Towards a generic framework for multi-party dialogue with virtual humans

Raoul Harel
Utrecht University
Utrecht, The Netherlands
admin@rharel.com

Zerrin Yumak
Utrecht University
Utrecht, The Netherlands
z.yumak@uu.nl

Frank Dignum
Utrecht University
Utrecht, The Netherlands
f.p.m.dignum@uu.nl

ABSTRACT

Existing approaches and frameworks for modeling virtual dialogue tend to be designed with dyadic interactions in mind, and are often built to serve solely in task-oriented domains. However, modeling realistic action and turn-taking in more general scenarios remains a challenge. In this paper we propose a generic framework to aid in development of multi-modal, multi-party dialogue. It contains mechanisms inspired by social practice theory for both action selection and timing — including handling of interruption. As a proof-of-concept, we employ these ideas in a virtual couples-therapy session, demonstrating their potential in modeling complex real-life situations.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**;

KEYWORDS

virtual dialogue, social expectations, turn-taking, interruption

ACM Reference Format:
Raoul Harel, Zerrin Yumak, and Frank Dignum. 2018. Towards a generic framework for multi-party dialogue with virtual humans. In *CASA 2018: 31st International Conference on Computer Animation and Social Agents, May 21–23, 2018, Beijing, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3205326.3205327

1 INTRODUCTION

Despite large progress towards realism in video games, realistic multi-party dialogues are inadequate. Currently, dialogue with virtual humans is overwhelmingly dyadic, lacking in mixed-initiatives, and neglects to handle interruption. Initiative in conversations with non-player characters (NPC) is driven entirely by the player. In addition, at most one agent is active at a time, meaning the player is not given an opportunity to interrupt NPCs, nor can NPCs interrupt the player or each other.

In video games, there is justification for why NPCs have little initiative: it is easier to plan a narrative around the choices of a single agent rather than many. Likewise, virtual personal assistants are intended to serve a single user, and therefore a lack of multi-party interaction support is understandable. But a more realistic model of human-human interaction is necessary for the training of medical professionals dealing with tough conversations. Or in education, where a virtual teacher leads a class. In such serious games all three of multi-party interaction, mixed-initiative, and interruption handling are desired. Thus our contribution focuses on the specification and implementation of the following four components:

- A communication management system that coordinates a multi-modal perception/action cycle between agents.
- A baseline model for authoring conversational agency systems. It decomposes an agent's perception/action into several modules, and is designed specifically with multi-party, interruption-enabled interaction in mind.
- A proposed implementation mechanism for two modules of that model: It uses the concept of social 'expectations' as a guide for both action selection and timing.
- A demo application showcasing how our social expectations mechanism, built on top of the agency model and running side by side with the communication manager, is used to simulate a minimal couples-therapy session scenario.

The remainder of this paper is structured as follows: First we discuss previous work, and highlight where our contributions fit in. Next, we elaborate on each contribution in order: beginning with the communication management scheme, followed by the modular agency model, then the social-expectation-based action mechanism, and finally the couples-therapy case study. We end with a review of our research and directions for future work.

2 RELATED WORK

Traum et al. [21] identify numerous challenges that arise when making the transition from dyadic to multi-party interactions. They also specify a dialogue model tackling many aspects of the problem, including support for multi-modality, turn-taking, flow of initiative, and conversational topic management [22]. Allen et al. [1] classify five dialogue management techniques according to their complexity. These methods (and the majority of work done in general) pertains to task-oriented interactions [11]. However, we are specifically interested in systems playing the role of another agent and not that of a tool used to achieve a technical goal. Relevant contributions in this domain include the Virtual Human Toolkit, 'Sensitive Artificial Listeners', and Lopez et al.'s work on agents for collaborative virtual training [7, 13, 18]. Notable dialogue management frameworks are RavenClaw for task-oriented domains, and IrisTK for statechart-based multi-party interaction [4, 19]. Though the most relevant for this paper is Trindikit, which models the relationship between external events and a system's information

