SPA: Verbal Interactions between Agents and Avatars in Shared Virtual Environments using Propositional Planning

Andrew Best*
Dept. of Computer Science
UNC Chapel Hill

Sahil Narang†
Dept. of Computer Science
UNC Chapel Hill
http://gamma.cs.unc.edu/pedvr/

Dinesh Manocha‡
Dept. of Computer Science
University of Maryland

ABSTRACT

We present a novel approach for generating plausible verbal interactions between virtual human-like agents and user avatars in shared virtual environments. Sense-Plan-Ask, or SPA, extends prior work in propositional planning and natural language processing to enable agents to plan with uncertain information, and leverage question and answer dialogue with other agents and avatars to obtain the needed information and complete their goals. The agents are additionally able to respond to questions from the avatars and other agents using natural-language enabling real-time multi-agent multi-avatar communication environments.

Our algorithm can simulate tens of virtual agents at interactive rates interacting, moving, communicating, planning, and replanning. We find that our algorithm creates a small runtime cost and enables agents to complete their goals more effectively than agents without the ability to leverage natural-language communication. We demonstrate quantitative results on a set of simulated benchmarks and detail the results of a preliminary user-study conducted to evaluate the plausibility of the virtual interactions generated by SPA. Overall, we find that participants prefer SPA to prior techniques in 84% of responses including significant benefits in terms of the plausibility of natural-language interactions and the positive impact of those interactions.

Index Terms: Computing methodologies—Artificial intelligence—Distributed artificial intelligence—Multi-agent systems;

1 INTRODUCTION

There is great recent interest in generating immersive social experiences. Increasingly, games, training, and entertainment seek to provide a user with the experience of embodying a digital *avatar* and sharing a virtual space with other user-controlled avatars as well as computer-controlled characters, or *agents*. Such multi-agent multi-avatar applications range from immersive games, social virtual-reality (VR) hangouts, training simulations [15, 34], treating social-phobias [30] or visiting virtual spaces such as museums or landmarks.

The plausibility and effectiveness of multi-avatar simulations can be improved by the presence of interactive human-like virtual agents [12]. However, virtual agents that do not interact in plausible ways can reduce the sense of presence in virtual environments [1, 38]. Moreover, the context of the simulation may necessitate agents that have independent goals and are not purely focused on the co-present avatars. The virtual world should feel like a place the avatar is visiting, as opposed to one constructed purely for the avatar. In such cases, agents must be capable of engaging in meaningful interactions with avatars and other agents, either proactively or in response to the actions of others. These interactions may include both verbal as well as non-verbal means of communication including movement and navigation, gesturing, gazing etc. Recent studies have highlighted the critical role of verbal communication and its significant impact on the perceived naturalness of user-agent interactions, and the overall effectiveness of the application [25, 26].

Most prior work in enabling interactions between avatars and agents is limited to embodied conversational agents (ECA), wherein an anthropomorphic virtual agent demonstrates human-like face-to-face communication [5]. However, ECA is generally restricted to single agent-avatar pairwise interactions and is often avatar-centric. The agent participates in interaction with the aim of assisting the avatar in achieving a goal, or foiling the avatar, but does not plan its own intentions outside the context of the avatar-agent interaction. There is also prior work in multi-agent navigation that has explored communication behaviors [18, 29]. However, these methods rely on message-passing or implicit communication which preclude verbal interaction with user-controlled avatars. Overall, simulating plausible verbal interactions in shared multi-avatar multi-agent environments remains a challenge.

There are several core challenges in simulating the behaviors of virtual agents in such multi-avatar multi-agent environments. First, agents must be capable of independently planning egocentric behaviors in potentially uncertain conditions. Much like the real world, an agent may possess an imperfect understanding of the world and must be capable of proactively communicating with other entities to derive knowledge such that it can accomplish its goal.

Second, agents must be capable of communicating with avatars and other agents in unstructured conditions. In effect, agents must be able to interpret language, generate meaningful responses and exchange information, agnostic of whether the other entity is a user-controlled avatar or another virtual agent.

Third, agents must be capable of generating plausible behaviors, including asking and answering questions, based on their interpretable understanding of the virtual world. In effect, agents should be able to absorb information through communication and behave appropriately based on the new information.

Main Results: In this paper, we seek to address the problem of simulating many virtual agents that can effectively plan individual actions, interact, and communicate with avatars and other agents using natural language. To this end, we present *Sense-Plan-Ask (SPA)*, an interactive approach to enable virtual agents to accomplish their individual goals with uncertain information in complex multi-agent multi-avatar environments (Section 3). The SPA approach consists of following novel contributions:

- **Propositional-planning with Automatic Uncertainty Resolution**: We present a least-commitment-based planning approach [35] to generate agent action plans with uncertain information. Agents automatically generate uncertainty resolution actions that may include navigational actions to explore the environment, or asking questions. Moreover, agents re-plan based on new information.

- **Multi-agent Natural language Interaction**: We present a

*e-mail: best@cs.unc.edu
†e-mail: sahil@cs.unc.edu
‡e-mail: dm@cs.umd.edu

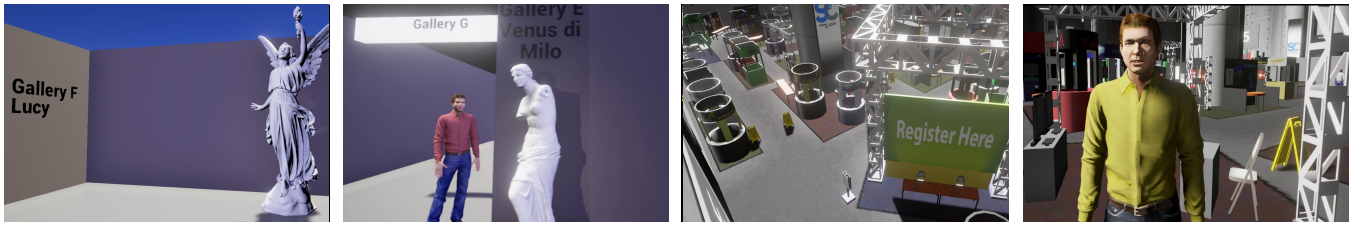


Figure 1: We performed a user-study to evaluate our novel method for generating verbal interactions between virtual agents and user-avatars. Participants in our study preferred our approach, SPA, in 84% of responses. Participants also indicate strong preferences for our approach in terms of plausibility of interactions and how well the scenario reflected real-world scenarios. Our study consisted of two trials. **(A)**: In the museum, the avatar searches for the statue of Lucy in a gallery at the far-side of the museum. **(B)**: While exploring, agents approach the avatar to ask the location of other statues the avatar has seen previously. The avatar can choose to provide information to the agents. **(C)**: In the tradeshow, the avatar must find the registration booth pictured. **(D)**: Using our method, the avatar can ask virtual agents for the location of the registration booth.

natural language communication approach that can parse utterances received from other agents and avatars, generate natural language responses as well as construct queries and learn new information based on propositional logic.

- **Proactive agents**: Our approach, SPA, allows agents not only to react to avatars and agents, but to proactively seek out interaction and engagement. The agents learn from interaction and respond accordingly, generating diverse and comprehensive simulations.

- **User Evaluation**: We present the results of a user study which demonstrates our method's advantages over prior approaches. Compared to methods which do not enable natural-language interaction between agents and avatars, participants showed significant preference for our method in terms of the plausibility of the scenarios and quality of agent-avatar interactions.

The rest of the paper is organized as follow: In Section 2, we detail relevant related work in multi-agent systems and task planning. We give an overview of SPA in Section 3. In Section 4, we detail our propositional-planning framework which enables planning under uncertainty via interaction. Section 5 details our natural-language processing and generation approach. We describe our simulation benchmarks, offer performance results, and detail the results of a user evaluation of our method in Section 6.

2 Related Work

In this section, we give an overview of relevant work in action-planning, multi-agent simulation, and verbal communication.

2.1 Action Planning

Action planning, sometimes referred to as classical planning, task planning, or logical planning in the literature [13,35], deals with constructing a set of actions taken by an intelligent agent to solve a problem or achieve a goal. Classical approaches such as STRIPS [10], ADL, [28] and PDDL [11] construct domain description formalisms which allows many distinct domains to be solvable with a common approach. Recent work has addressed uncertainty through continual planning [4], or by encoding sensing actions [31]. The latter approach encodes a specific "sensing" action for each kind of information the agent may need. Our approach seeks to minimize explicit sensing actions, instead employing natural-language interactions and generic exploration to resolve uncertainty. Epistemic planning approaches [9,21], typically employed in agent-interaction scenarios, typically solve uncertainty via each agent modeling their perception of the beliefs of other agents in a turn-based interaction. These models are inherently complimentary to our overall algorithmic approach. However, it is unclear how an agent would determine the best question to ask given an epistemic model or how "overhearing" and indirect communication are modeled in these cases.

