Promises Kept, Promises Broken: An Axiomatic and Quantitative Treatment of Fulfillment

G. I. Simari M. Broecheler V. S. Subrahmanian
Department of Computer Science
University of Maryland College Park
College Park, MD 20742, USA
{gisimari,matthias,vs}@cs.umd.edu

Sarit Kraus
Department of Computer Science
Bar-Ilan University
Ramat Gan, Israel
sarit@cs.biu.ac.il

Abstract

In this paper, we propose a theoretical framework within which to evaluate the reliability of promises that an agent makes, based on past performance of the agent. Our framework does not just propose one such measure, but defines axioms that govern the choice of measure. The framework is able to account for partial fulfillment of promises, late fulfillment of promises, fulfillment of variants of promises, and the like. Within this framework, we propose some specific measures to evaluate promises made by agents and develop algorithms to compute these efficiently. We tested our methods on a real world data set of airline flight information and show that our methods are both accurate and quickly computable, even on large data sets.

Introduction and Motivating Example

Politicians and political parties are prime examples of people and organizations that make and break promises. Conference and journal reviewers often promise to review papers by a deadline, but may not meet the deadline, may only review part of their assigned load, or not do it at all. Airlines promise to deliver passengers and their bags by a deadline, but may miss the deadline altogether. Suppliers to manufacturing plants and/or to retail outlets make promises about when inventory and/or supplies will be delivered, but may meet their promises partially or completely.

The goal of this paper is to develop a formal theory to *quantitatively* evaluate how well an agent has fulfilled its past promises and use that as a predictor of whether it will keep its current (as yet unfulfilled promises). An agent A can use the theory developed in this paper to assess the likelihood that an agent B will fulfill a given promise. Our framework takes into account three important factors not considered before: *partial fulfillment* of a promise is taken into account, as is *late fulfillment*, fulfillment of a promise that is *similar to, but not identical* to the promise that was made, and combinations thereof. The toy example below, called the Store example, is used throughout the paper.

Example 1. *Consider a Store agent and a Supplier agent. The supplier provides, among other things, shirts and balls to the store. The supplier promises to deliver 10 blue balls by time t_1, 5 green balls by time t_2, and 15 green shirts by time t_3. Here are some possible scenarios.*
(S1) He delivers 7 blue balls at time t_1 and 3 blue balls separately at time t_1.
(S2) He delivers 7 blue balls at time t_1 and 3 blue balls at time $t_1 + 1$.
(S3) He delivers 7 blue balls at time t_1, 2 at time $t_1 + 1$ and 1 at time $t_1 + 2$.
(S4) He delivers 10 blue balls at time $t_1 - 3$.
Most readers will agree that the degree of fulfillment of the supplier's promise in S1 exceeds that in S2 which in turn exceeds that in case S3. However, case S4 is less clear. Should early delivery be penalized? For example, a grocery store may want just in time *delivery as there are storage costs involved. An early delivery of fish by one supplier may cause the fish to rot if sufficient refrigeration is not available when the delivery occurs.*

In this paper, we make the following contributions. First, we define a formal syntax for expressing promises and actions that an agent might take to keep those promises. These include promises and actions with a numerical component. We then define distance measures between actions (with the same action symbol), followed by distance measures between sets of promises and actions. Rather than define distance measures directly, we develop *axioms* that such distance measures should satisfy and we then show some example distance measures that satisfy the axioms.

We then define the concept of an *enactment mapping*, which maps actions taken by an agent to the promises that those actions were intended to contribute towards. Given an enactment mapping, we can define a degree of fulfillment of a promise w.r.t. the enactment mapping. When the enactment mapping is not known, finding the enactment mapping that maximizes the degree of fulfillment is shown to be computationally intractable. Fortunately, one can get around this complexity result easily as long as the agent taking an action specifies which promise the action is supposed to contribute towards. We prove various desirable properties of our notion of fulfillment and show that when the enactment mapping is fixed, it leads to an incremental way of updating the degree of fulfillment when new actions are performed, *i.e.*, without having to process past actions all over again. We then derive specific methods to estimate the likelihood that a given promise will be fulfilled in the future.

In order to test our methods, we developed a prototype implementation of our system and tested it out on real US airline data where the promises pertain to on-time flight departures and arrivals. We show that our estimation methods have strong predictive power. We used our algorithms to predict how well airlines would perform (in terms of on-time flight arrivals) in 2007 based on previous years' data. Our predictions were highly accurate and took small amounts of compute time.

There is some past work on developing models of trust in agent systems. (Sierra and Debenham 2007) presents a model of decision making based on trust in simple *Offer, Accept, Reject* negotiations. Decision-making in this model integrates the utilitarian, information, and semantic views of the exchange of information, and the authors present summary measures that generalize trust, reliability, and reputation as an illustration of the model's capabilities. However, promises of the kind we discuss in this paper are not considered. Another important difference with our approach is that these measures assume the availability of probability distributions that describe the *ideal* enactments with respect to a given commitment, expected enactments, a more general *semantic similarity* measure that allows to gauge the similarity between the commitment and its actual enactment, and a measure of how much uncertainty we expect to have given a certain commitment. Other related work is that in the area of trust and reputation in agent systems (which can include both artificial and human agents). (Dondio and Barrett 2007) propose a generic method of selecting evidence that is recognized as support for trust, while (Dellarocas 2006) provides a recent survey of the area, focusing on Internet-based mechanisms (such as for online auctions). Game-theoretic treatments of this topic have also been developed, such as in (Ely, Fudenberg, and Levine 2002), but this approach has been criticized for placing too much importance on probability while underestimating its cognitive aspects, such as in (Falcone and Castelfranchi 2001). In this respect, our work takes a step in this direction by allowing agents to influence the measure of fulfillment according to their own preferences.

In contrast to this past work, we focus on developing a general model of promises that tries to quantitatively assess how well an agent has met its past commitments, taking into account the fact that time plays a role in whether a promise is met or not (promises often involve doing things by a deadline), that the "content" of an action sometimes - but not always - allows a promise to be replaced by a similar, but different promise (e.g. delivering red balls instead of green balls), and partial fulfillment where part of a promise is kept. Our framework is one of the first to develop a unified theory around these important concepts.

Preliminaries

We start by defining the notion of *temporal expression*, which is used to denote time points: we assume that time in our model is discrete.

Definition 1 (from (Dix, Kraus, and Subrahmanian 2006)). *(1) Every integer is a temporal expression. (2) t_{now} is a temporal expression. (3) If t_1 and t_2 are temporal expressions, then so is (t_1 + t_2).*

Symbol t_{now} represents the current time point; we assume its value gets automatically updated as time goes by in the environment.

