A shorter version of this paper with the same title
appeared in Proc. Int. Conf. on Intelligent Virtual Agents,
Edinburgh, UK, 2013.

# Activity Planning for Long-Term Relationships

William Coon, Charles Rich, and Candace L. Sidner

Worcester Polytechnic Institute
Worcester, MA, USA
{rich,sidner}@wpi.edu

**Abstract.** We have implemented a general-purpose algorithm for planning appropriate joint activities in the context of an interactive system that has a long-term relationship with its user. The algorithm is data-directed and explicitly models the difference between relationship stages, such as stranger, acquaintance and companion. We have conducted a short laboratory evaluation of the algorithm that demonstrates the plausibility of its results according to the judgements of participants.

## 1 Introduction

It is no longer uncommon for people to use electronic assistants of various kinds every day of their lives for years on end. Many of these systems change their behavior over time in a personalized way, such as remembering your purchasing preferences. However, none of these systems currently has the explicit goal of developing a close relationship with its user or the ability to plan for how to achieve that goal.

One of the main methods for developing closeness is appropriate shared activities. For example, an electronic home companion for isolated older adults[1] might support a wide range of activities, including chatting about the weather or sports, assisting with the maintenance of a personal appointment calendar, and coaching the user to get more exercise. Furthermore, the user might interact with the system several times per day for weeks or months or more. Clearly, certain activities, such as chatting about the weather, are appropriate on the very first day of interaction, while other activities, such as exercise coaching, should wait until the system and user develop a closer relationship. Even with a close friend, however, you don't normally start a conversation with a very difficult topic, such discussing a serious illness, but rather build up to it with some social chit-chat first.

Furthermore, because we want to support intelligent virtual agents that can sometimes take the initiative in starting activities, it is not adequate to always simply present the user with a list of all possible activities to choose from. An agent that can take initiative needs to have a dynamic model of the relationship and use it to itself plan which activities (and perhaps in what order) are appropriate for the current interaction session.

The planning algorithm presented in the remainder of this paper is very general. Both the specifics of the activities and interaction with the user are abstracted, so that it can be applied to any system that seeks to develop a long-term relationship with its user.

[1] such as `http://www.cs.wpi.edu/~rich/always`

### 1.1 Related Work

The psychology and sociology literature has explored many aspects of relationships including the factors contributing to relationships [6], the varying dimensions of relationships [11], and the behaviors involved in relationship maintenance [9]. In social penetration theory [1], the development of a relationship is modeled in terms of the increasing breadth and depth of topics available for discussion: as two people grow closer, they are able to discuss more intimate topics, as well as a broader range of topics at a given level of intimacy. Knapp [8] proposed a model of relationship development in terms of relational stages, such as stranger, acquaintance, friend, etc.

In the relevant AI and HCI literature, Bickmore [2] applied Thomason's notion of accommodation [10] to modeling collaborative relationships. Bickmore proposed that long-term collaborators develop mutual expectations that whenever certain common circumstances (activities) arise, each person will provide the appropriate assistance (accomodate) without being explicitly requested. Furthermore, Bickmore then defined the current stage of a relationship as the set of currently accomodated activities. The notion of *available activities* in our work (described further below) is inspired by Bickmore's model of accomodation, but avoids the need to commit to a fixed set of symbolic names for relational stages.

Bickmore and Schulman [4] developed a reactive algorithm based on Bickmore's staged model of relationship, which decided when it was appropriate to choose a particular dialogue act as a "bid" to advance the relationship to the next stage. In their system, the current closeness of the relationship was determined by explicit questionnaires given to the user every few days. In contrast, our approach is a planning algorithm that automatically keeps track of the closeness level and takes into account both the goal of increasing the closeness of the relationship, as well as instrumental goals, such as helping the user get more exercise.

Several other systems have also influenced our current work. The REA virtual agent [5] used a simple model of its relationship with the user in order to determine when to engage in social chit-chat versus task-oriented dialogue. The FitTrack virtual exercise coach [3] used affective language and nonverbal signals to try to increase closeness with the user. The Autom robotic weight loss coach [7] modeled its relationship with the user as being in one of three states: initial, normal and needing repair.