2.2 Human and Single Agent Interactions

Interactions between humans and automated agents can be subdivided into human-robot interactions (HRI) and human interactions with a digital or virtual agent, also called Embodied Conversational Agents (ECA).

In the domain of HRI, robots typically plan dialog to clarify a human operator's intentions in a collaborative context [2, 8]. The robot does not maintain independent goals or a non operator-centric representation of the domain. The robot works as a subservient collaborator, typically limited to interaction with a single human operator. Recent work has demonstrated a robot requesting assistance in a specific collaborative-task [41]. Our framework is complimentary to the proposed language model and could be used with such a system for embodied conversation on a physical robot. Some work has incorporated sensor uncertainty [42], but modeling unknown information remains a challenge across domains. Our approach allows agents to act on their own goals and employ natural-language interactions with multiple avatars or agents by specifying a range of domain information types without fully specifying the agent's knowledge.

Prior work in Embodied Conversation Agents (ECA) has demonstrated a single agent communicating with a human avatar using complex but well-structured dialogue [14]. Rickel and Johnson [34] demonstrated positive effects on team training when using virtual agents in a mixed agent-avatar environment. In these cases, the agent's behavior is fundamentally user-centric i.e. rather than behaving as an independent entity, the agent's plan revolves around the avatar. Some work has demonstrated capabilities in learning from natural language interaction [43, 47]. However, these methods limit the interactions to a single user and do not generalize to unstructured interactions.

2.3 Multi-agent Planning

Action planning for interactive multi-agent systems has typically been limited to locomotion-based actions, and comprises of choosing an appropriate goal position for each agent and computing its trajectory. A large body of research exists on planning paths to an agent's goal [39, 46] and local-avoidance behaviors [17, 36, 45] for agents. Our work is complimentary to these approaches and our agents can use any of these methods for path planning. However, many of these approaches rely on rigid finite-state machines to generate goals for the agents [7, 44]. Some approaches have been proposed to incorporate contextual interactions [27, 37], but in these cases the behaviors of the agents are pre-encoded and activated when specific conditions are met. Instead, our approach allows for dynam-

ically creating action plans for agents without a rigid or pre-encoded structure. Our agents generate and execute plans as needed to satisfy diverse goals. Agents can plan simple goal positions, or complex goals, i.e. interacting with a specific agent or acquiring some item from the environment.

2.4 Communication in Multi-agent systems

Some prior work has introduced communication capabilities in multi-agent systems using message-passing or packet-based approaches to model interaction [18], auditory cues [16], social contagion [6], or information sharing [29]. In each approach, agents communicate using a strict message structure, or implicitly through the sharing of data and thus preclude the use of interactive verbal communication. Sun et al. [40] generated animated conversation for agents in a simulated crowd, and Brenner and Kruijff-Korbayova [3] leveraged message-passing to enable multiple agents to collaborate with a simulated user (i.e. not an actively controlled avatar) in a shared-task. In each case, the natural-language dialog was generated post-simulation as an animation feature. By contrast, our approach allows the agents to communicate with other agents and avatars using plausible verbal communication in real time.

3 Background and Algorithm Overview

Our approach, Sense-Plan-Ask, couples propositional planning with natural-language processing to enable many agents to interact with other agents and avatars in shared environments. This section introduces relevant terms and notation used throughout the chapter, and provides an overview of our approach.

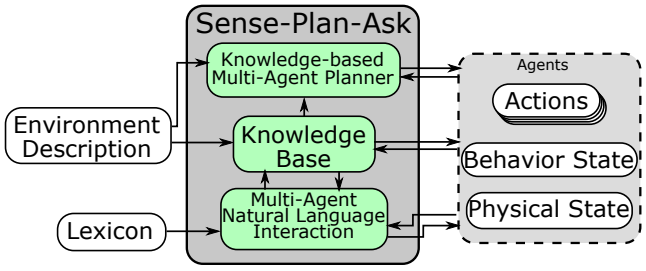


Figure 2: **Algorithmic Pipeline**: We present a novel interactive approach called Sense-Plan-Ask (SPA) which enables agents to interact with other agents and user-controlled avatars using natural language communication. Each agent accomplishes individual goals using SPA to compute explainable action plans based on uncertain information. The agent engages in natural language conversations with nearby agents and avatars to resolve uncertainty in its plan. The communication model increases the plausibility of the simulation and improves the user's experience in immersive virtual environments.

3.1 Agent Model

Each agent in the virtual environment can be defined by its *physical state* and its *behavioral state*. The physical state comprises of physical properties such as position, velocity and its current action. The *behavior state* regulates its decision making, including its current knowledge and the set of available actions.

3.1.1 Physical State

For the purpose of behavior and movement planning, we treat each agent i as a bounded disc in a 2D plane with scalar radius r_i. We denote the agent's position and velocity at time t by $\vec{p}_i^t$, and $\vec{v}_i^t$ respectively. Furthermore, we denote the current action being executed by the agent as $a_i^t | a_i^t \in \mathcal{A}_i$. $\mathcal{A}_i$ is the set of all actions available to the agent and may include movement or navigation as well as non-movement actions, such as speaking, or interacting with the environment, i.e. affecting change to the simulation domain. We assume a path planner is available for locomotion and is used to compute the path and avoid collisions with other agents, avatars, and obstacles in the environment. Collectively, the position, velocity, and current action describe the agent's time-varying physical state $\{\vec{p}_i^t, \vec{v}_i^t, a_i^t\}$.

3.1.2 Behavior State

We conceptualize our agents as Belief Desire Intention (BDI) agents [32]. Each agent maintains an independent understanding of the world represented by a set $\mathcal{B}$ of facts called *beliefs*. We use the term belief to reflect the fact that the agent's knowledge may be wrong. Moreover, each agent is given a set $\mathcal{D}$ of high-level *desire*s, or goals, to achieve in the simulation. Each agent plans a set of intermediate actions called *intentions* to accomplish these desires.

We encode the agent's beliefs, desires, and intentions in a first-order propositional planning language and store the agent's knowledge in relational database. Our formalism is a subset of ADL [28] with the exception that we do not allow disjunction in desires, and limit desire specification to grounded literals. First-order propositional planning provides sufficient representational power for our interactive benchmarks (see Section 6) and fits well with the types of questions we support in our interaction. We define a set Σ comprising of *entities* that are symbolic constants through the simulation. These may include agents, items, physical locations, etc. We apply a *type* to each entity and a set of attributes associated with the type.

We describe predicates as first-order formulae over Σ. We allow constraints on the type of arguments in the predicate schema to reduce the search space, and we explicitly allow negated predicates in state specification. Furthermore, we categorize predicates in two categories. *Knowledge predicates* describe relationships or facts the agent might know. *Fluent predicates* describe transitive properties the agent may hold, i.e. being at a specific location [35]. A belief-state for agent i at time t, denoted $\mathcal{B}_i^t$, consists of the set of all beliefs known to be true or false to the agent. For compactness, we assume Σ is known to all agents and is implicitly included in $\mathcal{B}$. Consider a problem-solving domain consisting of a series of keys and locks, the following would be a valid state specification:

$$\mathcal{B}_i^t = \{Have(Key_1), \neg Have(Key_2), Opens(Key_1, Safe_1), \\ \neg Locked(Safe_2)\}$$

We do not assume a complete state specification. That is, $\mathcal{B}_i^t$ contains all *known* information. Missing predicates are considered unknown as opposed to false. In the example above, agent i knows that Key_1 opens $Safe_1$, but does not know of anything which opens $Safe_2$. In Section 4, we detail how our planning approach allows agents to plan uncertainty resolution and in Section 5 we describe how they use verbal communication to resolve uncertainty.

We define a set of operators, or *actions* $\mathcal{A}$, over beliefs of the form $O < parameters, conditions, effects >$. Parameters describe elements in Σ passed to the action, subject to type constraints on the element. Conditions are predicates which must hold in $\mathcal{B}_i^t$ for the action to be applicable. Effects are predicates added to $\mathcal{B}_i^t$ upon application of the action. Should a predicate in $\mathcal{B}_i^t$ be contradicted by new information, it is removed corresponding to the agent updating its beliefs. Continuing from the earlier example, the following would be a valid action schema corresponding to opening the safe:

$$Open < \{key : X, safe : Y\}, \\ \{Locked(Y), Opens(X, Y), Have(X)\}, \\ \{\neg Locked(Y)\} >$$

The desires of the agent are a time-varying set $\mathcal{D}_i^t$ of beliefs which must be achieved by actions over the agent's initial state,

$\mathcal{B}_i^0$. The agent's behavior state for time t is compactly described as $\{\mathcal{B}_i^t, \mathcal{D}_i^t, \mathcal{A}_i^t\}$.