CASA 2018, May 21–23, 2018, Beijing, China

ACM ISBN 978-1-4503-6376-1/18/05…$15.00
https://doi.org/10.1145/3205326.3205327

state using so-called update rules [9, 12]. However, Trindikit only supplies this formulation as a basis, and it is rather its users who are left to decide on the information state structure and the controller of update rules. Additionally, we are concerned that update rules alone would not scale to scenarios of higher complexity. We address these two concerns using (1) an agency model (borrowing from Trindikit, but with stricter specification) designed with multi-party scenarios in mind; and (2) a transformation of update rules into a more accessible abstraction: social expectations.

Aside from dialogue management, there is also the issue of turn-taking and interruption, both of which play a significant role in interpersonal communication [5]. We classify current approaches to this problem under three categories: (1) passive, (2) data-driven, and (3) decision-theoretic. A recurring theme in turn-taking models is the desire to avoid overlapping speech, and empirical observation justifies it [20]. A straightforward strategy complementing this is to never interrupt and always yield the floor when interrupted. However, in mixed-initiative settings the reluctance to interrupt makes the artificial party seem unnatural.

Data-driven methods are used in simulation of commonly occurring patterns in conversation. They use available corpora to infer relevant distribution parameters of important conversational features. For example, in [10] the ICSI meeting corpus is used to train a probabilistic model capable of generating speech/non-speech patterns. A primary drawback is the dependency on quality of training datasets, which often focus on niche scenario types and may be insufficiently annotated.

In a decision-theoretic formulation, agents have the option to either bid for the floor themselves or surrender it to others. The model developed by Sacks, Schegloff, and Jefferson (SSJ) [17] is an early example of a rule-based solution to this problem, and is still used as a basis for new work [8]. However, it leaves a couple of issues unresolved [14]: Firstly, it assumes turns are discrete, and that interruptions or overlapping speech are negligible. As seen in [2], it is found that the observed degree of overlapping speech in Spanish does not agree with this assumption. Secondly, extending the model to multi-party settings is not quite straightforward, as it does not specify what should be done in a state of simultaneous bids by multiple agents. Another class of methods for making the bid-or-not decision revolves around utility, which is a choice's contribution towards the agent's goals. Utility cost-functions may be heuristics often tailored for specific conversational scenarios [3]. Or, more complex ones may incorporate the mental state of an agent, its beliefs about other agents, and even a capability to predict future effects of a given choice on the conversation's state [16].

A common shortcoming of all prevalent turn-taking methods is the implicit assumption that a turn cannot be broken off. However, in dynamic applications such as serious-games, this issue must be addressed. If it is not, then it follows that once an agent takes a turn, it completes it with no regard to new events that transpire during it. This seems particularly unrealistic in multi-party settings. We present a framework that rectifies this and treats turns as having duration, and also propose one approach towards the handling of interruption using it.

3 A GENERIC FRAMEWORK FOR MULTI-PARTY DIALOGUE

We propose a generic framework for dialogue systems, designed specifically with multi-party, interruption-enabled applications in mind. It consists of three components: (1) A communication management system, (2) a modular agency model, and (3) a method for action selection and timing based on the concept of 'social expectations'.

Agents in the scene undergo a classic update cycle consisting of three elementary steps: perception, deliberation, and action [15]. The communication management system (CMS) is responsible for coordinating this cycle, so that events are perceived consistently across the population. It collects the actions produced by agents in one iteration, and makes them available for perception in the next. To accommodate for multi-modality, the actions collected are organized through channels, with each channel carrying actions belonging to a single modality.