We assume the existence of a logical alphabet that consists of a finite set $\mathcal{L}$ of constant symbols, a finite set $\mathcal{A}$ of *action symbols* (each with an associated arity), the predicate symbols *Do* and *Promise*, and an infinite set $\mathcal{V}$ of variable symbols. A constant or variable symbol is called a *term*.

Definition 2 (action atom). *If $\alpha \in \mathcal{A}$, and $t_1, ..., t_m$ $(m \geq 0)$ are terms (resp. constants), then $\alpha(t_1, ..., t_m)$ is called an* action atom *(resp. ground action atom).*

$atoms(\alpha)$ denotes the set of all possible action atoms of the form $\alpha(\ldots)$.

Definition 3 (*Do* and *Promise* atoms). *Suppose A, B are agents, T is a temporal expression, and X is an action atom. Then,* Do*(A,B,X,T) and* Promise*(A,B,X,T) are called* Do *and* Promise *atoms respectively.*

Intuitively, *Do*(A,B,X,T) is read "agent A *does* X for agent B at time T" while *Promise*(A,B,X,T) is read "agent A *promised* agent B that it would do X at time T". The following example, based on the Store example above, is presented in order to illustrate these concepts.

Example 2. *Let A and B be the Supplier and Store agents, Here are some example* Promise *and* Do *atoms involving the action $del(it, col, am)$ which states that am amount of item it of color col are delivered.*

$P_1 = \textit{Promise}(B, A, del(ball, blue, 10), T_1),$
$P_2 = \textit{Promise}(B, A, del(shirt, green, 5), T_2),$
$P_3 = \textit{Promise}(B, A, del(shirt, green, 15), T_3),$
$D_1 = \textit{Do}(B, A, del(ball, blue, 7), T_1),$
$D_2 = \textit{Do}(B, A, del(ball, darkBlue, 3), T_1),$
$D_3 = \textit{Do}(B, A, del(shirt, green, 20), T_3)$

However, it might be the case that the Store agent is neutral about whether the supplier delivers blue balls or green balls, even though the supplier promised blue balls. We therefore need a binary *replaceability* relation on action atoms in order to capture this type of situation.

Definition 4. *Let A be an agent, and $S_1 = \{\alpha_1(\vec{t_1}), ..., \alpha_l(\vec{t_l})\}$ and $S_2 = \{\beta_1(\vec{u_1}), ..., \beta_m(\vec{u_m})\}$ be two sets of action atoms. We assume each agent A has an associated relation of replaceability, denoted $S_1 \rightleftharpoons^A S_2$, read as:* S_1 is replaceable by S_2 for agent A. *We only require $S \rightleftharpoons^A S$ for any set S and agent A.*

When the agent is clear from context, we will simply write $S_1 \rightleftharpoons S_2$. The above definition allows us to consider a promise to be fulfilled when the agent has taken an action that is considered good enough, even though it does not exactly fulfill the promise as stated. The store manager who thinks it is all right to replace blue balls with green ones may set $\{del(ball, blue, N_1), \ldots, del(ball, blue, N_m)\} \rightleftharpoons \{del(ball, green, M_1), \ldots, del(ball, green, M_k)\}$ iff $N_1 + \ldots + N_m = M_1 + \ldots + M_k$.[1] One reason we need the

[1]Due to space constraints we do not provide an explicit syntax to express the $\rightleftharpoons$ relation.

$\rightleftharpoons$ relation is because a Supplier might have made multiple promises (of 5 green balls and 3 green balls all to be delivered at time 7) and may execute multiple *Do* actions (*e.g.*, by delivering two packages each of 4 green balls at time 7) that jointly meet the promises. In order to reason about this kind of situation, we need ways of aggregating promises together. We start by defining two sets. Given agents A, B, and a temporal expression T:

- $\mathcal{U}^{Prom}_{A,B,T} = \{\langle Promise(A, B, a_i, T), \omega_i\rangle \mid 0 \leq \omega_i \leq 1$ and *Promise*(A, B, a_i, T) is a promise atom $\}$. ω_i is any real number in the $[0, 1]$ interval called the *proportion component*.
- $\mathcal{U}^{Do}_{A,B,T} = \{Do(A, B, a_i, T) \mid a_i$ is an action atom $\}$.

The following definition specifies whether it is possible to merge multiple *Promise* atoms or *Do* atoms into one.

Definition 5. *Let A and B be agents, T be a temporal expression, $S_p \subseteq \mathcal{U}^{Prom}_{A,B,T}$, and $S_d \subseteq \mathcal{U}^{Do}_{A,B,T}$.*

- *Suppose a^* is an action atom such that $\{a_i \mid \langle Promise(A, B, a_i, T), \omega_i\rangle \in S_p\} \rightleftharpoons \{a^*\}$. The* promise composition *operator χ takes any subset $S_p \subseteq \mathcal{U}^{Prom}_{A,B,T}$ as input and returns the* Promise *atom $\chi(S_p) = Promise(A, B, a^*, T)$ if and only if $\sum_{\langle Promise(A,B,a_i,T),\omega_i\rangle \in S_p} \omega_i = 1$. Otherwise, it is undefined.*
- *If a^* is an action atom such that $\{a_i \mid Do(A, B, a_i, T) \in S_d\} \rightleftharpoons \{a^*\}$, then the* do composition *operator χ takes a set $S_d \subseteq \mathcal{U}^{Do}_{A,B,T}$ as input and returns the* Do *atom $\chi(S_d) = Do(A, B, a^*, T)$ if and only if $\{a_i \mid Do(A, B, a_i, T) \in S_d\} \rightleftharpoons \{a^*\}$. Otherwise, it is undefined.*

We let $\chi^{-1}(X)$, be the set of all sets $S \subseteq \mathcal{U}^{Do}_{A,B,T}$ or $S \subseteq \mathcal{U}^{Promise}_{A,B,T}$ such that $\chi(S) = X$. Informally, compositions and decompositions are simply ways in which to refer to "parts" of *Promise* and *Do* atoms. In the case of *Promise* atoms, decompositions are sets of pairs that include a proportion for each atom in the set, whereas in the case of *Do* atoms, a decomposition is just a set. In contrast, compositions specify a set of *Do* atoms as input and composes them, when possible, into a single *Do* atom.

The following is an example of combinations and decompositions of *Promise* and *Do* atoms.