3.2 Sense-Plan-Ask (SPA) Algorithm

Our proposed approach, Sense-Plan-Ask, generates plausible interactions between virtual agents by coupling a novel propositional planner with a natural-language processing framework. SPA is an agent-based simulation algorithm which enables the agents to communicate, plan, and interact with other agents and avatars. Figure 2 details our algorithmic pipeline.

Sense: We conceptualize the simulator as a discrete in time and continuous in space. The simulation updates at a fixed rate, Δt. Each update, the agents in the simulator "sense" their surroundings. They observe relevant predicates associated with nearby entities within a range, and "hear" utterances produced by agents and avatars in their vicinity. Based on the observations, the agents update their internal knowledge representation and react accordingly.

Plan: Each agent plans a series of actions to accomplish its goals. Section 4 describes how these plans are constructed based on uncertain information, and how the planner generates disambiguation actions such as asking questions of nearby agents or avatars to resolve the uncertainty. The planner allows the agents to process new information and re-plan rapidly, as is detailed in Section 6.

Ask: The natural-language approach combines shallow semantic parsing with template-based generation to produce intelligible verbal questions and answers between agents and avatars. Section 5 details how agents learn new information from their interactions and use that information to resolve uncertainties in their plans. In addition, agents can interact with avatars and other agents verbally by engaging in question and answer-based dialogue.

Each simulation environment is configured from a domain specification file, which contains the set of entities, predicate schema, action schema, and initial knowledge for each agent in the environment. This specification also gives initial conditions of agents and other objects in the environment. This domain description is paired with an English lexicon to automatically generate training data for a shallow-semantic parser as described in Section 5. This domain specification is also used to construct a query-able knowledge-base, encoded in a SQL database, for each agent which allows the agents to recall and respond to changes in the environment.

In the following sections, we describe our algorithms for linking propositional planning and natural-language parsing and generation. The SPA approach is general, however, and could be extended to other planning approaches or natural-language generation methods with some necessary adaptation of the translation mechanism from uncertainty to natural-language utterance.

4 Knowledge-based Multi-Agent Planning with Incomplete Information

Our proposed planning algorithm enables each agent to plan actions to achieve its desires based on its current, potentially incomplete set of beliefs. Our algorithm tracks the uncertainty in the agent's plan and determines new actions such as verbal interaction or exploration of the environment to resolve the uncertainty. Moreover, it enables agents to update their beliefs in real-time and re-plan based on their new set of beliefs.

4.1 Two-stage Action Planning with Incomplete Information

In many propositional planning languages, predicates absent in the state description are considered false, and uncertainty is prohibited [13]. Typical first-order propositional planning approaches may allow incomplete specifications, but require an explicit action for determining the truth state of each individual pre-condition of an action. This can lead to a combinatorial explosion in the number of actions and subsequently the planning time of the approach. [35]. Our approach addresses these limitations and allows for efficient planning under uncertainty by leveraging a two-stage planning algorithm.

Given agent i's desire set, $\mathcal{D}_i$, a set of actions must be constructed to satisfy all of the desires. We apply a least-commitment, backward state-space search approach [35]. Each planning step may comprise of multiple iterations. In each iteration j, the agent chooses an action which satisfies the first desire $d_0 \in \mathcal{D}_i^j$. A new desire set $\mathcal{D}_i^{j+1}$ is created consisting of the remaining unsatisfied desires and any unsatisfied pre-conditions of the chosen action.

As described in Section 3, we differentiate between knowledge and fluent predicates when computing $\mathcal{D}_i^{j+1}$. Knowledge predicates present in $\mathcal{B}_i$ are considered satisfied, and any absent from $\mathcal{B}_i$ are added to an uncertainty set, U_i, as opposed to the new desire set, $\mathcal{D}_i^{j+1}$. Any other arguments of the action aside from those needed to satisfy the desire are left unbound, i.e. they are not assigned any entity. For each unbound argument k, a candidate set of all entities which may satisfy k is constructed, termed C_k. Candidates are chosen based on two criteria: they must satisfy any type and property constraints of the predicate, and, given the candidate binding, all pre-conditions of the action must be either hold in B_i (true) or be absent from B_i (unknown).

Our planning algorithm continues in this fashion, achieving desires until either the desire set is empty, $D_i^j = \emptyset$, or the desire set represents a subset of the agent's current belief state, $D_i^j \subset B_i$. The result is a plan-template, $\mathcal{P} = \{A_i^{t_0}, ..., A_i^{t_n}\}$, i.e. a sequence of actions which satisfy the agent's desires, a set of candidates for all unbound arguments $\mathcal{C} = \{C_k | k \in K\}$, and a set of unknown predicates,U_i, associated with the plan. The action order is inverted for execution, consistent with the backward state-space search.

The second stage of our planning algorithm finds appropriate bindings for the candidates in $\mathcal{C}$. We first construct a set of grounded plans such that all candidates are given a specific binding. These plans are sorted according to the number of predicates from U_i found in B_i given the candidate bindings. In effect, the agent prefers plans with the least uncertainty. For each predicate in U_i not found in B_i, an uncertainty resolution action is inserted into the plan prior to the first occurrence of the predicate. Uncertainty resolution actions include exploring the environment or asking questions. The final plan now consists of the original actions in $\mathcal{P}$ and an uncertainty resolution for each unknown predicate. We refer the reader to the supplemental material for a diagram of our planning approach.

4.2 Plan Execution and Re-planning

Each action described in the problem domain is mapped to a simulation controller in the agent. The controller is responsible for executing the action in the simulation environment. These controllers include uttering, moving to a location, interacting with the environment, waiting a specific amount of time, etc.

If a controller fails to accomplish the specified action for the agent, or the agent is unable to acquire the the information needed for an uncertain belief, the plan binding fails. If other bindings are available for all entities in $\mathcal{C}$, the next binding in order of uncertainty is chosen. Each time an agent acquires new information, the uncertainty of remaining bindings is updated and the set of candidates adjusted to prevent repetitive questions.

If no suitable bindings are available, the planner discards the plan template and back-tracks to the prior branch in the backward state-space search. If no additional branches can be chosen, the plan is discarded and the planner fails. The agent waits a pre-defined amount of time before restarting the planning procedure. These failures are often caused by a failure to acquire information, and are likely to succeed on subsequent planning attempts as other agents and avatars move near the agent.

Table 1: Sample mapping from the shallow parser to knowledge queries in the museum benchmark (Section 6). Utterances are parsed into NL-Is (natural-language intentions) and NLEs (natural language entities). Entities are matched to relationships and a knowledge query is constructed.

Utterance	NL-I	NLEs	Mapped Belief
where is the venus de milo	predicate question	knowledge entity: venus de milo predicate: where	InSpace(?,Venus)
What material is the venus de milo	attribute question	knowledge attribute: material knowledge entity: venus de milo	statue(Venus).material=?
venus de milo is located in gallery a	predicate answer	knowledge entity: venus de milo predicate: located knowledge entity: gallery a	InSpace(Venus, GalleryA)

Figure 3: **Simulation Benchmarks**: We demonstrate our algorithm's performance in a set of simulated environments. **(A)** Agents in the anti-podal circle scenario search for items while requesting information from one another. **(B)** Agents evacuate a building during an emergency. The right-most agent acts as a first-responder, warning the other agents to avoid the fire-blocked hallway. **(C)** Agents explore a densely-packed tradeshow scene, visiting booths and exhibits. The agents request location and booth information from one-another to resolve uncertainty in their plans and reach their desired goals more effectively. **(D)**: A user explores the tradeshow with a first-person view in immersive settings. Our approach allows virtual agents to interact with both agents and avatars simultaneously. **(E)**: The user's avatar (shown from a third-person view) interacts with a virtual agent face-to-face in the tradeshow.

5 Natural Language Interaction between Virtual Agents and Avatars

This section details our natural-language processing and generation approach that allows virtual agents to engage in natural-language interactions with other agents and user-controlled avatars. This includes responding to agent or avatar questions and statements, as well as posing questions which resolve uncertainty in the agent's plans and facilitate achievement of the agent's goals.

5.1 Parsing Natural Language Utterances

To understand incoming utterances, our agents leverage state of the art shallow semantic parsing [33]. Shallow semantic parsers are trained on a corpus of example sentences to label the sentence with an "intention" and extract from it natural-language *named-entities* [20]. To avoid confusion with knowledge entities, we refer to named-entities as NLE (natural-language entity) and intentions as NL-I (natural-language intention).

The NL-I of an utterance is a label which is used to categorize the utterance and determine an appropriate response. We provide training examples for specific questions and answers our agents should be capable of responding to. We refer to these as in-domain NL-I. They are: predicate question, predicate answer, attribute question, and attribute answer. Aside from domain specific NL-I, we provide example data for five generic NL-I, which we refer to as out-of-domain NL-I. These are: greeting, thanks, farewell, affirmation, and fallback (i.e. random dialogue). Section 5.1.1 details how we generate training sentences automatically for our in-domain NL-I.

