intents are autogenerated by the authoring tool; for example, the "affinity" type will by default have two intents: "affinity-up" and "affinity-down." These intents are the most abstract layer of an action.

In the tool, it is possible to add any number of *social exchanges* within an intent. These social exchanges externally represent a social interaction that a character can undertake to modify the social state; these are made available to the player in a list, depending on which actions are available at a given step in the game [14]. For instance, "affinity-up" could include the social exchanges "compliment," "help someone," and "give a gift." Internally, social exchanges do not reflect a fully specified action, but rather a general grouping of related actions. As the figure illustrates, the "help someone" social exchange can contain numerous specific *instantiations*. So that actions make sense within the current game state, certain instantiations are only possible within certain contexts; the preconditions of an instantiation determine which one should be chosen [15]. These instantiations are split into two kinds, *accept* and *reject*, which determine respectively whether the action will be a success or a failure. An example of an accept instantiation for "help someone" could be "keep a secret for someone."

Once a user has created an instantiation, it is time to actually construct a specific action. As seen in Figure 4, the editor itself resembles the rule editor, having both yellow and green boxes containing predicates. The yellow box contains all of the conditions

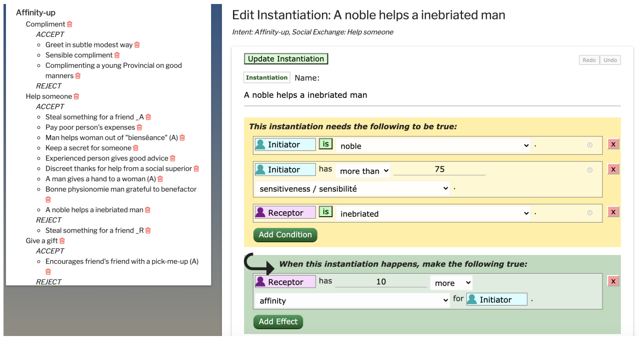

Figure 4: Actions Editor

that must be true in order for a character to be able to perform this instantiation, while the green box contains all that will become true if the action is performed. Typically, accept and reject instantiations in the same social exchange will contain different effects, so as to illustrate the relative effects of an action's success or failure. In order to represent character directionality in an action, we use the character variables "initiator" and "responder," where the initiator is the subject taking the action, and the responder is the object of the action. Other than these special character terms, editing of the conditions and effects of an action instantiation is nearly identical to that of a rule, which makes it easy for action authors to leverage their existing knowledge of rule authoring.

7.4 Character Authoring

The last major feature of the *Ensemble* authoring tool is the character editor, where users can create a cast of characters for their game or experience. Creating a character is straightforward and can be seen in Figure 5. After giving the new character a name, authors can build up a list of predicates that describe that character's traits, beliefs, behaviors, and relationships at the start of the game.

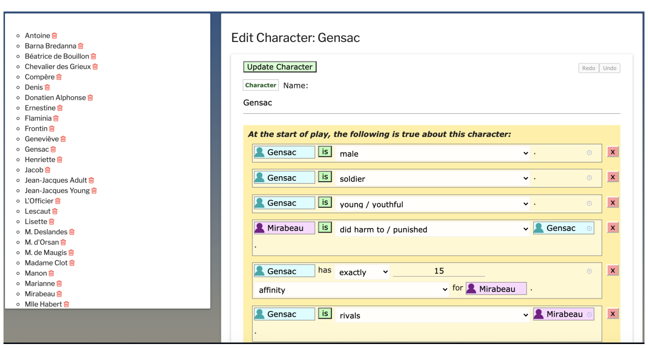

Figure 5: Characters Editor

8 FUTURE WORK

The VESPACE project will require a number of important changes to the *Ensemble* engine itself. As part of the project, we have already completed a translation of *Ensemble* from JavaScript to C# to facilitate development in Unity. We intend to begin incorporating *Ensemble* into various Unity projects, starting with VESPACE, including developing detailed API documentation which will be useful for Unity developers adding the library to their projects.

In order to implement methods of *citation as AI* within the *Ensemble* engine, we intend to also modify it so that rules, actions, and characters can be filtered by their sources. This added feature will make it possible to "activate" or "deactivate" rules, actions, and characters so as to execute different versions of the VR experience that present a social experience through the lens of a particular author, researcher, or methodology. Or, rather than completely "deactivating" a set of rules, *Ensemble* could weight them relative to other rules. For example, rules citing fiction sources might be weighted lower than those citing nonfiction sources, or vice versa. In this case, both sets of rules would have a chance of executing, but one with a lower probability than the other. A third way to weight rules could be to weight the effects of a volition rule– that is, if a volition rule adds +10 to a volition as an effect, that value of 10 could be multiplied by some factor indicating the rule's intended weight, with 1 being normal, 0 being deactivated, 0.5 being less powerful, and 2 being very powerful.

One important addition to the authoring tool will be the inclusion of guidelines and suggestions to help social physics authors to avoid crafting rules and actions too detailed to the point of becoming intractable. Currently, we must instruct authors to consider complexity in rule and action preconditions, and we manually author default actions which require few to no preconditions and which will lead to the binding of certain rules. However, by computing and also visualizing the interconnections of rules and actions *during* the authoring process, we could provide users with estimates and visual representations of the likelihood that a particular rule will fire or a particular action will become available to a character. Thus, we could notify authors when a rule or action appears overly complicated or intractable, and we could visually alert them of weak points in the interactive narrative. We also plan to simplify the generation of default actions with few to no preconditions, allowing authors to focus on writing descriptive actions.