Example 3. *Consider the* Promise *atoms and* Do *atoms in Example 2 and let:*

$P^1_1 = \textit{Promise}(B, A, del(ball, blue, 7), T_1)$,
$P^2_1 = \textit{Promise}(B, A, del(ball, blue, 3), T_1)$,
$D^1_3 = \textit{Do}(B, A, del(shirt, green, 5), T_3)$,
$D^2_3 = \textit{Do}(B, A, del(shirt, green, 15), T_3)$

Now, if χ_p is a promise composition operator and $S_p = \{\langle P^1_1, 0.7\rangle, \langle P^2_1, 0.3\rangle\}$, we have that $\chi_p(S_p) = P_1$. Similarly, if χ_d is a do composition operator and $S_d = \{D^1_3, D^2_3\}$, we have that $\chi_d(S_d) = D_3$.

We now define *event sets* and *action histories*.

Definition 6 (event sets and action histories)**.** *An* event set *is any finite set of ground* Do *and* Promise *atoms. An* action history *is a function h from $[0, ..., t_{now}]$ to event sets.*

An action history describes what promises were made and what actions occurred at each time point before t_{now}. We will generally be interested in *finite* action histories, *i.e.*, where $\{t \mid h(t) \neq \emptyset\}$ is finite.

A Distance Measure between Atoms

In order to determine the degree of fulfillment between promises and actions, we will develop distance functions in three phases: first between action atoms, then between *Promise* atoms and *Do* atoms, and finally between sets of *Promise* atoms and sets of *Do* atoms. Of course, these distance functions can be defined in many ways, and so we present *axioms* governing the definition of such distance functions so that application specific knowledge can play a role in our framework.

Distance between Two Action Atoms

We start with distance functions on action atoms by first considering two action atoms that share the same action symbol.

Definition 7. *A* distance measure *between two action atoms $\alpha(t_1, t_2, ..., t_n)$ and $\alpha(s_1, s_2, ..., s_n)$, from the point of view of agent A is a function $\delta^A_\alpha : atoms(\alpha) \times atoms(\alpha) \to \mathbb{R}^+ \cup \{0\}$. Function δ^A_α must satisfy the property of Weak Identity of Indiscernibles: If $a_1 = a_2$ then $\delta^A_\alpha(a_1, a_2) = 0$.*

Note that the distance measure δ^A_α is undefined when comparing atoms with different action symbols.

Example 4. *From the point of view of the Store agent, the distance between two atoms $deliver(i_1, c_1, q_1)$ and $del(i_2, c_2, q_2)$ may be $|q_1 - q_2|$ if and only if $i_1 = i_2$ and $c_1 = c_2$, and some very large constant $d \gg 0$ otherwise, indicating that the manager considers any deviation in the product to represent a large difference.*

Note that this is not a distance metric from a mathematical point of view, since symmetry and triangle inequality are not required by the definition; as we will argue in the following, these properties are not always desirable in this framework. For instance, consider actions $a_1 = del(ball, 7)$ and $a_2 = del(ball, 10)$. Here, three extra balls were delivered and we might want to set $\delta^A_{del}(a_1, a_2) = 3$. However, we might want to set $\delta^A_{del}(a_2, a_1) > \delta^A_{del}(a_1, a_2)$ because delivering three fewer balls may be less desirable. For triangle inequality, consider an order for screws with actions $a_1 = del(screws, 5mm)$, $a_2 = del(screws, 5.2mm)$, and $a_3 = del(screws, 5.4mm)$, where the first component refers to a standard sized bag of screws and the second refers to their size. If the allotted error range of the manufacturer is 0.3mm, then $\delta^A_{del}(a_1, a_2)$ and $\delta^A_{del}(a_2, a_3)$ might be 0, but $\delta^A_{del}(a_1, a_3)$ would be strictly positive.

We now present a set of axioms that describe the desired characteristics for a measure of distance between two action atoms; in the following, sharing the same action symbol. Let $a_1 = \alpha(t_1, t_2, ..., t_n)$, $a_2 = \alpha(s_1, s_2, ..., s_n)$, and $a_3 = \alpha(r_1, r_2, ..., r_n)$ be action atoms. The following definition is required before presenting the axioms.

Definition 8. *Let $a_1 = \alpha(t_1, t_2, ..., t_n)$ and $a_2 = \alpha(s_1, s_2, ..., s_n)$ be two action atoms. The* disagreement set *of the two atoms, denoted by disagree(a_1, a_2) is the set of all triples (t_i, s_i, i) such that $t_i \neq s_i$.*

We now present axioms that δ_α^A should satisfy.

Axiom A1: $\delta_\alpha^A(a_1, a_2) = 0$ iff $\{a_1\} \rightleftharpoons^A \{a_2\}$.

This axiom simply states that the distance between two actions is zero if and only if the singleton sets that contain each are replaceable from the point of view of the agent.

Axiom A2: If $disagree(a_1, a_2) \subseteq disagree(a_1, a_3)$, $a_2 \not\rightleftharpoons a_3$, and $a_1 \not\rightleftharpoons a_3$, then $\delta_\alpha^A(a_1, a_2) < \delta_\alpha^A(a_1, a_3)$.

Axiom $A2$ intuitively states that if the discordances between a_1 and a_2 are a subset of those between a_1 and a_3, and a_2 is not replaceable by a_3 (*i.e.*, the remaining differences are significant), then the distance between a_1 and a_2 is strictly smaller than that between a_1 and a_3. The following axiom deals with the case in which the remaining differences are not significant from the point of view of the agent.

Axiom A3: If $a_2 \rightleftharpoons^A a_3$, then $\delta_\alpha^A(a_1, a_2) = \delta_\alpha^A(a_1, a_3)$.

According to Axiom $A3$, the distance between an atom a_1 and two others a_2 and a_3, such that a_2 is replaceable by a_3, is the same. The following example illustrates these axioms.

Example 5. *Suppose we have $a_1 = del(bball, blue, 50)$, $a_2 = del(vball, blue, 45)$, and $a_3 = del(vball, white, 45)$. Here bball may refer to a beach ball, while vball refers to a volleyball. We then have $disagree(a_1, a_2) = \{(bball, vball, 1), (50, 45, 3)\}$ and $disagree(a_1, a_3) = \{(bball, vball, 1), (blue, green, 2), (50, 45, 3)\}$, and therefore the inclusion holds. Then, if $a_2 \not\rightleftharpoons a_3$ we have that the difference in color is significant and therefore $\delta_\alpha(a_1, a_2) > \delta_\alpha(a_1, a_3)$ according to $A2$. However, if the difference in color is not significant, which would be the case if $a_2 \rightleftharpoons a_3$, the two distances should be equal, as stated by axiom $A3$.*

Distance between a *Promise* and a *Do* Atom

We now deal with the problem of measuring the distance between a single *Promise* atom and a single *Do* atom, interpreted as being the *enactment* of the promise.