NLEs are specific words in an utterance which are considered important for understanding its meaning. Typically, example sentences for each NL-I are also annotated with the relevant NLEs. We specifically provide training data for five NLEs: attribute instances, attribute types, predicate types, knowledge entities, and addressees. Section 5.1.1 details how we generate training sentences automatically for our target NLEs.

As an example, the utterance "Where is object A?" would receive the NL-I *predicate question* as it relates to the location of the object. The parser would also recognize two NLEs, "object A" of type *knowledge entity* and "where" of type *predicate instance*.

5.1.1 Training the parser

We generate training data by coupling our domain descriptions with an English lexicon. The lexicon provides part of speech information for the set of words in our problem-domain as well as subjective and objective verb tense information. The lexicon also provides a set of sample usage sentences which we annotate with NL-Is and NLEs. To train the parser, sample sentences are drawn from the lexicon and the template parameters are bound to corresponding entries from the knowledge base. We also provide a set of basic responses for the set of out-of-domain NL-Is. Limiting the shallow parser to few NL-Is and NLEs allows us to train the parser using tens of examples rather than hundreds or thousands used for modern voice assistants.

For the results demonstrated in Section 6.1, we created a custom, limited lexicon. However, our method would generalize to a common lexicon provided that the sentence templates could be extracted. Recent work [19] has demonstrated the ability to extract planning domains automatically from text and may provide a potential avenue for automatic tagging of lexical entries.

5.2 Understanding Utterances from Avatars and Other Agents

Each agent "hears" utterances issued by other avatars and agents that are visible with respect to obstacles and are within a tunable hearing range. Each utterance is parsed and the NL-I and NLEs are returned. For out-of-domain NL-I, the agent responds with one of the example responses provided in the lexicon.

To map an utterance to the planning framework, the recognized NL-I must be an in-domain NL-I and the utterance must contain at-least one NLE of type *predicate instance*, *predicate type*, or *knowledge attribute*. Each entity in Σ is required to have a matching NLE in the lexicon. The agent maps the recognized NLEs to their knowledge-base equivalents. The agent constructs a belief from the

Table 2: **Benchmark comparisons with and without NL-I. This table compares performance of our algorithm with and without NL-I. We find that agents who are able to communicate accomplish their goals more quickly and benefit from nearby communication of other agents.**

Scene	Agents	Planning Time	Replans	Replan Time (S)	Solution Time(S)
Anti-podal Circle	10	76.390	4	0.007006	67.85
Without NL-I	10	75.917	4	0.005508	73.95
Evacuation	11	1.009	10	0.000278	36.60
Without NL-I	11	1.771	10	0.000263	53.70
Museum	5	6.811	9	0.013800	159.35
Without NL-I	5	12.361	2	0.006314	209.40
Trade Show	4	0.123	1	0.001375	52.65
Without NL-I	4	0.128	1	0.001116	98.95

entities according to whether a predicate or attribute was detected. For predicates, entities are matched to the slots of the predicate. For attributes, entities are mapped to the attribute relationship.

Consider this statement generated from the example above, "Key one opens safe one". The NLE "opens" would map to knowledge *predicate type*. "Key one" and "Safe one" would map to *knowledge entities*. The planner would construct the complete predicate instance $Opens(Key_1, Safe_1)$.

In the case of questions, if a slot is missing from the constructed predicate or attribute belief, the missing information is used as the subject of the question. If no information is missing and a complete belief can be constructed, the question is assumed to be a confirmation question. The question "Does key one open safe one?" would receive the NL-I predicate_question and same NLEs as the statement form. The NLE "opens" would map to knowledge *predicate type*. "Key one" and "Safe one" would map to knowledge *knowledge entities*. In this case, the complete predicate $Opens(Key_1, Safe_1)$ would be interpreted as a confirmation question. Similarly, the question "Which key opens safe one" would map to the predicate $Opens(?, Safe_1)$ and be interpreted as a question.

In some cases, such as the question "Is object A in location B?", no specific predicate information is given. However, if the agent can find a predicate which accepts all the detected entities, it can be inferred from the utterance. Table 1 provides several example mappings for NL-Is, NLEs, and constructed beliefs from the museum benchmark (see Section 6).

Once a belief is constructed, the agent queries its knowledge base to determine an appropriate response to the question. For confirmation questions, if the agent finds a belief matching the query, the agent responds in affirmation or negation, i.e. "Yes, Key one opens safe one." For information questions, the agent will issue a response for each candidate found which satisfies the query belief, i.e. "Key one opens safe one. The master key opens safe one." For utterances labeled as answers, if a complete belief is constructed, it is added to the agent's knowledge base.

5.3 Generating Questions and Statements

Questions: Each uncertain predicate in the agent's plan must be resolved in order to satisfy the agent's desires. To generate a question for an uncertain item, the agent first maps the attribute or predicate in question to its matching entry in the lexicon to discover potential template questions for the given item. The production templates are augmented with appropriate slots for entities, predicates, etc. The agent binds the entities from its uncertain predicate to the NLE slots in the production template. If all entities in the question are bound and all NLE slots in the production are complete, the production sentence can be uttered. We maintain several production templates for each predicate and attribute to generate variation in the agents' utterances. The agent may optionally determine to whom the question is addressed. If there is one nearby agent, the agent can specifically address the other agent using an arbitrarily assigned, unique phonetic name.

Statements: Similar to questions, statements are bound by matching the knowledge predicate to a sample production template. Once a question is received, the agent performs a query into the knowledge-base. If an answer is found, i.e. a predicate or attribute belief which satisfies the query, the agent matches the belief into production templates for the attribute or predicate. If no answer is found, the agents are given a set of generic responses representing a lack of knowledge, e.g. "I'm sorry. I don't know."

The agents process and produce natural language utterances as needed to respond to questions or pro-actively seek information. No implicit information is transmitted between agents, which allows the agents to communicate with agents or avatars without distinction between the methods of interaction. The supplemental material provides an example exchange between two agents.

6 Results

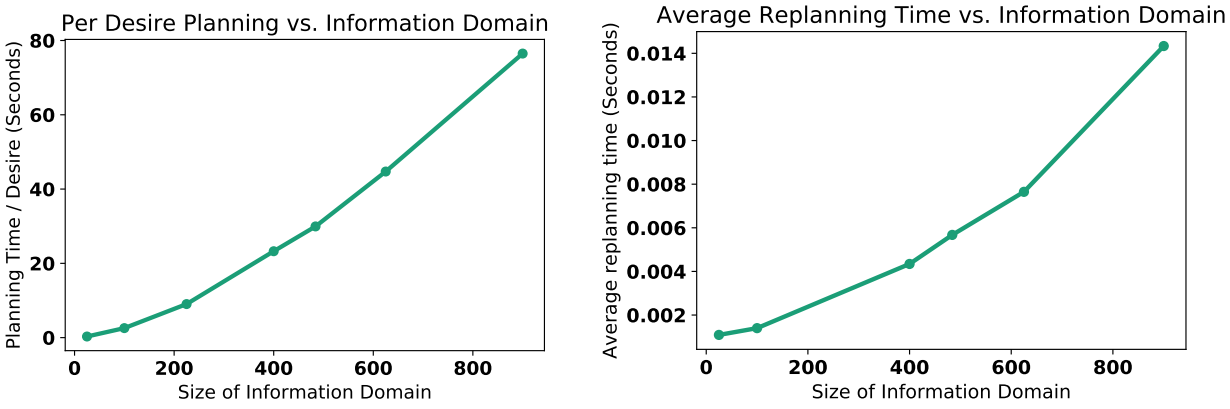
In this section, we highlight the effectiveness of our novel approach in generating interactions among virtual agents and between virtual agents and avatars given uncertain information, as well as the role of verbal communication in resolving uncertainty. For implementation and platform details, we refer the reader to the supplimental material.

6.1 Benchmarks

We demonstrate the results of our approach on several challenging multi-agent scenarios (Figure 3). Table 2 details a performance comparison between our method and a prior crowd simulation approach without natural language communication. Overall we find that our approach decreases the overall solution time with a small increase in replanning times. Specific details on the number of agents and desires in each benchmark can be found in the supplemental material. The simulation benchmarks tested include:

Anti-podal Circle: In this scenario, 10 agents and an equal number of 'goal-objects' are distributed on the circumference of a circle (Figure 3(a)). Each agent is given a desire to retrieve a set of randomly assigned goal-objects. However, the agent's initial belief set may not include information about the desired goal-objects. The agent engages in verbal communication with other agents in order to resolve uncertainties regarding the location of the goal-object. This benchmark illustrates the ability of our algorithm to plan with incomplete information, automatically generate uncertainty resolution actions, and facilitate verbal agent-agent communication. We find that asking questions reduced overall simulation time compared to our method without the ability to resolve uncertainty via interaction.