### 1.2 System Framework

Fig. 1 shows the activities planner in context. Whereas other parts of the interactive system are busy managing the moment-by-moment details of interaction, the role of the activities planner is to take a long-term view of the relationship with the user. In particular, the activities planner is concerned with increasing the closeness of the relationship through appropriate choice of joint activities. Closeness is important both for its own sake and because it is a prerequisite for some useful activities.

In order to allow the planner to be reused, we have made as few assumptions as possible about the system in which it is embedded. We call the basic operating cycle of the overall system a *session*. A session might be the interval between the user's login and logout or, in the case of a continuously operating robot, between engagement initiation and termination. In the virtual companion system we are building, we expect a typical session to last 20 to 30 minutes and occur several times per day over a six-week period.

At the start of each session, the activities planner produces a *session plan*. Figure 3 shows examples of session plans. Notice that these plans are very high-level. They specify the order of activities, such as "baseball chat" followed by "calendar help," but not the specific details of the activity, such as the dialogue, that will occur during the interaction session.

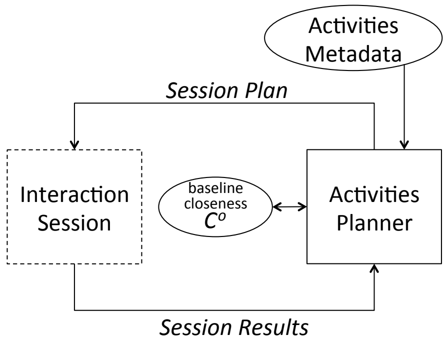



**Fig. 1.** System framework.

Session plans are also conditional. For example, the left plan in Fig. 3 specifies that after "baseball chat" either "humorous anecdotes" or "calendar help" can occur. The choice between these activities is made during the interaction session, either by the user (e.g., as the response to a menu choice) or by the system (e.g., based on its heuristics).

The key state variable in the framework in Fig. 1 is $C^o \geq 0$, the *baseline closeness*. This is the closeness level at the start of the next interaction session. It is used by the planner in the planning process and updated at the end of each interaction session, based on the results of the session.

The baseline closeness is similar to the "current stage of the relationship" in related work, except that in our framework, we use a (non-negative) number as the value of this variable to avoid committing to a predefined set of symbolic names for relational stages. Our approach also unifies the concept of the baseline closeness that carries over between sessions with the more contextual level of closeness that increases during a given session. The details of how closeness values are specified and used will be discussed further below.

The main input to the planning process is a database of *activities metadata*. For each activity, the planner needs to know the minimum baseline closeness at which it becomes available and the four planning parameters shown in the example activities database in Table 1. The meaning and use of this metadata in the planner will be explained in detail below. The important point here is that each activity name in the database corresponds to an activity that the overall system is capable of performing during an interaction session.

Finally in Fig. 1, at the end of each session the planner receives information about the results of the session, primarily which activities were successfully completed. The planner uses this information to compute if and how to update the baseline closeness for the next session.

## 2 The Planning Algorithm

The activities planner uses a simple exhaustive forward-chaining algorithm to find conditional plans all of whose final states have high utility.

### 2.1 Activities Metadata

Actions in the plan are activities, each of which is described by the following five metadata parameters:

- $C_A^o$, the minimum baseline closeness at which this activity becomes available,
- $C_A$, the minimum closeness required to start this activity,
- $\Delta t_A$, the expected duration of activity in minutes,
- $I_A$, the expected instrumental utility of this activity, and
- $\Delta C_A$, the expected relational utility of this activity.

Before describing how these parameters are used in the planning algorithm, it will be useful to explain their meaning from the point of view of the "author" of the metadata. For example, Table 1 shows the metadata we authored for the activities used in our evaluation study.