The cognitive model governing an agent is broken down into modules, six in total: (1) recent activity perception, (2) current activity perception, (3) state update, (4) action selection, (5) action timing, and (6) action realization. Figure 1 gives an overview of the different modules and their interaction. The recent activity perception module is responsible for interpreting data off the CMS as dialogue moves, which represent the meanings carried by agent behavior. The current activity perception module also analyzes data off the CMS, and uses it to classify agents as either passive or active. An agent is considered active when it is in the process of realizing any dialogue move but the special 'idle' one, and is said to be passive otherwise. The state update module takes the output of both perception modules and uses it to update the agent's internal state. Using this new state, the action selection module chooses a target move to perform. To decide whether now is indeed an appropriate moment to perform it, the action timing module is queried, which in return emits on of two signals: either 'Stop' or 'Go'. 'Go' indicates that realization of the target move may be initiated (or resumed if already in progress), 'Stop' indicates that realization may not commence (or canceled if already in progress) and should be retried later. Finally, the action realization module converts whatever move has been decided upon (either the target move or the idle move) to concrete action data, to be submitted onto the CMS in preparation for the next iteration.

Modules are defined by their input/output relations, but our agency model leaves their internal implementation abstract in order to remain flexible. However, in the next section we do propose a concrete mechanism by which state update, action selection and -timing could be implemented.

3.1 Describing social practices through expectations

Our proposed approach is inspired by social practice theory, which aims to articulate the symbiotic relationship between the actions of social beings and the systematic rules (be they explicit or implicit) that govern their societies [23]. In conversation, social practices dictate a great deal of our conduct: When greeted by an extension of the arm, we are expected to shake hands; During an encounter with authority we rather not speak unless spoken to first, while

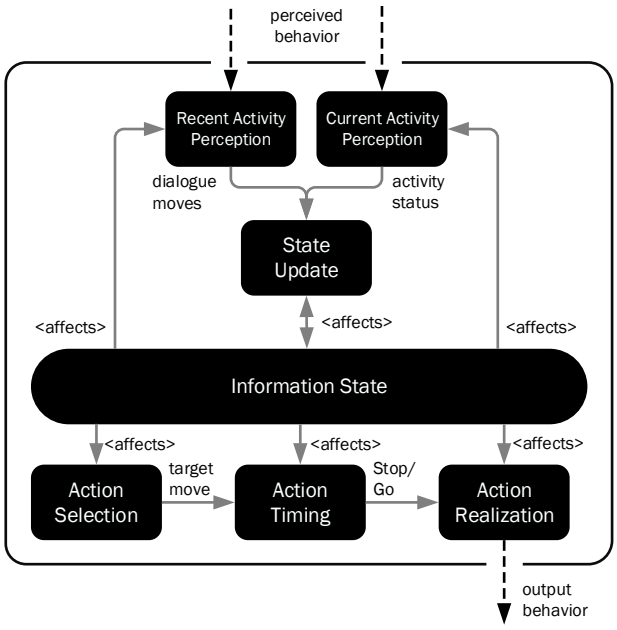

Figure 1: Overview of an agent's cognitive process.

during a friendly group chat interruptions and overlapping speech are much more likely to happen. Indeed social practices, like norms and values, are shared and lead to expectations of certain behavior in a context. Therefore, viewing dialogue from the perspective of social practice is a promising new angle for a solution to the problem of action selection and timing [6].

Incorporation of social practices in the development of agency systems brings with it some very appealing benefits. The first of which pertains to ease-of-use: For us, reasoning about practices is naturally intuitive, because they are concepts of which we are very much aware and make use of in everyday life. This fact alone already opens the door towards development of tools and methods that would allow anyone to take part in agency system modeling. A second benefit of social practices is that they support heterogeneous agents, since they specify an interaction and not the internal deliberation of agents themselves. Finally, the hierarchical nature of social practices easily yields itself to component reuse. Here, the 'hierarchy' refers to the fact that practices of higher complexity can often be expressed as a composition of practices of lower complexity. This allows us to specify basic practices only once and from then on have them be shared.