Evacuation: We simulate a fire evacuation scenario in a Y-shaped corridor (Figure 3 (b)). A group of 10 agents approach a junction and must choose one of two passageways to reach their goals. Each agent randomly selects its desired passageway. Unknown to the agents, the right passageway is obstructed due to a fire breakout. As the agents approach the junction, an agent acting as a first-responder redirects them to the leftmost passageway using verbal communication. Without our algorithm, agents are unable to communicate and



(A) (B)

Figure 4: **Performance Results per Desire and Replanning Call**: **(A) Per desire planning time varying domain size:** We observe that the per-desire planning time scales exponentially as a function of the domain size. This is consistent with proposition approaches. **(B) Average time per replanning call:** Replanning using our two-stage algorithm can be performed on the order of milliseconds even in complex information domains. This reduces the overall planning time substantially.

some take the rightmost passageaway forcing them to retreat. This leads to a 47% increase in evacuation time as is shown in Table 2.

Avatar & Multi-agent Tradeshow: In this scenario, agents explore a complex tradeshow scenario with a large number of booths (Figure 3 (c)). One of the agents is given the desire to go to the 'registration desk' but does not know its location. Using our two-stage planning approach, the agent generates a plan template and a set of candidates for the location of the registration desk. It then verbally interacts with nearby agents to resolve the uncertainty and find the location of the registration desk. We also simulate this scenario with a user in immersive settings (Figure 3 (d)). The user controls a virtual avatar and is given the task of finding the registration desk. The user asks questions, and receives meaningful responses from the agents (Figure 3 (e)).

Avatar and Multi-agent Museum: This scenario demonstrates an art museum. In the multi-agent case, each agent is given a set of statues to visit. However, knowledge of the locations of the statues is randomly assigned amongst the agents. Each agent must seek out other agents with whom they can interact to acquire the location of the statuary they are seeking. In the avatar case, the user is given a specific statue to find in the museum. During the user's exploration, virtual agents approach the user and request the locations of statues the user has previously visited. If the user responds with the appropriate information, the agents are able to complete their plans.

Multi-Avatar Hide-and-Seek: This scenario depicts multiple user avatars engaged in a hide-and-seek game in a virtual environment populated by many virtual agents. The hiding avatar chooses a room in which to hide, and the seeking avatar must interact with the virtual agents to obtain the location of the the hider. The necessary information is disbursed amongst several agents, requiring the seeker to interact with multiple virtual agents to find the hider.

6.2 Performance Analysis

We conducted a series of repeated trials on the Anti-podal circle benchmark described in Section 6.1. In each trial, we increased the number of virtual agents or the number of potential goal-objects for the agents. Overall, we find that our planning algorithm scales linearly in the number of potential targets for a single agent and linearly in the number of agents. Consistent with other propositional planning approaches, we find our algorithm scales exponentially in the size of the information domain. We further evaluated the replanning time for each agent as the agents resolve uncertainty in their plans. We find that our two-stage approach yields negligible replanning times even though agents must resolve new information and choose new candidates. Figure 4 details how our two-stage approach reduces overall computation time be enabling rapid replanning. Additional details can be found in the supplemental material.

In a typical scenario, the generation of candidates is only performed once, and can done as a pre-processing step for the scenario, leading to interactive agents capable of replanning as a response to verbal communication with extremely small overhead.

6.3 User Evaluation

We conducted a user study to evaluate the plausibility of agent-avatar interactions and the overall simulation generated as a result of our algorithm. Prior work establishes a procedure for evaluating new features of interactive agents against agents lacking the new capability [24]. In addition, prior work has demonstrated that implausible behavior from agents can lead to a reduction in the sense of quality and presence in a virtual environment [1, 38]. We therefore sought to establish whether our approach generated improvement in the overall perception of simulations between virtual agents and avatars compared with agents lacking the SPA interaction approach. We provide a summary of our user evaluation in this section and refer the reader to the supplemental document for extended details.

This study was conducted based on a within-subjects, paired-comparison design. Each scenario was displayed with a text-based prompt to provide the appropriate context. Participants were shown two pre-recorded videos of a subject interacting with the system in a side-by-side comparison of our method and one of two comparison methods, one with no virtual agents and one with agents lacking SPA. The study consisted of two trials described in Figure 1. After each trial, participants were asked to answer a short questionnaire before moving on to the next scenario. The order of scenario and the positioning of the methods was counterbalanced.

Participants indicated responses on a Likert scale from 1 to 7. Participants were asked to indicate which simulation more closely reflected a real-world scenario, and were asked the impact of the following items on their preference: the presence of natural language communication, the quality of the verbal responses from the agents, and the quality of the animation. The study was taken by 14 participants. For results reporting, we normalize participant responses such that 1 indicates a strong preference for our method for comparative questions and 7 indicates a strong positive impact for absolute questions. We additionally collapsed participant responses across trials.

Analysis: We found statistical significance on all metrics. For

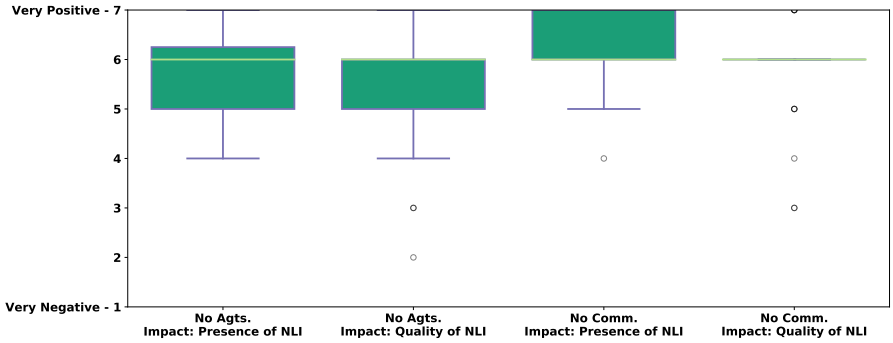


Figure 5: **Participant Impact Perception:** This figure shows the perceived impact on participant preference for our method over several factors. Overall, we observe that the presence and quality of the natural language interactions produced by SPA were a significant factor in participant preferences. **No Agts.**: Compared against the no agents condition, participants perceived the presence and quality of natural language interactions (NLI) provided significant positive impacts on their preference (5.82 ± 0.94 and 5.29 ± 1.24). **No Comms.**: Compared against the no communication agents condition, the presence and quality of NLI was an even more important factor in the participants preference (6.18 ± 0.77 and 5.75 ± 1.00). This is consistent with expectations, as the presence of the virtual agents is consistent across both examples and is less impactful.

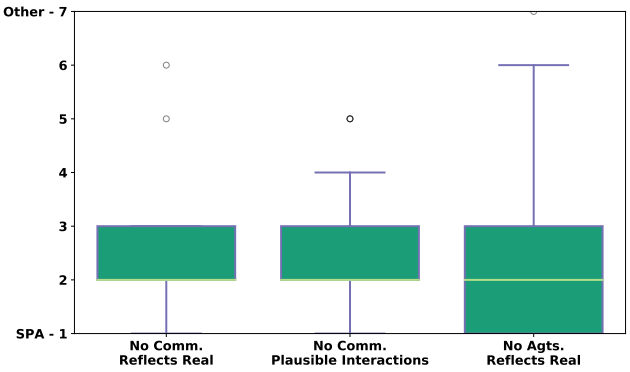


Figure 6: **Participant Preference in User Evaluation** This figure details participant preference scores for SPA compared to prior methods. The questions represented in this graph were "Which simulation more closely reflects a real-world scenario" and "In which simulation did the interactions between the user and the agents seem more plausible". The scores are normalized such that 1 indicates a strong preference for SPA and 7 indicates a strong preference for the other method. **No Agts.**: Compared to the no agents condition, participants felt SPA generated more realistic simulations (2.29 ± 1.15). **No Comm.**: Compared to agents without natural-language interaction, participants felt SPA generated more realistic simulations as well (2.29 ± 0.81). In addition, participants felt the natural-language interactions increased the overall plausibility of the agent-avatar interactions compared to agents without the ability to interact using natural-language interaction (2.46 ± 1.20).

our analysis methods, we refer the reader to the supplemental material. The participant responses clearly demonstrate the benefits of our algorithm in terms of generating plausible agent-avatar interactions (2.46 ± 1.20). Expanding the interaction capacity of the virtual agents beyond movement interactions has a positive impact, and participants believed the quality of the natural-language interaction generated by SPA was a positive factor when comparing against agents without the ability to interact (mean impact:5.75 ± 1.00) or cases with no agents (mean impact: 5.29 ± 1.24). In both comparative conditions, either without agents or without communication, participants found our method to generate simulations and interactions with better reflect real-world scenarios (2.29 ± 1.15 and 2.29 ± 1.46).