First notice that the activities (rows) in the table are divided into three groups. This grouping specifies the minimum baseline closeness at which each activity becomes available. An activity is *available* if and only if its minimum baseline closeness is satisfied, i.e.,

$$C_A^o \leq C^o \tag{1}$$

Thus the first four activities are always available; the next group are available only when the baseline closeness is at least 2; the third group only when it is at least 4.

The scale used for closeness is entirely up to the metadata authors. In our minds, we thought of 0 as corresponding roughly to the system's relational

| **Activity** A | $C_A$ | $\Delta t_A$ | $I_A$ | $\Delta C_A$ |
|---|---|---|---|---|
| $C_A^o = 0$ (*stranger*) | | | | |
| baseball chat | 0 | 10 | 1 | 2 |
| weather chat | 0 | 10 | 1 | 1 |
| play cards | 0 | 20 | 1 | 2 |
| watch TV | 0 | 10 | 0 | 1 |
| $C_A^o = 2$ (*acquaintance*) | | | | |
| humorous anecdotes | 2 | 10 | 1 | 2 |
| discuss politics | 2 | 20 | 2 | 2 |
| favorite books | 3 | 20 | 1 | 4 |
| childhood stories | 3 | 20 | 2 | 4 |
| calendar help | 3 | 10 | 3 | 1 |
| exercise routine | 3 | 30 | 5 | 2 |
| $C_A^o = 4$ (*companion*) | | | | |
| talk about family | 4 | 10 | 2 | 3 |
| organize medications | 4 | 10 | 3 | 1 |
| scrapbook | 5 | 20 | 2 | 3 |
| family emails | 5 | 20 | 2 | 3 |
| discuss non-recent family death | 7 | 20 | 1 | 6 |
| convince user write important letter | 7 | 10 | 5 | 1 |
| discuss sibling illness | 8 | 20 | 4 | 4 |
| discuss serious medical diagnosis | 8 | 20 | 6 | 3 |

**Table 1.** Activities metadata for evaluation study.

"stage" being a *stranger*, 2 an *acquaintance* and 4 a *companion.* We will use these terms in this paper as a convenient abbreviation for these closeness levels, but they do not exist in the implementation.

The first parameter column ($C_A$) in Table 1 contains values from zero to 8, specifying the minimum required closeness required to *start* an activity within a session. The fact that all of the activities available at the stranger stage also have a minimum required closeness of zero means that they can all be started at the beginning of any session. In contrast, the required closeness value of 8 for discussing a serious medical diagnosis (last row in Table 1) means that even with a companion, this activity (although available) needs to wait until later in a session when some additional contextual closeness has been built up (as we will see in the planning algorithm below).

The second parameter column ($\Delta t_A$) is the author's rough estimate of how long the activity typically takes. This information is used primarily to allow the planner to limit its effort to some reasonable time horizon. The estimated duration does not need to be highly accurate.

The third and fourth parameter columns specify the two components of the expected utility function that the planner uses to prune the plans it constructs. As with the required closeness, the scale for these values is entirely up to the metadata authors. (The weighting factor $v$ in Equation 2 is used to adjust between the closeness and utility scales.) The *instrumental utility*, $I_A$, is intended to capture the practical benefit of the activity, whereas the *relational utility*, $\Delta C_A$, is intended to capture the contribution of an activity to increasing the closeness of the relationship, i.e., the purely social benefit of the activity.

For example, in the fourth row of Table 1 we model watching TV as having no instrumental utility ($I_A = 0$) and a small relational utility ($\Delta C_A = 1$). Other activities, such as discussing a serious medical diagnosis (last row in Table 1), may have both high instrumental utility ($I_A = 6$) and high relational utility ($\Delta C_A = 3$), and so on.