Definition 9. *A* distance measure *between a* Promise *atom $P = Promise(B, A, a_1, T_1)$ and either a* Do *atom $D = Do(B, A, a_2, T_2)$ or the special constant Null, from the point of view of agent A, is a function $\phi_\alpha^A(P, D) \rightarrow \bar{R}^+ \cup \{0\}$, where a_1 and a_2 are action atoms that share the same action symbol α.*

When clear from context, we will simply write $\phi_\alpha(P, D)$. The *Null* constant stands for the "lack of enactment", it is a key aspect of the treatment of degree of fulfillment presented in the next section. We now present axioms that constrain the value that the degree of fulfillment function can take given the various situations.

Axiom F1:

$$\phi_\alpha^A(Promise(B, A, a_1, T_1), Do(B, A, a_2, T_2)) \geq \delta_\alpha^A(a_1, a_2)$$

This basic axiom states that the distance between a *Promise* atom and a *Do* atom cannot be less than that between the actions they refer to.

Axiom F2: $\phi_\alpha^A(Promise(B, A, a_1, T_1), Null) = \infty$.

This axiom states that the distance between a *Promise* atom and the constant *Null* is infinite.

Axiom F3: If $T_1 = T_2$ then

$$\phi_\alpha^A(Promise(B, A, a_1, T_1), Do(B, A, a_2, T_2)) = \delta_\alpha^A(a_1, a_2)$$

If the action enactment was performed at the time agreed in the promise, then the distance between the promise and the enactment must be the distance between the two actions. In particular, if the action agreed upon is replaceable by the one that performed, then the distance between the promise and its enactment must be zero, according to Axiom A1.

Two key situations, early completion and late completion, are unconstrained by the axioms. This is because different scenarios can arise, both where either of these are beneficial and detrimental to the agent to which the promise was made. For instance, while early delivery of an email is most likely harmless, a manager receiving items like fish and meat that require refrigeration earlier than expected must have the appropriate storage space (e.g. refrigerator space) to store it. The following proposition presents a class of functions that satisfy all of the axioms.

Proposition 1. *Let $P = Promise(B, A, a_1, T_1)$ and $D = Do(B, A, a_2, T_2)$ be two atoms such that a_1 and a_2 have action symbol α. Any ϕ function of the form:*

$$\phi_\alpha^A(P, Null) = \infty$$

$$\phi_\alpha^A(P, D) = \begin{cases} f_\ell(T_1, T_2) + k_\ell \delta_\alpha^A(a_1, a_2) & \text{if } T_2 > T_1, \\ \delta_\alpha^A(a_1, a_2) & \text{if } T_1 = T_2, \\ f_e(T_1, T_2) + k_e \delta_\alpha^A(a_1, a_2) & \text{if } T_1 > T_2, \end{cases}$$

where $f_\ell(.)$ and $f_e(.)$ are positive real functions, and k_ℓ and k_e are constants in $R^{\geq 1}$, satisfies axioms F1, F2, and F3.

Proof. We have to show that $\phi_\alpha^A(P, D)$ satisfies all three axioms F1, F2, and F3.

Satisfaction of Axiom F1. Case 2 of the function definition, for $T_1 = T_2$, trivially satifies this axiom by definition. For cases 1 and 3, we assume that $k_\ell, k_e \geq 1$ and get:

$$\begin{aligned} \phi_\alpha^A(P, D) &= f_{\ell/e}(T_1, T_2) + k_{\ell/e} \delta_\alpha^A(a_1, a_2) \\ &\geq k_{\ell/e} \delta_\alpha^A(a_1, a_2) \geq \delta_\alpha^A(a_1, a_2) \end{aligned}$$

since $f_\ell(T_1, T_2)$ and $f_e(T_1, T_2)$ are both positive functions.

Satisfaction of axiom F2. Satisfied trivially by definition.

Satisfaction of axiom F3. Satisfied by case 2 of the function definition. □

There are many such examples of reasonable ϕ functions that fall into this category, such as one that simply fixes the weight assigned to every time unit under or over the deadline by defining $f_\ell = \ell \times |T_2 - T_1|$, $f_e = e \times |T_2 - T_1|$, for some $e, \ell \in R^+$, and $k_\ell = k_e = 1$. Of course, functions outside this class can also be defined.

A Function to Measure Degree of Fulfillment

The ϕ function presented above is the backbone of the final measure that we will present, which allows an agent to measure the *degree of fulfillment* given a set of *Promise* atoms and a set of *Do* atoms. Before introducing this function, we need the definition of an *enactment mapping*:

Definition 10. *Let A and B be agents, and $S_p = \{P_1, \ldots, P_n\}$ and $S_d = \{D_1, \ldots, D_m\}$ be sets of* Promise *and* Do *atoms, respectively. Let*

$$\Delta_p = \bigcup_{i=1}^{|S_p|} S_i \quad and \quad \Delta_d = \bigcup_{j=1}^{|S_d|} S_j$$

for some $S_i \in \chi^{-1}(P_i)$ and $S_j \in \chi^{-1}(D_j)$.
An enactment mapping *between S_p and S_d is defined as any $M_{S_p,S_d} : \Delta_p \rightarrow \Delta_d \cup \{Null\}$ that is quasi injective,* i.e.*, $\forall p_i, p_j \in S_p : M_{S_p,S_d}(p_i) = M_{S_p,S_d}(p_j) \wedge M_{S_p,S_d}(p_i) \neq$ Null $\implies p_i = p_j$.*

Intuitively, an enactment mapping associates "parts" of *Promise* atoms with "parts" of *Do* atoms (or *Null*), stating that the latter "counts towards" the former. The space of all possible such mappings is given by the different possible compositions of the Δ sets given the variation in the S sets involved. The following example shows a simple enactment mapping function:

Example 6. *Consider the set of atoms from Example 3. A possible enactment mapping M is the following:*

$M(P_1^1) = D_1$, $M(P_1^2) = D_2$,
$M(P_2) = D_3^1$, $M(P_3) = D_3^2$.

In this case, promise P_1 was split into two atoms in order to map it to D_1 and D_2, as was enactment D_3, in order to map it to P_2 and P_3.