In the context of conversation, we use the term *practice* to denote a socially-mandated pattern of interaction. For example: agents mutually greeting each other at the onset of a dialogue is one of the most basic instances of such a pattern. And even though practices may vary in complexity, context, and frequency of occurrence, their common denominator is their assignment of various *expectations* at different times to either individual agents or groups. Each expectation falls under one of two kinds: either *atomic* or *composite*. Atomic expectations (a, m) predict the realization of concrete dialogue moves (m) by specific agents (a). We refer to the agent performing a move as its *source*, and to the pairing of a dialogue move with its source as an *event*. For this reason, we use the terms 'event/atomic expectation' interchangeably for the remainder of this paper. In contrast with atomics, composites serve as containers for other expectations. We use the term *parent* to refer to the composite itself and *children* to refer to the collection of expectations it contains. Composites are used to indicate some temporal relationship amongst their children, and/or to impose logical constraint(s) upon them. We define six types of composite expectations: *sequential*, *conjunctive*, *disjunctive*, *divergent*, *repeating*, and *conditional*. Expectations can be viewed as specialized behavior trees with increased emphasis on the interaction as a whole rather than a single agent, and with the goal of arriving at contextual guidelines rather than concrete action.

Using expectations and events, an instance of a practice can be formally represented by a three-tuple $\rho = (A, M, X)$, where A is the set of participating agents, M the set of possible dialogue moves, and X being a (composite) expectation that serves as the root ancestor of all others, and whose descendant atomics all exclusively refer to events in $A \times M$. The size of M depends on the desired abstraction level of actions in a particular scene, for example: answering a yes/no question may be represented as a single move 'Y/N', or split among two {Yes, No}. Regardless, expectations aid in identifying the subset of M that is acceptable in a given moment.

Before we move on to discuss specific types of expectation in detail, we want to first be able to relate a practice's expectation X to the dynamic progression of an ongoing interaction. To do this, we augment expectations of X with two kinds of state-annotation: *relevance* and *resolution*. Relevance refers to the applicability of a given expectation to a given time. Remember that composite expectations express a temporal relation between their children. This means that throughout the course of an interaction different expectations may be applicable at different times. We label those expectations that are currently applicable as *relevant/active* and those that are not as *irrelevant/inactive*. Accompanying the annotation for relevance is another for resolution. Recall that expectations are merely a human-friendly way for representing patterns of interaction. In this context, we view interaction as a sequence of events $I = (e_1, e_2, \ldots)$ that take place while the expectation is relevant. The expectation itself can be regarded as an implicit definition of a set $E = \{I_1, I_2, \ldots\}$ whose members include all possible interactions that adhere to the constraints imposed by that particular expectation. At the onset of interaction $I = \emptyset$, and proceeds to grow as new events take place. The expectation's resolution status then answers the following question: Will it ever be the case that $I \in E$? If it is already so, we resolve the expectation with *satisfaction*. On the other hand, if no member of E is prefixed by I we resolve the expectation with *failure*. We label expectations that are either satisfied or failed as *resolved*, and those that are neither as *unresolved/pending*. Note that a failure of expectations does not mean a failure of the interaction! It means that some agents deviated from the expected pattern of the social practice. Individual agents can use this fact to adjust their behavior using private deliberations. We now go over individual expectation types in detail, and for composites also specify how the relevance and resolution status of the parent relates to that of its children.

Indefinite event expectations (a kind of atomic) represent a pattern of interaction containing a concrete event e, and serve as the basic building block for more complex composites. This type of expectation cannot resolve as a failure, but rather remains pending until eventually becoming satisfied once e takes place.

A sequential expectation p represents a pattern of interaction wherein a sequence of $n \geq 2$ children $(x_1, x_2, \ldots, x_n)$ is satisfied one by one in the specified order. Consequently, at most a single child is active at a time. Let us identify this child by its index i in the sequence. At the onset of interaction $i = 1$, and it remains that way for as long as x_i is pending resolution. Once x_i is resolved, one of two scenarios unfolds: If x_i failed, then p fails as well. Otherwise, i is incremented by one unless $i = n$, in which case we have satisfied the entire sequence and therefore also p itself. Sequential expectations are useful for segmenting an interaction into chronological blocks, as seen in Algorithm 1.