Equation 2 below defines the total expected utility $U_A$ of performing activity $A$ as the weighted sum of these two parameters, where $u$ and $v$ are the weighting factors. Notice that $U_A$ includes the relational utility because the agent has the explicit goal of achieving a closer relationship with the user.

$$U_A = uI_A + v\Delta C_A \tag{2}$$

### 2.2 Generating Planning States

The planning state is represented by a three-tuple of non-negative numbers $\langle t, C, U \rangle$, where

- $t$ is the time in minutes from the start of the session,
- $C$ is the current closeness in the session, and
- $U$ is the expected utility of reaching this planning state.

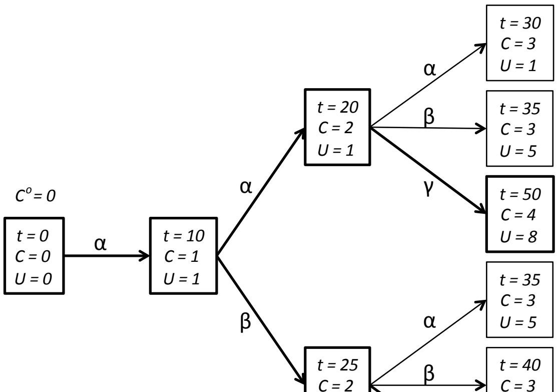
Fig. 2 is an abstract example of the execution of the planning algorithm that illustrates how planning states are generated based on the action descriptions (activity metadata). Planning states are represented by rectangles and actions by arrows. (Ignore the difference between bold and non-bold lines for now.) The inset table shows the planning metadata parameters for three abstract activites, $\alpha$, $\beta$ and $\gamma$, all of which are available when $C^o = 0$. The utility weighting factors (see Equation 2) are both 1.

The initial planning state is always at $t = 0$, with baseline closeness, $C = C^o$ (in this example $C^o = 0$), and utility $U = 0$. The possible next planning states are generated by applying each available activity, $A$, whose minimum required closeness is satisfied, i.e.,

$$C_A \leq C \quad (3)$$

In this example only $\alpha$'s required closeness is satisfied.

The components of the next state $\langle t', C', U' \rangle$ resulting from applying activity $A$ to state $\langle t, C, U \rangle$ are:




| A | $C_A$ | $\Delta t_A$ | $I_A$ | $\Delta C_A$ |
|---|---|---|---|---|
| $\alpha$ | 0 | 10 | 0 | 1 |
| $\beta$ | 1 | 15 | 3 | 1 |
| $\gamma$ | 2 | 30 | 5 | 2 |

**Fig. 2.** Abstract example of planning algorithm.

$$t' = t + \Delta t_A \quad (4) \qquad C' = C + \Delta C_A \quad (5) \qquad U' = U + U_A \quad (6)$$

Notice that the relational utility, $\Delta C_A$, is used *twice* in these updating rules. It is added to both the closeness and also to the utility (via Equation 2 above).

Returning to Fig. 2, in the state after applying $\alpha$, the closeness is 1, so that both $\alpha$ and $\beta$ can be applied, yielding the two possible successor states shown. All three activities can be applied in each of these two states, leading to 6 terminal states with a planning horizon of 30.

### 2.3 Utility Damping

Notice that this plan involves possibly performing activity $\alpha$ twice in a row. It is usually inappropriate to perform the same activity repeatedly, because some activities intrinsically deliver diminishing returns and/or users simply become bored. Our planning algorithm therefore incorporates a refinement of the utility calculation above that takes into account preceding occurrences of the same activity *in the same session.* Longer-term shifts in the utility values for activities are outside the scope of the activity planner and could be handled by other learning or user modeling techniques.

Thus in our refined planning algorithm the utility of an activity is modeled as a function of time defined as follows, where there are $n \geq 0$ preceding completed occurrences of $A$ at times $t_i < t$:

$$U_A(t) = U_A \prod_{i=0}^{n} \left(1 - \frac{1}{(1+t-t_i)^d}\right) \tag{7}$$

This definition has the effect of reducing (damping) the utility of $A$ asymptotically to zero whenever there are preceding occurrences. The smaller the value of the damping constant, $d$, the stronger the damping effect. For example, in Fig. 2 we are using a very small value of $d$ that reduces the utility of the second occurrence of $\alpha$ to approximately zero (which is why $U = 1$ in the state after the second application of $\alpha$).