The degree of fulfillment is then defined as:

Definition 11. *Let M be an enactment mapping between two sets S_p and S_d and let $P = Promise(A, B, a, T)$. The* degree of fulfillment *of P, denoted $degFulfill_M(P)$, is defined as:*

$$e^{-\sum_{P_i \in Dom(M_P)} \omega_i \phi_\alpha^A(P_i, M(P_i))}$$

where Dom_{M_P} is the domain of M restricted to considering only atom P. The degree of fulfillment for S_p, denoted $degFulfill_M(S_p)$, is then defined as:

$$\frac{\sum_{P_i \in S_p} \gamma^{diff(t_{now}, T_i)} degFulfill_M(P_i))}{\sum_{P_i \in S_p} \gamma^{diff(t_{now}, T_i)}}$$

where $\gamma \in (0, 1]$, T_i is the time point associated with P_i, and $diff(x, y) = x - y$ if $x > y$ and 0 otherwise.

Note that *degree of fullfillment* according to this definition is a real number in $[0, 1]$. Intuitively, this definition assumes the existence of a set of *Promise* atoms S_p and a set of *Do* atoms S_d, that represent promises made by an agent B to an agent A and what actions were carried out by B towards fulfilling such promises. In order to obtain the associated *degree of fulfillment*, A will evaluate the *distance* between the promises and the enactments by establishing a mapping from some suitable decomposition of S_p to some suitable decomposition of S_d. With such a mapping, individual degrees of fulfillment are obtained using the first part of the definition, and then each individual degree is weighted according to the time at which the promise was due as in the second part of the definition.

In Definition 11, the term $\phi_\alpha^A(P_i, M(P_i))$ refers to the distance between a given promise P_i and the action $M(P_i)$ that was performed to fulfill that promise according to enactment mapping M. Multiplying this term by the proportion ω_i of P_i gives us a weighted assessment of this distance (for cases in which the P_i's correspond to parts of an original promise). The summation over all P_i's in $Dom(M_P)$ gives us all sub-promises P_i associated with P and computes their individual fulfillments. Taking e to the negative power of this summation weights promise P's fulfillment in a way that is inversely proportional to these distances, resulting in a value in $[0, 1]$. The following example shows a simple calculation of degree of fulfillment:

Example 7. *Let us return to Examples 2 and 3, where D_1 is now changed to $Do(B, A, del(ball, blue, 5), T_1)$ instead. Consider the functions*

$\delta_{del}(del(I_1, C_1, X_1), del(I_2, C_2, X_2)) = |X_1 - X_2|$

iff $I_1 = I_2$ and $C_1 = C_2$, and ∞ otherwise, and

$\phi_{del}(Promise(B, A, a_1, T_1), Do(B, A, a_2, T_2)) =$
$\delta_{del}^A(a_1, a_2) + |T_1 - T_2|$

In this case, using mapping M from Example 6, we have the following individual degrees of fulfillment:

$degFulfill_M(P_1) = e^{-(0.7*2\ +\ 0.3*0)} = 0.246$
$degFulfill_M(P_2) = e^{-1\ *\ 0} = 1$
$degFulfill_M(P_3) = e^{-1\ *\ 0} = 1$

Then, assuming $\gamma = 0.9$, that the different time points are at unit distance, and that $t_{now} = T_3 + 1$, we get:

$$\frac{0.9^3 * 0.246 + 0.9^2 * 1 + 0.9 * 1}{0.9^3 + 0.9^2 + 0.9} \approx \frac{1.889}{2.439} \approx 0.774$$

As we have seen, mapping M plays a major role in how the degree of fulfillment is computed, and there are many ways in which this mapping can be obtained. For instance, it can be built by the agents involved in the promises made, since they can agree on this mapping when each action described by a *Do* atom is performed. Another way would be to perform a search through the space of possible mappings in order to obtain one that *maximizes* the degree of fulfillment that is obtained, *i.e.*, the mapping that is *most beneficial to agent B*. However, this approach has a high computational cost, as shown in the following result.

Proposition 2. *Given two sets S_p and S_d of* Promise *and* Do *atoms, respectively, and a real number $k \in [0, 1]$, finding an enactment mapping M such that $degFulfill_M(S_p) > k$ is NP-complete.*

Proof. We will first show membership in NP and then NP-hardness.
Membership in NP: If we are given a mapping M, checking that it is well defined (*i.e.*, that its domain and codomain are valid decompositions of S_p and S_d, respectively) can be done in polynomial time. Hence, it remains to be proven, that the size of M (where M is a relation, *i.e.* a set of pairs) is polynomial in the size of the input, *i.e.* $|S_p|$ and $|S_d|$. For this, it is important to observe that the number of elements in any minimal decomposition $\chi^{-1}(P)$ of any promise $P \in S_p$

is bounded by $|S_d|+1$. To prove this, assume the contrary. Hence, we have $\left|\chi^{-1}(P)\right| > |S_d|+1$, from which we can conclude that at least two elements p_1, p_2 must be mapped onto elements d_1, d_2 which are part of the decomposition of a single do atom $D \in S_d$, *i.e.*, $d_1, d_2 \in \chi^{-1}(D)$. Hence, we can merge p_1, p_2 and d_1, d_2 within their respective decompoositions and arrive at a new mapping M^* which is semantically equivalent (*i.e.* promises are fulfilled by the same *Do*'s or decompositions thereof) due to the fact that M is quasi injective and hence violates the minimality of the original decomposition $\chi^{-1}(P)$. The case where both p_1 and p_2 are mapped onto *Null* is similar to the one presented above, whereby we only merge p_1, p_2. Consequently, the size of M is bounded by $|S_p|\left(|S_d|+1\right)$ as had to be proven. Note that we only consider the bound on minimal mappings (minimality with respect to the decompositions). The argument above shows that this does not exclude reasonable mappings.

NP-hardness: We will reduce the problem of SUBSET-SUM (SS) with positive integers to our problem in polynomial time in order to prove NP-hardness. This corresponds to deciding, given a set S of positive integers and an integer c, if there exists $S' \subseteq S$ such that $\sum_{e_i \in S'} e_i = c$.

Given an instance of SS, we must then provide an instance of our problem such that its solution provides an answer to SS. Let $S_p = \{Promise(A, B, \alpha_0, 0)\}$, and $S_d = \{Do(A, B, \beta_j, 0) \mid j \in S\}$, where α_i and β_j are dummy action symbols of arity zero. We fix the replaceability relation $\risingdotseq$ such that it states that $\{\alpha_0\} \risingdotseq D$ if and only if $D = \{\alpha_i \mid i \in S\}$ and $\sum_{\alpha_i \in D} i = c$. Next, $\phi(\alpha_i, \beta_j) = 0$ if and only if $i = j$ and ∞ otherwise. for $i, j \in S$. Lastly, let $k = 0$.