Overall, the results of our study show that participants find agents with our SPA natural-language interaction capability to be significantly more plausible than those without. More importantly, they show that participants found the natural-language interactions generated by SPA to be a significantly positive factor on their preferences. This indicates that SPA yields natural-language interactions which are plausible and effective. Figure 5 and Figure 6 provide further details of our analysis. Our evaluation indicates that further work is merited to evaluate SPA against current proposed methods for multi-agent, multi-avatar interactions and establishes a baseline for comparison.

7 Conclusion

We have presented a novel algorithm for generating virtual agent plans under uncertain conditions and natural language interactions between virtual agents and avatars for multi-agent multi-avatar environments. Our approach allows agents to plan with uncertain information, engage in question and answer-based dialog and effectively accomplish their individual goals while facilitating plausible avatar-agent interactions. We have demonstrated how our approach can be used to provide significant improvements to behavior plausibility for virtual agents in a shared environment and detailed a user-study which provides preliminary verification of our approach's advantage. Moreover, our approach can simulate dozens of interactive virtual agents in real time. Overall, SPA seeks to improve limitations of interactivity in typical multi-agent planning approaches and addresses limitations of single agent-avatar interactions in typical conversation agent approaches.

Our algorithm is part of ongoing research and has some limitations. As with many propositional approaches, our algorithm's performance degrades with the complexity of the agents' desires and the problem domain. This could be addressed by planning as a pre-computation step and caching plan templates for subsequent simulations. We will additionally explore partial-order planning [35] to improve plan computation time.

In addition, while the use of shallow-semantic parsing enables verbal interactions, it is very sensitive to training data, making the communication quality sensitive to lexicon quality. Our agents are able to produce a relatively small set of dialog interactions. In the future, we will seek to expand the interaction capability of our virtual agents to include conversational context keeping and other dialog actions as well. We will also explore how SPA can be integrated with existing context-aware dialogue approaches. We will also work to improve our attention models. Research in human-agent interaction

offers avenues for exploring different attention models [23]. Finally, our evaluation establishes that SPA generates plausible interactions. Future work is needed to evaluate our algorithm against proposed state-of-the-art multi-agent multi-avatar interaction approaches.

8 APPENDIX

A ADDITIONAL METHOD DETAILS

Figure 7 details our two-stage planning pipeline. Figure 8 details our natural-language communication pipeline in an example dialog.

Sample Lexion Entry: The lexicon used to generate the results presented in section 6 of the main document consists of a small subset of English words annotated with sample sentences and reference hints for the parser. The word "location" as it appears in the lexicon is tagged with the hint "where" and "InSpace" which are other forms seen in our domain descriptions. As a predicate, It is assigned several template sentences with annotated natural-language intentions. One such sentence with the label *predicate answer* is

"the [PREDICATE:NAME] of [PREDICATE-ENTITY:DEF-ARTICLE-NAME] is [PREDICATE-ENTITY:DEF-ARTICLE-NAME:GALLERY]. "

This template provides parameters for binding an entity with a definite article, e.g. the statue, to an entity with the type specifier gallery. A sample binding of the sentence would be

"The location of the Venus de Milo is Gallery B".

B ADDITIONAL PERFORMANCE RESULTS

B.1 Implementation and Performance Benchmarks

Our experiments were conducted on a desktop pc with an Intel Xeon E5 CPU, NVIDIA TitanX GPU and 16gb of RAM. We coupled our propositional planner with Rasa NLU [33] for semantic parsing. User utterances were captured via microphone and automated speech recognition. Our algorithm was implemented in python, and our VR experiments were performed with the Occulus Rift HMD. We couple our approach with the 3D animation system described in [22].

In addition to the results reported in section 6 of the main document, we evaluated the algorithm's performance as a function of domain size and number of agents. Consistent with prior propositional planning approaches, our algorithm scales linearly in the number of agents and exponentially in the size of the problem domain. Figure 9 details our experimental results. Table 3 provides additional details about the number of agents, desires, and verbal interactions in our benchmarks.

B.2 User Evaluation

This section provides a formal description of the user study we conducted to evaluate the plausibility of agent-avatar interactions and the overall simulation generated as a result of our algorithm. In addition, we provide the complete set of participant responses and additional response analysis.

Experiment Goals & Expectations: We hypothesize that verbal communication between agents and avatars will enhance the perceived plausibility of the simulation, and generate positive impressions as compared to the control conditions.

B.2.1 Comparison Conditions

- **No Agents**: In the no agents case, a user avatar explores a virtual environment without any virtual agents present.
- **No Communication**: In the no communication case, a user avatar explores a virtual environment with agents who could not interact using natural-language communication.

B.2.2 Experimental Design

This study was conducted based on a within-subjects, paired-comparison design. Each scenario was displayed with a text-based prompt to provide the appropriate context. Participants were shown two pre-recorded videos of a subject interacting with the system in a side-by-side comparison of our method and one of the comparison methods. They were then asked to answer a short questionnaire before moving on to the next scenario. The order of scenario and the positioning of the methods was counterbalanced.

B.2.3 Environments

The multi-agent tradeshow scenario and multi-agent museum were used in this study. Three confederates were recruited to participate as the avatar in the environments. In trials using our method, the confederate was allowed to interact with the agents using natural-language communication. In each case, the avatar was piloted from a first-person view. Their interactions were recorded via screen capture and a microphone.

Tradeshow: The avatar was instructed to find the "registration booth". They were shown a picture of the booth before beginning their task but were not told its location. In the SPA case, virtual agents in the environment were able to interact and provide the location of the booth to the avatar. We refer the reader to the main document for visual examples of the benchmarks.

Museum: The avatar was instructed to find a specific statue in the museum but was not told the location of the statue. The statue in question was Lucy, courtesy of the Stanford University Computer Graphics Laboratory. In the SPA case, a virtual agent near the avatar's starting position was provided knowledge of the location. The avatar was able to ask this agent the location of the statue. In addition, two agents were placed along the path to the goal who would interrupt the avatar's progress and ask the avatar for the locations of other statues as they passed.

B.2.4 Metrics

Participants were asked a set of common questions for both comparison methods, with specific additions for each comparison method.

Common Metrics: Participants were asked to indicate which simulation more closely reflected a real-world scenario on a Likert scale with 1 indicating strong preference for the method presented on the left, 7 indicating strong preference for the method presented on the right, and 4 indicating no preference. They were then asked the impact of the following items on their preference: the presence of natural language communication, the quality of the verbal responses from the agents, and the quality of the animation. These were answered on a Likert scale with 1 indicating strong negative impact, 7 indicating strong positive impact, and 4 indicating no impact.

No Agent Metrics: Participants were additionally asked what impact the presence of the virtual agents had on their preference.

No Communication Metrics: Participants were additionally asked which of the methods demonstrated more plausible interactions, in which simulation did the agents appear to benefit more from their interactions with the avatar, and in which simulation did the avatar appear to benefit more from their interactions with the virtual agents.

B.2.5 Results

The study was taken by 14 participants. We normalized the data for comparative questions such that a response of 1 indicates strong preference for our method. We collapsed the common metrics across trials as well as plausibility of interactions question for the No Communication metric and the presence of virtual agents from the No Agents metric. We performed a one-sample t-test comparing the mean of each question with a hypothetical mean of 4 (no preference or no impact). We limit our discussion below to questions which directly deal with natural-language interaction and preference for the

Figure 7: **Two-stage Action Planning with Incomplete Information**: Each agent is given a desire to achieve during simulation. We propose a two-stage planner which generates action plans despite uncertainties in the agent's knowledge. The first stage generates a plan template and a set of candidates for each argument in the plan. The second stage generates a set of candidate bindings. The algorithm selects the plan with the least uncertainty and generates an action plan from the bindings. If any uncertain information is present, an uncertainty resolution action is created, yielding the final action plan which may include asking questions, or exploring the environment.