### 2.4 Pruning

The exhaustive plan generation algorithm described above stops whenever a planning state's time meets or exceeds a specified limit (the horizon). For example, in Fig. 2 we have set the planning horizon at 30 minutes. Each terminal state in the plan has an expected utility. In Fig. 2 these utility values range from 1 to 12. The final step in the planning algorithm is to prune all paths in the plan that lead to a terminal state with utility below some threshold. In Fig. 2 we have used a threshold of 8. The bold lines in the figure indicate those parts of the original plan that remain in the session plan after pruning.

## 3 Running the Framework

Returning now to Fig. 1, let's look at an example of how the planning algorithm works together with the interaction session. The left plan in Fig. 3 shows was produced with a baseline closeness of 3 and the metadata in Table 1. Let's suppose that during the interaction session using this plan, the user successfully completes the baseball chat, chooses and successfully completes the calendar help activity, and then ends the session.

Information about the successfully completed activities (signified by the check marks in Fig. 3) is passed back to the activities planner at the end of the session, as indicated in Fig. 1, which allows the planner to calculate that the closeness at the end of the session was 6 (the two completed activities added to the initial closeness of 3). The planner therefore updates the baseline closeness to 6, so that, for example, if the next session starts *immediately*, it will continue with the same contextual closeness as where the current session left off. This also means that the relationship has in effect advanced to the companion stage, so that more activities will be available in the next session.

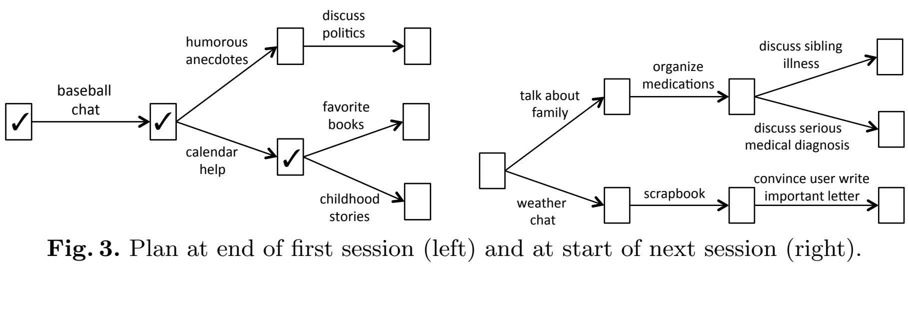



**Fig. 3.** Plan at end of first session (left) and at start of next session (right).

### 3.1 Time Decay

This updating rule for baseline closeness is too simple, however, because it does not take into account the fact that closeness decays with the passage of time if there is no interaction (i.e., between sessions). We therefore use the following refined definition for baseline closeness as a function of time, where $C^o$ is the updated baseline closeness at the end of the previous session (as described above) and $t_f$ is the ending time of the previous session. Notice that unlike in the planning update rules above, where $t$ is the number of minutes from the start of the session, time in this equation is real-world time.

$$C_0(t) = \max\left(\text{stage}(C^o),\ \frac{C^o}{(1+t-t_f)^k}\right) \tag{8}$$

The 'stage' function above returns the largest minimum baseline closeness in the current activities database less than or equal to the given value, i.e., $\text{stage}(C^o) \leq C^o$. In this example, stage(6) returns 4 for Table 1, i.e., the minimum baseline closeness for companions. The rest of this definition has the effect of reducing (decaying) the baseline closeness to $\text{stage}(C^o)$ as time passes. The larger the value of the decay constant, $k$, the more quickly the closeness decays.

Thus, returning to our example, when the next session starts some time later, the planner uses Equation 8 to calculate the baseline closeness for the initial planning state. The right plan in Fig. 3 is the resulting plan, in which we assume that enough time has passed since the first session to fully decay the baseline closeness to 4, which is used as the starting value for the planner, as explained above. Notice that the plan for the this session includes activities that were not available for the first session, such as talk about family and organize medications.