9 CONCLUSION

The VESPACE project represents a step forward in the application of social physics. It stands to produce a novel gaming experiment and pedagogical device– an interactive, virtual social experience transporting users back hundreds of years. However, it would not be possible without our work in collaborative authoring; through the development of our new authoring tool, we have facilitated the creation of thousands of complex social rules that will make this game a reality, *without* forcing authors to learn first-order predicate calculus. Our findings in this project suggest that it is possible to train authors of various backgrounds in social physics and to have them produce a working corpus of domain-specific schemata, rules, and actions within a relatively short period of time. Whether in academia or in the industry, future projects could use similar tools and techniques to generate their own domain-specific social worlds, whether large or small, which they can then bring to life in their games by leveraging the computational expressiveness of libraries like *Ensemble*.

ACKNOWLEDGMENTS

The week long VESPACE workshop was sponsored by National Endowment for the Humanities Digital Humanities Advancement Grant HAA-266501-19.

REFERENCES

[1] Electronic Arts. 2000. The Sims. [PC CD-ROM].

[2] Ruth Aylett. 2000. Emergent narrative, social immersion and "storification". In *Proceedings of the 1st International Workshop on Narrative and Interactive Learning Environments*. 35–44.

[3] Sherol Chen, Mark J Nelson, Anne Sullivan, and Michael Mateas. 2009. Evaluating the Authorial Leverage of Drama Management.. In *AAAI Spring Symposium: Intelligent Narrative Technologies II*. 20–23.

[4] Paul François, Florent Laroche, Jeffrey Leichman, and Françoise Rubellin. 2018. Mettre en place les fondations d'un outil de restitution à large audience: VESPACE. (2018).

[5] Gonzalo Frasca. 2001. The Sims: Grandmothers are cooler than trolls. *Game Studies* 1, 1 (2001).

[6] Jacob Garbe, Aaron A. Reed, Melanie Dickinson, Noah Wardrip-Fruin, and Michael Mateas. 2014. Assistance Visualizations for IceBound , A Combinatorial Narrative.

[7] Daniel Green, Charlie Hargood, and Fred Charles. 2018. Contemporary issues in interactive storytelling authoring systems. In *International Conference on Interactive Digital Storytelling*. Springer, 501–513.

[8] Manuel Guimaraes, Pedro Santos, and Arnav Jhala. 2017. CiF-CK: An architecture for social NPCS in commercial games. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 126–133.

[9] Manuel Guimarães, Pedro Santos, and Arnav Jhala. 2017. Prom Week Meets Skyrim.. In *AAMAS*. 1790–1792.

[10] Henry Jenkins. 2004. Game Design as Narrative Architecture. *Computer* 44, 53 (2004), 118–130.

[11] Hartmut Koenitz. 2019. Narrative in Video Games.

[12] Hartmut Koenitz and Mirjam Palosaari Eladhari. 2019. Challenges of IDN Research and Teaching. In *International Conference on Interactive Digital Storytelling*. Springer, 26–39.

[13] Josh McCoy, Mike Treanor, Ben Samuel, Michael Mateas, and Noah Wardrip-Fruin. 2011. Prom Week: social physics as gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*. 319–321.

[14] Joshua McCoy, Mike Treanor, Ben Samuel, Aaron A. Reed, Michael Mateas, and Noah Wardrip-Fruin. 2013. Prom Week: Designing past the game/story dilemma. In *FDG*.

[15] Joshua McCoy, Mike Treanor, Ben Samuel, Aaron A Reed, Michael Mateas, and Noah Wardrip-Fruin. 2014. Social story worlds with comme il faut. *IEEE Transactions on Computational intelligence and AI in Games* 6, 2 (2014), 97–112.

[16] Josh McCoy, Mike Treanor, Ben Samuel, Aaron A Reed, Noah Wardrip-Fruin, and Michael Mateas. 2012. Prom week. In *Proceedings of the International Conference on the Foundations of Digital Games*. 235–237.

[17] Josh McCoy, Mike Treanor, Ben Samuel, Brandon Tearse, Michael Mateas, and Noah Wardrip-Fruin. 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*, Vol. 50. Citeseer.

[18] Joshua McCoy, Mike Treanor, Ben Samuel, Noah Wardrip-Fruin, and Michael Mateas. 2011. Comme il faut: A system for authoring playable social models. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

[19] Françoise Rubellin and Paul François. 2018. Le théâtre du XVIIIe siècle, plus vivant que jamais. (2018).

[20] Benjamin Samuel. 2016. *Crafting Stories Through Play*. Ph.D. Dissertation. UC Santa Cruz.

[21] Ben Samuel, Dylan Lederle-Ensign, Mike Treanor, Noah Wardrip-Fruin, Josh McCoy, Aaron Reed, and Michael Mateas. 2015. Playing the worlds of prom week. In *Narrative Theory, Literature, and New Media*. Routledge, 87–105.

[22] Ben Samuel, Aaron A Reed, Paul Maddaloni, Michael Mateas, and Noah Wardrip-Fruin. 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.

[23] Yotam Shibolet, Noam Knoller, and Hartmut Koenitz. 2018. A framework for classifying and describing authoring tools for interactive digital narrative. In *International Conference on Interactive Digital Storytelling*. Springer, 523–533.

[24] Chris Swain. 2008. The augmented conversation engine: a system for achieving believable conversation in games and interactive stories. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. 213–218.

[25] Nicolas Szilas, Thomas Boggini, Monica Axelrad, Paolo Petta, and Stefan Rank. 2011. Specification of an open architecture for interactive storytelling. In *International Conference on Interactive Digital Storytelling*. Springer, 330–333.

[26] Mike Treanor, Josh McCoy, and Anne Sullivan. 2015. Social play in non-player character dialog. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.