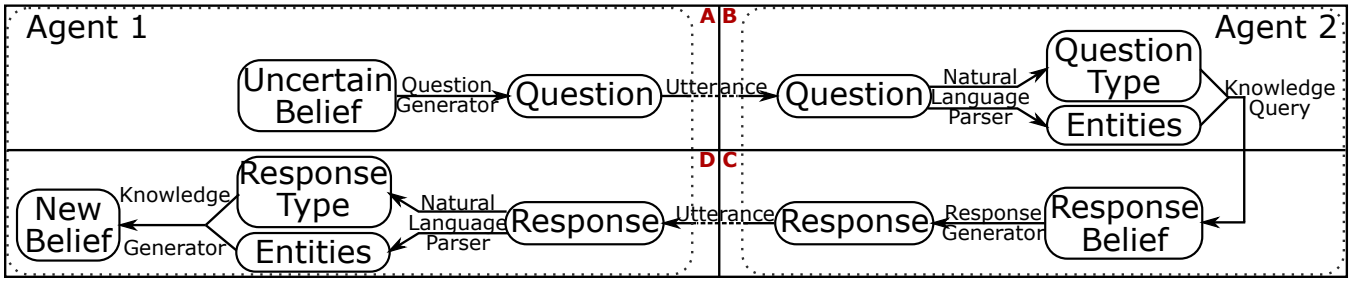


Figure 8: **Natural Language Communication for Virtual Agents**: This figure illustrates a sample interaction between two agents using our natural-language approach (clockwise from top). **(A)** Agent 1's plan yields an uncertain belief. The agent generates a question from the belief template. The question is communicated as a natural language utterance. **(B)** Agent 2 receives the utterance and parses it into the relevant question type and entities. The agent queries its knowledge-base for an answer to the question, yielding a response predicate. **(C)** Agent 2 uses our approach to generate a response utterance. **(D)** Agent 1 receives the utterance, generates the appropriate response type and entities, and processes these into a new belief which is stored in the knowledge-base.

methods. Table 4 gives complete details of the participant responses collected for our user evaluation.

We found the question "Which simulation more closely reflects a real-world scenario" significant in both the no agents condition ($t(27) = -6.204, p < 0.000$), and the no communication condition ($t(27) = -7.887, p < 0.000$). We found the question "What impact did the presence of natural language interaction have on your answer" significant in both the no agents condition ($t(27) = 10.200, p < 0.000$), and the no communication condition ($t(27) = 14.925, p < 0.000$). We found the question "What impact did the quality of the verbal responses from the agents have on your answer" significant in both the no agents condition ($t(27) = 5.473, p < 0.000$), and the no communication condition ($t(27) = 9.218, p < 0.000$). We found the question "In which simulation did the interactions between the user and the agents seem more plausible" significant in the no communication condition ($t(27) = -6.765, p < 0.000$). It was not asked of the no agents condition. We found the question "What impact did the presence of virtual agents have on your answer" significant in the no agents condition ($t(27) = 13.478, p < 0.000$). It was not asked of the no communication condition. perception of the impact of natural language interactions.

Analysis: As detailed in section 6 of the main document, participant responses demonstrate the benefits of our algorithm in terms of generating plausible agent-avatar interactions (2.46 ± 1.20). In both comparative conditions, either without agents or without communication, participants found our method to generate simulations and interactions with better reflect real-world scenarios (2.29 ± 1.15 and 2.29 ± 1.46).

Overall, Participants preferred our approach in 84% of responses. Of those responses, 84% were strong preferences ($r \leq 2$). Figure 10 and Figure 11 provide additional details about our method's advantages over prior approaches.

Acknowledgments

This work was supported in part by ARO Grants W911NF1910069 and W911NF1910315 and Intel.

References

[1] J. N. Bailenson, K. Swinth, C. Hoyt, S. Persky, A. Dimov, and J. Blascovich. The independent and interactive effects of embodied-agent appearance and behavior on self-report, cognitive, and behavioral markers of copresence in immersive virtual environments. *Presence: Teleoper. Virtual Environ.*, 14(4):379–393, Aug. 2005. doi: 10.1162/105474605774785235

[2] Y. Bisk, D. Yuret, and D. Marcu. Natural language communication with robots. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 751–761, 2016.

[3] M. Brenner and I. Kruijff-Korbayová. A continual multiagent planning approach to situated dialogue. *Proceedings of the Semantics and Pragmatics of Dialogue (LONDIAL)*, p. 61, 2008.

[4] M. Brenner and B. Nebel. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, 19(3):297–331, 2009.

[5] J. Cassell, T. Bickmore, L. Campbell, H. Vilhjálmsson, and H. Yan. Conversation as a system framework: Designing embodied conversational agents. *Embodied conversational agents*, pp. 29–63, 2000.

[6] W.-M. Chao and T.-Y. Li. Simulation of social behaviors in virtual crowd. In *Computer Animation and Social Agents*, 2010.

[7] S. Curtis, A. Best, and D. Manocha. Menge: A modular framework for simulating crowd movement. *Collective Dynamics*, 1(0):1–40, 2016. doi: 10.17815/CD.2016.1

[8] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[9] M. Eger and C. Martens. Keeping the story straight: A comparison of commitment strategies for a social deduction game. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[10] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[11] M. Fox and D. Long. Pddl2. 1: An extension to pddl for expressing temporal planning domains. *Journal of artificial intelligence research*, 2003.

Table 3: **Performance Benchmark Details. We detail number of agents, desires, and the NL-I details of the benchmark scenarios including how many questions were asked, statements made, facts overheard by agents, and parser failures. We observe, as expected, that as the number of agents and desires increases, the amount of information gained from overhearing nearby agents increases.**

Scene	Agents	Desires	Statements	Questions	Facts Overheard	Parser failures
Anti-podal Circle	10	30	14	5	28	0
Evacuation	11	10	4	0	20	0
Museum	5	9	20	13	3	0
Trade Show	4	1	3	2	0	0

Figure 9: **Performance Results Varying Number of Agents and Problem Size**: **(A) Varying agents on a fixed domain size (100 predicates):** We observe that our algorithmic approach's performance scales linearly in the number of agents. **(B) Varying domain size for a fixed number of agents (10 agents):** We observe that our algorithm's performance scales exponentially in the size of the problem domain. This is consistent with propositional approaches. However, our two stage planning approach enables rapid replanning after the initial planning step, reducing overall planning time.

[12] M. Garau, M. Slater, D.-P. Pertaub, and S. Razzaque. The responses of people to virtual humans in an immersive virtual environment. *Presence: Teleoper. Virtual Environ.*, 14(1):104–116, Feb. 2005. doi: 10.1162/1054746053890242

[13] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: theory and practice*. Elsevier, 2004.

[14] A. C. Graesser, H. Li, and C. Forsyth. Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science*, 23(5):374–380, 2014.

[15] R. W. Hill Jr, J. Gratch, S. Marsella, J. Rickel, W. R. Swartout, and D. R. Traum. Virtual humans in the mission rehearsal exercise system. *Ki*, 17(4):5, 2003.

[16] P. Huang, M. Kapadia, and N. I. Badler. Spread: sound propagation and perception for autonomous agents in dynamic environments. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 135–144. ACM, 2013.

[17] I. Karamouzas, B. Skinner, and S. J. Guy. Universal power law governing pedestrian interactions. *Physical Review Letters*, 113(23):238701, 2014.

[18] K. Kullu, U. Güdükbay, and D. Manocha. Acmics: an agent communication model for interacting crowd simulation. *Autonomous Agents and Multi-Agent Systems*, pp. 1–21, 2017.

[19] A. Lindsay, J. Read, J. F. Ferreira, T. Hayton, J. Porteous, and P. J. Gregory. Framer: Planning models from natural language action descriptions. In *Proc. of International Conference on Automated Planning and Scheduling (ICAPS 2017)*, 2017.

[20] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pp. 55–60, 2014.

[21] C. Martens. Ceptre: a language for modeling generative interactive systems. In *Eleventh artificial intelligence and interactive digital entertainment conference*, 2015.

[22] S. Narang, A. Best, and D. Manocha. Simulating movement interactions between avatars & agents in virtual worlds using human motion constraints. *Proc. of IEEE VR*, 2018.

[23] S. Narang, A. Best, T. Randhavane, A. Shapiro, and D. Manocha. Pedvr: Simulating gaze-based interactions between a real user and virtual crowds. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pp. 91–100, 2016.

[24] C. Nass, K. Isbister, E.-J. Lee, et al. Truth is beauty: Researching embodied conversational agents. *Embodied conversational agents*, pp. 374–402, 2000.

[25] C. Nass, J. Steuer, and E. R. Tauber. Computers are social actors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 72–78. ACM, 1994.

[26] N. Novielli, F. de Rosis, and I. Mazzotta. User attitude towards an embodied conversational agent: Effects of the interaction mode. *Journal of Pragmatics*, 42(9):2385–2397, 2010.

[27] S. Paris and S. Donikian. Activity-driven populace: A cognitive approach to crowd simulation. *Computer Graphics and Applications, IEEE*, 29(4):34–43, 2009.

[28] E. P. Pednault. Adl: Exploring the middle ground between strips and the situation calculus. *Kr*, 89:324–332, 1989.

[29] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler. Crowd simulation incorporating agent psychological models, roles and communication. *First International Workshop on Crowd Simulation*, 2005.

[30] D.-P. Pertaub, M. Slater, and C. Barker. An experiment on public speaking anxiety in response to three different types of virtual audience. *Presence: Teleoperators and virtual environments*, 11(1):68–78, 2002.

[31] R. P. Petrick and F. Bacchus. Extending the knowledge-based approach to planning with incomplete information and sensing. In *ICAPS*, pp. 2–11, 2004.