## 4 Evaluation

The ultimate evaluation of the activities planner will occur when we embed it in our companion agent and field test it in long-term relationships. However, since that field test is more than a year away, we undertook a laboratory user study to evaluate whether the planner produces plans that at least are consistent with general expectations of how relationships develop. Our basic approach is to see whether changing key features of the planner's algorithm has a measurable effect on the naturalness and plausibility of the resulting plans.

### 4.1 The Antisocial Planner

For reasons that will be made clear below, we call the modified planner the "antisocial" planner, as compared to our current "social" planner. The antisocial planner violates three key features of our activities planning algorithm.

First, the antisocial planner ignores Equation 1. In other words, whereas the social planner only applies activities that are appropriate to the current baseline closeness, the antisocial planner considers all activities as available at all times.

Second, the antisocial planner ignores Equation 3. In other words, whereas the social planner waits until later in a session to start activities to require more than the baseline closeness, the antisocial planner will start any activity at any time.

Finally, the antisocial planner goes even further toward being antisocial by using a different utility computation. Instead of Equations 2 and 7, it uses the following equation to compute the utility of activity $A$.

$$\overline{U}_A(t) = uI_A - v\Delta C_A + \max(0, C_A - C(t)) \quad (9)$$

In this equation, the relational utility ($\Delta C_A$) is *subtracted* instead of added to the overall utility, causing the antisocial planner to prefer activities that contribute less to the relationship. The last term in this equation causes the antisocial planner to not only include activities that would be unavailable in the social planner, but to *prefer* such activities—specifically, the more that an activity's required closeness is violated, the higher its utility. (The max function prevents any positive utility being added when $C_A$ is greater than $C(t)$.)

### 4.2 Hypothesis

The hypothesis of our study is that people will judge interaction scenarios based on plans generated by the social planner as more natural and plausible than scenarios based on plans generated by the antisocial planner.

### 4.3 Study Materials

Fig. 4 shows an example of a novel questionnaire we designed to elicit judgements regarding the relative naturalness and plausibility of several alternative interaction scenarios. In particular, we were concerned about not forcing the participants to specify a total order on the scenarios, because we thought it likely that some choices might be "tied" in their preferences. Our solution was to allow participants to enter more than one letter (identifying a scenario) in each of the boxes on the left of the form. Thus, in the example questionnaire, the participant has indicated that she thinks A is the most natural scenario, C is the least natural, and B and D are in between. We discuss below how these questionnaires were scored.

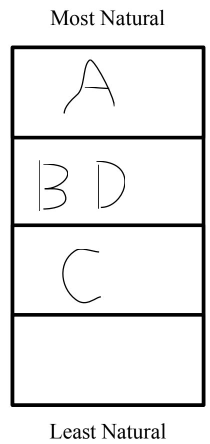
We used this questionnaire design to compare judgements in six different conditions: social versus antisocial planner, each with baseline closeness values of 0 (stranger), 2 (acquaintance) and 4 (companion).

First, we ran the planner in each condition and chose the two highest-utility paths through the six generated conditional plans, giving us 12 total scenarios. Then, we produced 4 versions of the questionnaire for each baseline closeness level, each contained two scenarios generated by the social planner and two generated by the antisocial planner in a different random overall order.

We also produced a training questionnaire using the same format, based on food preferences, for the participants to practice with (see below).

**In these scenarios, Samantha and Katherine have met several times over the past two weeks, and have become somewhat comfortable with each other.**

**Please rank the four interactions on the diagram below in terms of how natural and plausible they seem, using the labels A, B, C, and D.**




**A**

*First*, Samantha and Katherine talk about a recent baseball game for 20 minutes.
*Next*, they trade humorous anecdotes for 20 minutes.
*After that*, Katherine tells childhood stories for 40 minutes.
*Then*, they organize Katherine's schedule for 20 minutes.
*Next*, they talk about Katherine's family for 20 minutes.
*Finally*, they sort out Katherine's medicine for 20 minutes.