A conjunctive expectation p represents a pattern of interaction wherein $n \geq 2$ children are satisfied in any order. That is to say: all are activated at the onset of interaction, and remain so until resolved. If any child fails, then p fails, and only once all children have been satisfied will p be satisfied. Conjunctions are used when all that matters is a postcondition, e.g. during greetings it is that they are exchanged, in whatever order. In contrast to conjunctions, disjunctive expectations require at least one child to be satisfied for the parent to be satisfied. Disjunctions come in handy as triggers for context transitions, for example: in Algorithm 2 a disjunction enables the transition away from an interaction's main phase and onto its conclusion.

A repeating expectation represents a pattern of interaction wherein a single child x is satisfied zero or more times. Consequently, this type of expectation cannot be satisfied, but rather remains perpetually pending unless there comes a time where x fails. Repeating expectations can be combined with sequences and disjunctions to set bounds on the number of repetitions expected.

A conditional expectation p represents a pattern of interaction wherein a single child x is satisfied, but is only activated while a specified predicate C holds. In other words, p remains pending while $\neg C$, and is assigned the same resolution status as x otherwise.

A divergent expectation represents a pattern of interaction wherein exactly one out of a set of $n \geq 2$ children $\{x_1, x_2, \ldots, x_n\}$ is selected to be satisfied. At the onset of interaction, a divergence p behaves in an identical manner to that of a disjunction, and continues to do so until x_i is partially satisfied for some $i \in [1, n]$ (a partially satisfied expectation is one with a satisfied descendant). At that point, all children are deactivated except for x_i. From then on, p's resolution status is assigned the same value as that of x_i. Divergences are needed to portray a branching of an interaction into mutually exclusive parts, for example: if an agent's response to a yes/no question is expected, and each answer entails the activation of different follow-up expectations.

3.2 Action selection and timing

We utilize the expectation mechanism to arrive at a concrete implementation for the action selection and timing modules of our agency model. When deriving action from a practice $\rho = (A, M, X)$, it is always with respect to one agent $a \in A$. As an ongoing interaction progresses, expectations in X undergo changes of state including both their relevance and resolution. By going through X and looking at active nodes, we collect a set of events E that ρ implies are currently expected. We call E the *expected event set*. We filter from E those events which are relevant for an agent a, yielding: $E_a = \{(a, m) \mid (a, m) \in E\}$. From here, we define the *candidate dialogue move set* for a:

$$M_a = \begin{cases} \{m \mid (a, m) \in E_a\}, \text{ if } E_a \neq \varnothing \\ \{m_0\}, \text{ otherwise } (m_0 \text{ denotes the idle move}) \end{cases}$$

Note that some candidate moves might not be possible to execute e.g. if a move realization requires an object that is not currently in the agent's possession. The agent a can now pick one target move to perform out of the candidate set. When $|M_a| = 1$ this is trivial. When $|M_a| \geq 2$ the agent can use other knowledge and experience to determine the best choice. Notice that M_a is a small subset of the complete set of available moves! Thus the deliberation becomes much more efficient and focused.

Aside from using practice descriptions to select a target move m_t to perform, we also use them to decide on an appropriate time to execute it. When $m_t = m_0$ the problem is trivial, because we regard the idle move to be universally appropriate at all times. When $m_t \neq m_0$ the only remaining source of potential objection to its realization would lie in the turn-taking aspect of conversation. Define the *dialogue floor* to be the set $F = \{a \mid a \in A,\ a \text{ is active}\}$ containing all agents who are currently in the process of realizing non-idle moves. If $F = \varnothing$ or {Self}, then m_t can be performed directly. In the case where other agents beside Self are active and Self has a pending move to perform we have two possibilities: when Self $\notin F$ we decide on a potential *initiation of interruption*; when Self $\in F$ we decide on a potential *surrender to interruption*. We use the expectation X of the current practice to evaluate both decision scenarios.