This transformation yields the desired results, since an enactment mapping M such that $\mathit{degFulfill}_M(S_p) > 0$ exists if and only if S_p can be decomposed into a set of *Promise* atoms that represent a subset of S that sums to c. If this is not possible, then by Definition 11, $\mathit{degFulfill}_M(S_p) = 0$. Lasly, note that this reduction can be done in polynomial time. □

We conclude this section by stating some propositions that characterize the degree of fulfillment introduced in Definition 11. We first show that the overall degree of fulfillment does not depend on the reference time point t_{now}, and hence gives justification for our notation which leaves the time point t_{now} implicit with the context.

Proposition 3. *The overall degree of fulfillment, $\mathit{degFulfill}_M(S_p)$, is independent of the reference time point t_{now},* i.e.*, evaluating $\mathit{degFulfill}_M(S_p)$ w.r.t. two reference time points t^1_{now} and t^2_{now} such that $\forall P_i \in S_p : T_i \leq t^1_{now}, t^2_{now}$ yields the same value.*

Proof. Let M be a fixed mapping between promise and do decompositions and let S_p be a set of promises. Let t^1_{now} and t^2_{now} be two time points such that $\forall P_i \in S_p : T_i \leq t^1_{now}, t^2_{now}$. Then we have:

$$\begin{aligned}
\mathit{degFulfill}_M(S_p) &= \frac{\sum_{P_i \in S_p} \gamma^{\mathit{diff}(t^1_{now}, T_i)} \mathit{degFulfill}(P_i, S_d)}{\sum_{P_i \in S_p} \gamma^{\mathit{diff}(t^1_{now}, T_i)}} \\
&= \frac{\sum_{P_i \in S_p} \gamma^{t^1_{now} - T_i} \mathit{degFulfill}(P_i, S_d)}{\sum_{P_i \in S_p} \gamma^{t^1_{now} - T_i}} \\
&= \frac{\gamma^{t^1_{now}} \sum_{P_i \in S_p} \gamma^{-T_i} \mathit{degFulfill}(P_i, S_d)}{\gamma^{t^1_{now}} \sum_{P_i \in S_p} \gamma^{-T_i}} \\
&= \frac{\sum_{P_i \in S_p} \gamma^{-T_i} \mathit{degFulfill}(P_i, S_d)}{\sum_{P_i \in S_p} \gamma^{-T_i}} \\
&= \frac{\gamma^{t^2_{now}}}{\gamma^{t^2_{now}}} \frac{\sum_{P_i \in S_p} \gamma^{-T_i} \mathit{degFulfill}(P_i, S_d)}{\sum_{P_i \in S_p} \gamma^{-T_i}} \\
&= \frac{\sum_{P_i \in S_p} \gamma^{t^2_{now} - T_i} \mathit{degFulfill}(P_i, S_d)}{\sum_{P_i \in S_p} \gamma^{t^2_{now} - T_i}} \\
&= \frac{\sum_{P_i \in S_p} \gamma^{\mathit{diff}(t^2_{now}, T_i)} \mathit{degFulfill}(P_i, S_d)}{\sum_{P_i \in S_p} \gamma^{\mathit{diff}(t^2_{now}, T_i)}}
\end{aligned}$$

□

The following result shows how the overall degree of fulfillment can be incrementally computed.

Proposition 4. *Let S_d be a set of promises and let $\rho = \mathit{degFulfill}_M(S_p)$ denotes its overall degree of fulfillment. Suppose $\tilde{S_p} = S_p \cup \{P\}$ and $\mathit{degFulfill}_M(P)$ denotes the degree of fulfillment of P according to M. Then we have:*

$$\mathit{degFulfill}_M(\tilde{S_p}) = \frac{\rho\tau + \gamma^{\mathit{diff}(t_{now}, \mathit{time}(P))} \mathit{degFulfill}_M(P)}{\tau + \gamma^{\mathit{diff}(t_{now}, \mathit{time}(P))}}$$

where τ is the denominator of the degree formula, i.e. $\tau = \sum_{P_i \in S_p} \gamma^{\mathit{diff}(t_{now}, T_i)}$.

Proof. Starting from the definition of $\mathit{degFulfill}_M(\tilde{S_p})$, we have the following derivation:

$$\mathit{degFulfill}_M(\tilde{S_p}) = \frac{\sum_{P_i \in \tilde{S_p}} \gamma^{\mathit{diff}(t_{now}, T_i)} \mathit{degFulfill}_M(P_i))}{\sum_{P_i \in \tilde{S_p}} \gamma^{\mathit{diff}(t_{now}, T_i)}}$$

$$= \frac{\sum_{P_i \in S_p} \gamma^{\mathit{diff}(t_{now}, T_i)} \mathit{degFulfill}_M(P_i)) + \gamma^{\mathit{diff}(t^1_{now}, \mathit{time}(P))} \mathit{degFulfill}_M(P)}{\sum_{P_i \in S_p} \gamma^{\mathit{diff}(t_{now}, T_i)} + \gamma^{\mathit{diff}(t^1_{now}, \mathit{time}(P))}}$$

$$= \frac{\rho\tau + \gamma^{\mathit{diff}(t_{now}, \mathit{time}(P))} \mathit{degFulfill}_M(P)}{\tau + \gamma^{\mathit{diff}(t^1_{now}, \mathit{time}(P))}}$$

since $\mathit{degFulfill}_M(S_p) = \frac{\rho}{\tau}$ where τ is the denominator of the degree formula, *i.e.*, $\tau = \sum_{P_i \in S_p} \gamma^{\mathit{diff}(t_{now}, T_i)}$. □

The following result shows that the degree of fulfillment of a single promise changes by a constant factor for different weightings of time in the particular distance measure introduced in Proposition 1.

Proposition 5. *Let $\phi_\mu(P, D) = \ell_\mu \times |T_2 - T_1| + \delta^A_\alpha(a_1, a_2)$, for $P = \mathit{Promise}(B, A, a_1, T_1)$ and $D = \mathit{Do}(B, A, a_2, T_2)$, be distance measures similar to the one defined in Proposition 1 where $\ell_\mu \geq 0$ are a set of real numbers to weight time delays. Let M be the fixed mapping between promise and* *do decompositions as before. Let degFulfill$_M^{\phi_\mu}(P)$ denote the degree of fulfillment for a single promise P with respect to the distance measure ϕ_μ. Then we have:*

$$\frac{\textit{degFulfill}_M^{\phi_k}(P)}{\textit{degFulfill}_M^{\phi_j}(P)} = e^{-(\ell_k-\ell_j)\sum_{P_i \in D_{\{P\}}} \omega_i |\textit{time}(P_i)-\textit{time}(M(P_i))|}$$