**B**

*First*, Samantha and Katherine talk about a recent baseball game for 20 minutes.
*Next*, they organize Katherine's schedule for 20 minutes.
*After that*, they sort out Katherine's medicine for 20 minutes.
*Then*, they talk about Katherine's family for 20 minutes.
*Next*, Katherine tells childhood stories for 40 minutes.
*Finally*, they trade humorous anecdotes for 20 minutes.

**C**

. . .

**D**

. . .

**Fig. 4.** Part of sample questionnaire used in evaluation study.

### 4.4 Study Protocol

We conducted a study in our laboratory with 12 participants (7 male, 5 female), all of whom were enrolled students at WPI and were compensated $5 or lab credit for their participation.

Each participant was first trained on the food preference questionnaire by an experimenter until they were both satisfied that the participant understood how to use the format.

The experimenter then explained that the participants were to evaluate the naturalness and plausibility of the presented scenarios in the context of a community worker, Samantha, visiting an older adult, Katherine, in her home. Each participant was then, in random order, given a randomly chosen version of the questionnaire for the stranger, acquaintance and companion conditions. The participant had as much time as they wanted to fill out each questionnaire.

### 4.5 Coding and Scoring

The raw data for each of the 12 participants is shown in Table 2. Each row of the table corresponds to one participant. The first three columns show how they filled in the questionnaire for each closeness condition.

The raw data is presented as follows. The two scenarios on each questionnaire generated by the social planner are coded as 1 and 2; the two scenarios generated by the antisocial planner are coded as 3 and 4. (Remember that these scenarios are randomly ordered on the questionnaire and the participants only see the letters A, B, C and D.) Thus, for example, the questionnaire in Fig. 4 (which corresponds to the grayed cell in Table 2) is coded as "1 / 2 4 / 3" because A and B on this form were generated by the social planner and C and D by the antisocial planner.

We scored the data by counting the number of times that a social scenario is preferred to an antisocial scenario (a *positive* count) and vice versa (a *negative* count). These counts over all three closeness conditions for each participant are shown in the last two columns of Table 2. Table 3 summarizes the positive and negative counts and the p-values for their difference for all participants overall and by closeness condition.

| $C^o = 0$ *(stranger)* | $C^o = 2$ *(acquaintance)* | $C^o = 4$ *(companion)* | social > anti | anti > social |
|---|---|---|---|---|
| 1 2 / 3 4 | 1 2 / 3 / 4 | 4 / 3 / 1 / 2 | 8 | 4 |
| 1 / 3 / 2 / 4 | 1 / 2 4 / 3 | 2 / 4 / 1 4 | 8 | 2 |
| 1 / 2 / 3 / 4 | 2 / 1 / 3 4 | 1 2 / 3 / 4 | 12 | 0 |
| 1 / 2 / 3 / 4 | 2 / 1 / 4 / 3 | 4 / 1 / 2 / 3 | 10 | 2 |
| 1 / 3 / 2 / 4 | 3 / 2 / 4 / 1 | 2 / 1 / 3 4 | 8 | 4 |
| 1 / 2 / 3 / 4 | 2 / 1 / 3 / 4 | 4 / 2 / 1 / 3 | 10 | 2 |
| 1 2 / 3 / 4 | 3 / 1 2 / 4 | 3 4 / 1 2 | 6 | 6 |
| 1 2 / 3 4 | 1 / 2 / 3 / 4 | 2 4 / 1 3 | 9 | 1 |
| 1 / 3 / 2 4 | 1 / 2 4 / 3 | 1 2 / 4 / 3 | 9 | 1 |
| 3 / 1 2 / 4 | 2 / 1 4 / 3 | 2 / 1 4 / 3 | 8 | 2 |
| 1 / 2 3 4 | 2 / 1 4 / 3 | 1 2 / 3 4 | 9 | 0 |
| 2 / 1 / 3 /4 | 2 / 1 3 / 4 | 4 / 2 / 3 / 1 | 8 | 3 |