Recall that X is a tree of expectations. With each node $x \in X$ we associate two sets of rules $R_i(x)$ and $R_s(x)$: the first concerning interruption initiation and the second for surrender. Each rule is a four-tuple made up of the following: (1) a precondition indicating whether the rule is currently applicable, (2) an indicator function signaling which agents of A the rule affects, (3) an implication value in $\{true,\ false\}$, and (4) a weight $w > 0$. To decide on the interruptions, we visit active nodes in X and consult relevant rules from each. A rule is considered relevant when both its precondition holds and Self is affected by it. Each rule casts a weighted vote in favor of its implication. Once all rules have cast their votes, the implication with the greater tally decides the answer. For potential interruption initiation, *true* implies the go-ahead to interrupt and *false* instructs to remain idle. For potential surrender, *true* implies Self should abort whatever move is being realized and *false* instructs to ignore other active agents and press on.

4 CASE STUDY

We implemented the communication management system, the modular agency model, and the expectation-based controller for action selection/timing to employ them in a proof-of-concept application. The scenario being simulated is a short couples-therapy session

with three agents: one therapist and a patient couple. Our choice was motivated by the following attributes of this setting: (1) It contains the minimal number of participants required for an interaction to be multi-party, (2) its agents are clearly heterogeneous, and (3) it involves an interaction whose flow follows a clear agenda — as it is in most formal conversations — and therefore it is more easily expressed using social practices. In our implementation, the patient couple's behavior is driven artificially, while the therapist is controlled by a human player in order to illustrate a potential application domain: a professional in training.

Here the CMS carries two channels: one for speech and one for dialogue moves. Any action output by an agent is submitted to the CMS, which synchronizes its broadcast to all other participants. Agents act according to the modular agency model we specified earlier. Implementing recent activity perception is easy, since recent moves are submitted to the CMS directly. For current activity perception, we classify agents as active if and only if they have output some speech in the last t seconds, where t is predetermined to approximate the maximal pause duration between consecutive utterances in natural language. Using output of the perception stage, we process the set of recent moves to determine their effect on the relevance/resolution status of expectations in our practice description. The updated expectations yield a set M_a of candidate moves. For artificial agents — in our case, the patients — we choose one member of M_a at random to perform. For the therapist, we present M_a to the player and let him/her make the choice. To emphasize the non-discrete nature of speech and to allow for interruption, an agent realizing a selected move m with associated speech text $T(m)$ outputs $T(m)$ to the CMS one word at a time. This gives opportunity to interrupt mid-sentence. Only once the entirety of $T(m)$ has been output, is m itself posted to the dialogue move channel. This simulates the fact that a move can only be recognized once it has been realized fully.

The practice $\rho = (A, M, X)$ governing the scenario is as follows: A = {Alice (PatientA), Bob (PatientB), Charles (Therapist)}, and M is as listed in Table 1. The root node of X, Session, is a sequential expectation segmenting the entire interaction into three parts: Greetings, Counseling, and Goodbyes (Algorithm 1). Greetings and Goodbyes are both conjunctive expectations, each detailing an exchange of respectively a Greeting/Goodbye between each patient and the therapist. We use a conjunction here to signal that it is not important in what order the moves are exchanged, as long they all do.

Algorithm 1 The sequential expectation at the root of a couples-therapy session scenario.

```
1: function Session
2:     expect sequence
3:         Greetings
4:         Counseling
5:         Goodbyes
6:     end sequence
7: end function
```

The Counseling expectation is a bit more complex. First, the therapist invites one of the patients to share an issue he/she is having with their partner. Then, the patient either accepts and elaborates, or declines (this chain of events is encapsulated within the Discussion cycle). This process repeats until either all patients decline any further discussion or the therapist decides it is time to end the counseling session. Algorithm 2 shows how a disjunction between the SessionClosing event and the Discussion cycle signals that the therapist may end the session at any time, and how a divergence is used to ensure exactly one issue is being discussed at a given moment.