Proof. We start from the deifinitions of *degFulfill*$_M^{\phi_k}(P)$ and *degFulfill*$_M^{\phi_j}(P)$; we then have the following derivation:

$$\frac{\textit{degFulfill}_M^{\phi_k}(P)}{\textit{degFulfill}_M^{\phi_j}(P)} = \frac{e^{-\sum_{P_i \in D_{\{P\}}} \omega_i \phi_k(P_i, M(P_i))}}{e^{-\sum_{P_i \in D_{\{P\}}} \omega_i \phi_j(P_i, M(P_i))}} =$$

$$= e^{-\sum_{P_i \in D_{\{P\}}} \omega_i(\phi_k(P_i,M(P_i)) - \phi_j(P_i,M(P_i)))} =$$

$$= e^{-\sum_{P_i \in D_{\{P\}}} \omega_i((\ell_k-\ell_j)\times|\textit{time}(P_i)-\textit{time}(M(P_i))|+d-d)}$$

where $d = \delta_\alpha^A(a_1, a_2)$; continuing with the derivation:

$$e^{-\sum_{P_i \in D_{\{P\}}} \omega_i((\ell_k-\ell_j)\times|\textit{time}(P_i)-\textit{time}(M(P_i))|)} =$$

$$= ke^{-(\ell_k-\ell_j)\sum_{P_i \in D_{\{P\}}} \omega_i|\textit{time}(P_i)-\textit{time}(M(P_i))|} \quad \square$$

The next result shows how linear changes in the distance function on actions impact the degree of fulfillment.

Proposition 6. *Let $\phi_1(P, D) = \ell \times |T_2 - T_1| + \delta_\alpha^A(a_1, a_2)$ and $\phi_2(P, D) = \ell \times |T_2 - T_1| + \lambda\delta_\alpha^A(a_1, a_2)$, for $P =$ Promise(B, A, a_1, T_1) and $D =$ Do(B, A, a_2, T_2), be two distance measures. Then we have*

$$\frac{\textit{degFulfill}_M^{\phi_2}(P)}{\textit{degFulfill}_M^{\phi_1}(P)} = e^{-(\lambda-1)\sum_{P_i \in D_{\{P\}}} \omega_i \delta_\alpha^A(\textit{action}(P_i),\textit{action}(M(P_i)))}$$

Proof. We start from the deifinitions of *degFulfill*$_M^{\phi_1}(P)$ and *degFulfill*$_M^{\phi_2}(P)$; we then have the following derivation:

$$\frac{\textit{degFulfill}_M^{\phi_2}(P)}{\textit{degFulfill}_M^{\phi_1}(P)} = \frac{e^{-\sum_{P_i \in D_{\{P\}}} \omega_i \phi_2(P_i, M(P_i))}}{e^{-\sum_{P_i \in D_{\{P\}}} \omega_i \phi_1(P_i, M(P_i))}} =$$

$$= e^{-\sum_{P_i \in D_{\{P\}}} \omega_i(\phi_2(P_i,M(P_i)) - \phi_1(P_i,M(P_i)))} =$$

$$= e^{-\sum_{P_i \in D_{\{P\}}} \omega_i((\ell-\ell)\times|\textit{time}(P_i)-\textit{time}(M(P_i))|+(1-\lambda)\times d)} =$$

where $d = \delta_\alpha^A(a_1, a_2)$; continuing with the derivation:

$$= e^{-\sum_{P_i \in D_{\{P\}}} \omega_i((1-\lambda)\times\delta_\alpha^A(a_1,a_2))} =$$

$$= e^{-(1-\lambda)\sum_{P_i \in D_{\{P\}}} \omega_i \delta_\alpha^A(a_1,a_2)} \quad \square$$

Application and Experiments

In this section, we use the preceding results to estimate the *future behavior* of agents that made a promise in the past. We show that our fulfillment measures have a strong predictive power (unlike past papers on this topic which did not demonstrate predictive power).

In the rest of this section, we assume that mapping M used in Definition 11 above is fixed *a priori*[2]. We now discuss two different ways in which an agent can reason about the likelihood of the different outcomes that can arise in the presence of a pending promise or set of promises.

[2]This assumption is made without loss of generality, and is needed in order to avoid unwanted variations in the way in which the mapping is done when changing the set of relevant promises taken from historic information. An easy way to fix a mapping is the following: when an agent performs an action, it merely states which promise that action is intended to fulfill, partially or completely.

- **FFIP Strategy** (Future fulfillment is identical to the Past). Agent A decides that the likelihood that a certain promise $P^* =$ *Promise(B,A,a,T)* where $T > t_{now}$ will be kept by agent B at a future time T is completely determined by the experiences with past promises. Hence, in this case, we set *FFIP* $=$ *degFulfill*$_M(S_p)$, which simply states that we expect the agent to fulfill its promises to the degree of fulfillment associated with its past promises. The agent is free to choose which promises should be included in this computation, since taking different subsets into account (taking into account the type of promise) may have an impact on how accurate the estimation is.
- **FFLT Strategy** (Future fulfillment is a Linear Trend based on the Past). Agent A evaluating agent B's promise notices that the reliability of B has changed over time. For instance, its reliability at time 1 was r_1, its reliability at time 2 was r_2, and so forth. The reliability at any time t is computed using *degFulfill*$_M(S_p)$ as above. The agent now considers the r_i's as a time series and uses linear regression to predict the value of this time series at time T. This method allows our system to establish a more controlled way in which to penalize an agent that has broken recent promises (even though promises in the distant past were kept well) or to reward an agent that has kept its promises recently (even though it broke promises in the more distant past). Of course, this method can be easily extended to the use of other kinds of regression models such as logistic regression or higher degree polynomial regressions used commonly in statistics; we study the application of these models in an extended version of this paper.

The US Airline On-Time Performance Dataset

As an example application, we use our approach to analyze the reliability of US airlines. The dataset used in this experimental evaluation corresponds to the on-time performance data for over 117 million flights in the US, recorded over a span of 20 years. For each flight, 55 attributes are stored, including flight dates, origin and destination, departure and arrival delays, whether the flight was canceled or diverted, and information about who was responsible for delays and/or cancelations (BTS 2008).

We considered each flight stored in the database to represent both a promise made by the airline to the customer (of departing and arriving on time, without deviating from the agreed on departure and arrival airports) and its enactment. Therefore, we have a single action symbol *fly* of arity 3, *i.e.*, actions are of the form *fly(from, to, depTime)*, while promises and enactments have the form *Promise(A, C, f, arrTime)* and *Do(A, C, f, arrTime)*, respectively, where A is the airline agent, C is the customer agent, f is a *fly* atom, and *arrTime* is the arrival time promised. The information provided in the database for each flight is enough to derive these atoms.