**Table 2.** Raw tabulation of questionnaires.

### 4.6 Discussion

According to our hypothesis, the positive counts should exceed the negative accounts. This is strongly supported overall and for both the stranger and acquaintance conditions individually.

| | social > anti | anti > social | p-value |
|---|---|---|---|
| $C^o = 0$ *(stranger)* | 40 | 5 | $7.88e^{-8}$ |
| $C^o = 2$ *(acquaintance)* | 38 | 5 | $2.5e^{-7}$ |
| $C^o = 4$ *(companion)* | 27 | 17 | 0.174 |
| Overall | 105 | 27 | $2.07e^{-11}$ |

**Table 3.** Comparison of evaluations.

The fact that the result for the companion condition was weak is understandable on both psychological and algorithmic grounds. On the one hand, behavior that deviates from social norms is more easily tolerated from those we are close to. On the other hand, the fact that the antisocial planner ignored the minimum required closeness for availability made less of a difference at the companion closeness level, because more of the activities were available to the social planner anyways.

## 5 Conclusion

Informed by the relevant psychological and sociological models, we have implemented a general-purpose algorithm for planning appropriate joint activities in the context of a long-term human-computer relationship. This algorithm will become a module in the virtual agent companion system we are building for isolated older adults. However, we believe our algorithm could also be useful in many other types of intelligent interactive systems as they become more pervasive and persistent.

Before investing in a long-term in-situ study, we have completed a short laboratory evaluation of the algorithm, which successfully demonstrated its plausibility.

Finally, we want to point out that this work only scratches the surface of computationally modeling long-term human-computer relationships. For example, we have ignored the important effects of factors such as relative status, gender and personality, to name just a few. There is a rich literature on all of these topics that is waiting to be adapted into practical algorithms.

**Acknowledgments** This work is supported in part by the National Science Foundation under award IIS-1012083. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

1. Altman, I., and Taylor, D. *Social Penetration: The Development of Interpersonal Relationships*. Holt, Reinhart and Winston, New York, 1973.
2. Bickmore, T. *Relational Agents: Effecting Change through Human-Computer Relationships*. PhD thesis, MIT Media Laboratory, 2003.
3. Bickmore, T., and Picard, R. Establishing and maintaining long-term human-computer relationships. *ACM Trans. on Computer Human Interaction 12*, 2 (2005), 293–327.
4. Bickmore, T., and Schulman, D. Empirical validation of an accomodation theory-based model of user-agent relationship. In *Proc. Int. Conf. on Intelligent Virtual Agents* (Santa Cruz, CA, 2012).
5. Cassell, J., and Bickmore, T. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User Modeling and User-Adapted Interaction 13*, 1 (2003), 89–132.
6. Cole, T., and Bradac, J. A lay theory of relational satisfaction with best friends. *J. Social and Personal Relationships 13*, 1 (1996), 57–84.
7. Kidd, C., and Breazeal, C. A robotic weight loss coach. In *Proc. 22nd National Conference on Artificial Intelligence* (Vancouver, Canada, 2007).
8. Knapp, M. *Interpersonal Communication and Human Relationships*. Allyn & Bacon, Boston, MA, 1984.
9. Stafford, L., Dainton, M., and Haas, S. Measuring routine and strategic relational maintenance: Scale revision, sex versus gender roles, and the prediction of relational characteristics. *Communication Monographs 67* (2000), 306–323.
10. Thomason, R. Accommodation, meaning, and implicature: Interdisciplinary foundations for pragmatics. In *Intentions and Communication*, P. R. Cohen, J. L. Morgan, and M. E. Pollack, Eds. MIT Press, Cambridge, MA, 1990, 325–364.
11. Wish, M., Deutsch, M., and Kaplan, S. Perceived dimensions of interpersonal relations. *J. Personality and Social Psychology 33*, 4 (1976), 409–420.