Algorithm 2 The counseling phase.

```
1: function Counseling
2:     expect any
3:         expect repeat
4:             expect one of
5:                 Discussion(PatientA)
6:                 Discussion(PatientB)
7:             end divergence
8:         end repeat
9:         expect Therapist: SessionClosing
10:    end disjunction
11: end function
```

In addition to the therapist closing the session, we still need to model the case where neither patient wishes to discuss issues any further. To do this, we wrap the contents of any Discussion with a conditional expectation referencing a special flag: open(p), indicating that a patient p is open to discussion. Initially, open(p) = $true$ for all patients, and is only set to $false$ when p declines an invitation to discuss an issue.

We also have sets of interruption rules, each associated with one or more expectations. These rules affect the artificially-driven agents in the scene exclusively, while the player retains his/her freedom to perform moves at any point in time. The rules governing Greetings and Goodbyes represent indifference to conflict, that is to say agents perform their target moves immediately and to completion. On the other hand, rules governing the Counseling phase represent conflict-avoiding behavior, where agents avoid interrupting others and surrender the floor when they are interrupted themselves. In effect, this rule assignment allows for overlapping speech during the socially-lax parts of the interaction:

```
Alice:   Hello [ Charles. ]
Bob:           [ Hello    ] Charles.
Charles: Hello Alice.
```

...but interruption surrender of patients in favor of the therapist during actual counseling:

```
Charles: Would you like to share an issue
         with us, Alice?
Alice:   Alright.
Alice:   I love my partner, but sometimes-
Charles: Well, I'm afraid our time is up.
Alice:   Alright.
Bob:     Alright.
```

5 CONCLUSION

In this paper we propose a framework to aid in the authoring of multi-party, mixed-initiative, interruption-enabled virtual dialogue:

Move Type	Details	Performer
Acknowledgement	—	Any
Greeting	—	Any
Goodbye	—	Any
SessionClosing	Declaring the end of the counseling session.	Therapist
IssueSharingInvitation	Invitation to share an issue with one's partner.	Therapist
AdviceDispensation	Dispensation of advice in response to a shared issue.	Therapist
IssueSharing	Elaboration on an issue with one's partner.	Patient
IssueSharingDeclination	Declination to an issue-sharing invitation.	Patient

Table 1: A listing of the available dialogue moves for the scenario.

(1) A system which facilitates coordination of perception/action for agents using multiple modalities, (2) a modular model for agency, and (3) a dialogue system built on top of the aforementioned model, and which utilizes social expectations as a means to drive action selection and timing. We combine all three tools in a proof-of-concept application simulating a virtual couples-therapy session, and thereby showcase the ability of social expectations to derive complex interaction dynamics from a relatively small blueprint. We utilize expectations to paint a rough sketch of what a socially-acceptable interaction looks like, and even though agents should aim to follow it, the result is but a set of candidate actions which under exceptional circumstances may be overruled by private deliberations. In addition, the patterns implied by expectation arrangements are not necessarily deterministic, which allows us to package many interactions in one practice description. Future work should supplement practices with new expectation types, investigate how scalable the expectation hierarchy is for larger scenarios, and confirm its believability through empirical evaluation. Finally, social practices alone can only bring us so far, but the most reasonable approach to the final action decider should be goal-oriented [6]. A promising approach would be the development of a reasoning-engine capable of considering each potential action, looking ahead to predict its effect on the expectation arrangement, and selecting the one that most closely aligns with the agent's personal desires.

ACKNOWLEDGMENTS

This work is partly supported by the Horizon 2020 RAGE (Realizing an Applied Gaming Ecosystem) project (Grant no. 644187).

REFERENCES

[1] James F Allen, Donna K Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001. Toward conversational human-computer interaction. *AI magazine* 22, 4 (2001), 27.