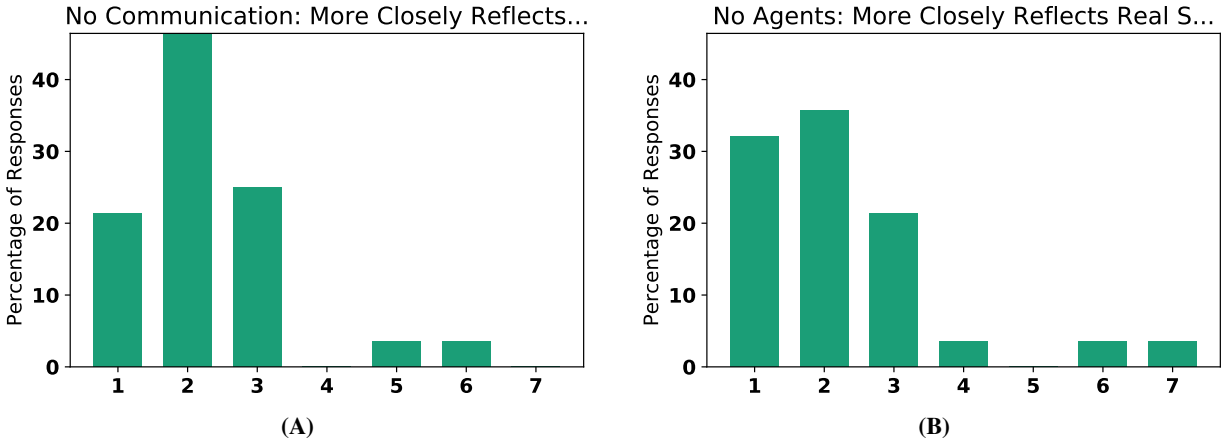


Figure 10: **Histogram data of user responses for Which scenario better reflects real-world scenarios**: Participants in our evaluation found simulations using SPA significantly more plausible compared to simulations with a prior approach **(A)** and simulations with no agents **(B)**. This indicates that the presence of agents has a positive impact on plausibility and that our agents behave sufficiently well to increase plausibility with respect to agents lacking SPA.

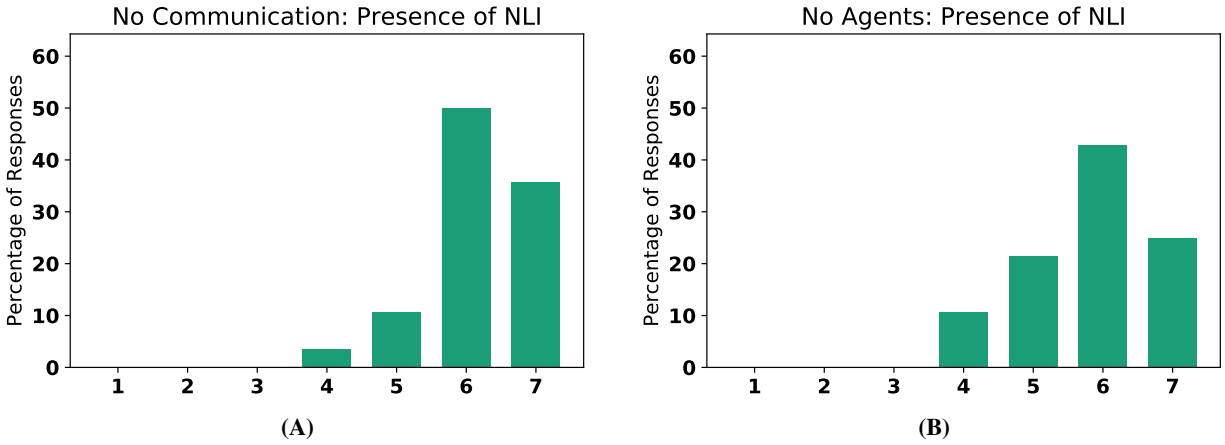


Figure 11: **Histogram data of user responses for impact of natural language interactions** Participants in our evaluation found the presence and quality of the natural language interactions had a significant impact on their preference for our approach to simulations with agents lacking SPA **A** and simulations without agents **b**. In addition, the preference for the quality of the natural language interactions generated with SPA is stronger when compared to agents not able to communicate.

[32] A. S. Rao and M. P. Georgeff. BDI Agents: From Theory to Practice. *Proceedings of the First International Conference on Multiagent Systems*, 95:312–319, 1995. doi: 10.1.1.51.9247

[33] Rasa.ai. Language understanding with rasa nlu, 2017.

[34] J. Rickel and W. L. Johnson. Virtual humans for team training in virtual reality. In *Proceedings of the ninth international conference on artificial intelligence in education*, vol. 578, p. 585, 1999.

[35] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd ed., 2009.

[36] A. Schadschneider. Cellular automaton approach to pedestrian dynamics - theory. *Pedestrian and Evacuation Dynamics*, pp. 75–86, 2002.

[37] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Symposium on Computer Animation*, pp. 19–28, 2005.

[38] M. Slater, B. Lotto, M. M. Arnold, and M. V. Sanchez-Vives. How we experience immersive virtual environments: the concept of presence and its measurement. *Anuario de psicología*, 40(2), 2009.

[39] G. Snook. Simplified 3d movement and pathfinding using navigation meshes. In *Game Programming Gems*, chap. 3, pp. 288–304. Charles River, Hingham, Mass., 2000.

[40] L. Sun, A. Shoulson, P. Huang, N. Nelson, W. Qin, A. Nenkova, and N. I. Badler. Animating synthetic dyadic conversations with variations based on context and agent attributes. *Computer Animation and Virtual Worlds*, 23(1):17–32, 2012.

[41] S. Tellex, R. A. Knepper, A. Li, D. Rus, and N. Roy. Asking for help using inverse semantics. In *Robotics: Science and systems*, vol. 2, 2014.

[42] M. Tenorth and M. Beetz. Representations for robot knowledge in the knowrob framework. *Artificial Intelligence*, 247:151–169, 2017.

[43] J. Thomason, S. Zhang, R. Mooney, and P. Stone. Learning to interpret natural language commands through human-robot dialog. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai):1923–1929, 2015.

[44] B. Ulicny and D. Thalmann. Towards interactive real-time crowd behavior simulation. *Computer Graphics Forum*, 21(4):767–775, 2002.

[45] J. Van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Inter. Symp. on Robotics Research*, pp. 3–19, 2011.

[46] W. G. van Toll, A. F. Cook, and R. Geraerts. Real-time density-based crowd simulation. *Computer Animation and Virtual Worlds*, 23(1):59–69, 2012.

[47] S. I. Wang, P. Liang, and C. D. Manning. Learning Language Games through Interaction. *The 54th Annual Meeting of the Association for Computational Linguistics*, pp. 2368–2378, 2016.

Table 4: **Frequency of Responses in User Evaluation. This table shows the frequency of participant responses in the user evaluation, as well as the means and p-value for a one-sample t-test with a hypothetical mean of 4. For comparative questions, responses less than 4 indicate preference for our agents. For impact questions, responses greater than 4 indicate positive impacts. We found participant responses to all question significant.**

Question	1	2	3	4	5	6	7	mean	std	p-value
NL-I Agents vs Non-Interactive Agents										
Comparative Questions (NL-I Agents left)										
More closely reflects real scenario	6	13	7	0	1	1	0	2.29	$\pm$1.15	< 0.000
Agents benefit more from interaction	11	4	1	11	0	1	0	2.57	$\pm$1.53	< 0.000
User benefits more from interaction	17	10	0	0	0	1	0	1.54	$\pm$1.00	< 0.000
More plausible interactions	5	13	5	2	3	0	0	2.46	$\pm$1.20	< 0.000
Impact Questions										
Presence of natural Language	0	0	0	1	3	14	10	6.18	$\pm$0.77	< 0.000
Quality of the verbal interactions	0	0	2	1	3	18	4	5.75	$\pm$1.00	< 0.000
Animation of the virtual agents	0	0	4	13	3	3	5	4.74	$\pm$1.36	0.010
NL-I Agents vs No Agents										
Comparative Questions (NL-I Agents left)										
More closely reflects real scenario	9	10	6	1	0	1	1	2.29	$\pm$1.46	< 0.000
Impact Questions										
Presence of the virtual agents	0	0	0	0	8	10	10	6.07	$\pm$0.81	< 0.000
Actions of the virtual agents	0	0	0	6	9	10	3	5.36	$\pm$0.95	< 0.000
Presence of natural Language	0	0	0	3	6	12	7	5.82	$\pm$0.94	< 0.000
Quality of the verbal interactions	0	1	2	3	7	12	3	5.29	$\pm$1.24	< 0.000
Animation of the virtual agents	0	0	3	11	10	2	2	4.61	$\pm$1.03	0.004