[2] Anne Berry. 1994. Spanish and American Turn-Taking Styles: A Comparative Study. (1994).

[3] Dan Bohus and Eric Horvitz. 2011. Decisions about turns in multiparty conversation: from perception to action. In *Proceedings of the 13th international conference on multimodal interfaces*. ACM, 153–160.

[4] Dan Bohus and Alexander I Rudnicky. 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. (2003).

[5] Angelo Cafaro, Nadine Glas, and Catherine Pelachaud. 2016. The effects of interrupting behavior on interpersonal attitude and engagement in dyadic interactions. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 911–920.

[6] Virginia Dignum and Frank Dignum. 2014. Contextualized Planning Using Social Practices. In *Coordination, Organizations, Institutions, and Norms in Agent Systems X*. Springer, 36–52.

[7] Jonathan Gratch, Arno Hartholt, Morteza Dehghani, and Stacy Marsella. 2013. Virtual humans: a new toolkit for cognitive science research. *Applied Artificial Intelligence* 19 (2013), 215–233.

[8] Dušan Jan, David Herrera, Bilyana Martinovski, David Novick, and David Traum. 2007. A computational model of culture-specific conversational behavior. In *Intelligent virtual agents*. Springer, 45–56.

[9] Staffan Larsson and David R Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering* 6, 3&4 (2000), 323–340.

[10] Kornel Laskowski, Jens Edlund, and Mattias Heldner. 2011. A single-port non-parametric model of turn-taking in multi-party conversation. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 5600–5603.

[11] Cheong-Jae Lee, Sang-Keun Jung, Kyung-Duk Kim, Dong-Hyeon Lee, and Gary Geun-Bae Lee. 2010. Recent approaches to dialog management for spoken dialog systems. *Journal of Computing Science and Engineering* 4, 1 (2010), 1–22.

[12] Peter Ljunglöf. 2009. trindikit. py: An open-source Python library for developing ISU-based dialogue systems. *Proc. of IWSDS* 9 (2009).

[13] Thomas Lopez, Pierre Chevaillier, Valérie Gouranton, Paul Evrard, Florian Nouviale, Mukesh Barange, Rozenn Bouville, and Bruno Arnaldi. 2014. Collaborative virtual training with physical and communicative autonomous agents. *Computer Animation and Virtual Worlds* 25, 3-4 (2014), 485–493.

[14] Richard JD Power and Maria Felicita Dal Martello. 1986. Some criticisms of Sacks, Schegloff, and Jefferson on turn taking. *Semiotica* 58, 1-2 (1986), 29–40.

[15] Anand S Rao, Michael P Georgeff, et al. 1995. BDI Agents: From Theory to Practice.. In *ICMAS*, Vol. 95. 312–319.

[16] Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies*. ACL, 629–637.

[17] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *language* (1974), 696–735.

[18] Marc Schroder, Elisabetta Bevacqua, Roddy Cowie, Florian Eyben, Hatice Gunes, Dirk Heylen, Mark Ter Maat, Gary McKeown, Sathish Pammi, Maja Pantic, et al. 2012. Building autonomous sensitive artificial listeners. *IEEE Transactions on Affective Computing* 3, 2 (2012), 165–183.

[19] Gabriel Skantze and Samer Al Moubayed. 2012. IrisTK: a statechart-based toolkit for multi-party face-to-face interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 69–76.

[20] Tanya Stivers, Nicholas J Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter, Kyung-Eun Yoon, et al. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences* 106, 26 (2009), 10587–10592.

[21] David Traum. 2003. Issues in multiparty dialogues. In *Workshop on Agent Communication Languages*. Springer, 201–211.

[22] David Traum and Jeff Rickel. 2002. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*. ACM, 766–773.

[23] Andrea Zhok. 2009. Towards a theory of social practices. *Journal of the Philosophy of History* 3, 2 (2009), 187–210.