Airline	FFIP	FFLT	Actual	$dist_{FFIP}$	$dist_{FFLT}$
A_1	0.933	0.924	0.924	0.009	0
A_2	0.922	0.911	0.877	0.045	0.034
A_3	0.914	0.909	0.883	0.031	0.026
A_4	0.942	0.936	0.935	0.007	0.001
A_5	0.924	0.918	0.895	0.029	0.023
A_6	0.935	0.927	0.904	0.031	0.023
A_7	0.934	0.926	0.899	0.035	0.027
A_8	0.923	0.907	0.908	0.015	0.001

Table 1: Predictions for 2007: all past data and linear trend

Empirical Results

Out of the airlines that reported on-time performance for their flights, we chose the eight that have reported continuously from 1988 to 2007; this set includes all major US airlines active today, but we will keep their names anonymous in reporting our results.

For these preliminary evaluations, we computed degrees of fulfillment over sets of promises made throughout entire years, in order to avoid seasonal variations (such as increased delays during winter). However, each individual flight made a contribution to the final degree computed, as dictated by Definition 11. We implemented the FFIP and FFLT strategies that an individual traveler or a travel agent could adopt in order to predict the degree of fulfillment that a promise will have when made by a certain airline. Table 1 shows how these strategies performed when trying to predict the degree of fulfillment for the year 2007 based on information of all flights from 1988 to 2006. All degrees of fulfillment reported in these tables were obtained using a distance function ϕ that ignored delays in departures, and used a "step" function for assigning distances regarding arrival delays. This step function is defined as follows: 0.1 for delays up to 15 minutes, 0.2 up to 30 minutes, 0.8 up to 45 minutes, 2.0 up to 60 minutes, and 10.0 for 90 minutes or more. This means, for instance, that a flight arriving 18 minutes late is considered to be fulfilled to a degree of $e^{0.2}$, which is about 0.818. A value of 0.99 was used for γ, t_{now} was set to January 2, 2008, and the unit of time granularity was set to 30 days, meaning that a flight that occured in January of 1988 is 244 time units away, and its weight is $0.99^{244} \approx 0.086$.

Fulfillment Model Construction Time. The time taken to compute these degrees depends linearly on the number of promises, as can be deduced from Definition 11. For example, as a general indication of the time required to perform this computation, all 15.6 million flights for airline a_1 (from 1988 to 2006) were processed at a rate of about 0.18 millisconds per promise. All computations were performed on a computer with an Intel Xeon CPU at 3.4GHz and 32GB of RAM under the Linux Operating System (2.6.9-42.0.10.ELlargesmp kernel); the database engine used was PostgreSQL version 7.4.16.

FFIP and FFLT query processing time. Most computations for the FFIP and FFLT strategies are performed during the model construction time; actual query processing involves a small number of primitive operations (looking up a value, and computing a linear function, respectively), and therefore query processing times are under 1ms.

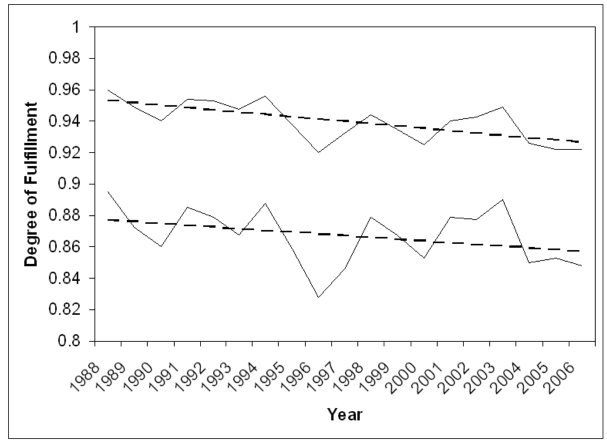

Figure 1: Evolution of degree of fulfillment for a single airline over time, for two different ϕ functions sensitive only to arrival delays.

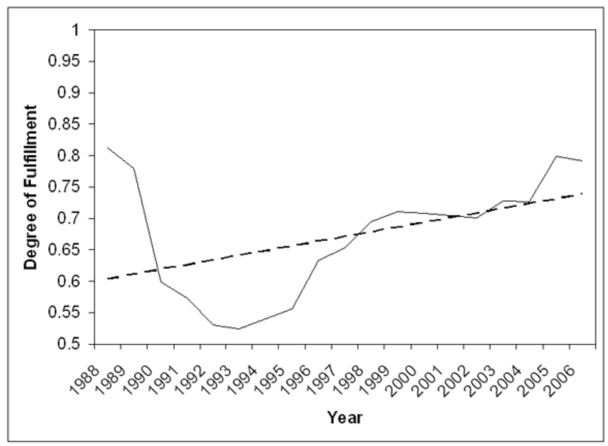

Figure 2: Evolution of degree of fulfillment for a single airline over time, for a ϕ function sensitive to departure delays only; note that the trend is positive, unlike those shown in Figure 1 (note the difference in scale in the x axis w.r.t. that figure).

Accuracy of Predictions. Two things to note from the results in Table 1 are:

1. When comparing what actually happened in terms of an airline's performance (the "Actual" column) and what FFIP and FFLT predicted, the distances between them was relatively small — under 0.03 in almost all cases.
2. When comparing FFIP and FFLT against each other, FFLT was closer to the actual degree of fulfillment in every single case. We then conclude that FFLT is the better algorithm.

Figure 1 presents an example of two trend analyses. The top curve shows how the degree of fulfillment (grouped in years) has evolved for a certain airline from 1988 to 2006, while the dotted line indicates the linear trend it follows. The bottom curve and dotted line correspond to the same analysis, but w.r.t. a different, "harsher" ϕ distance function. This function is similar to the one presented above, but assigns larger distances, namely: 0.1 for delays up to 5 minutes, 0.2 up to 15 minutes, 0.8 up to 25 minutes, 2.0 up to 35 minutes, and 10.0 for 45 minutes or more. For the example flight above which was 18 minutes late, this function declares a degree of fulfillment of $e^{0.8}$, which is about 0.449.

We observe that this yielded overall lower degrees of fulfillment, but the shape of the curve is more or less the same, with each inflection being more exaggerated than its counterpart for the previous function. These changes correspond to what was expected given the change in how the ϕ function was defined.

Finally, Figure 2 shows an example of a trend analysis for the same airline shown in Figure 1. The ϕ distance function used in this case is a different one, which is only sensitive to delays in departures instead of arrivals (the specific values are the same as for the first function presented above). It is interesting to observe that, depending on the perspective of the user, the same airline displays both a downward trend and an upward trend in the evolution of its degree of fulfillment over time. This shows one of the strengths of the framework, *i.e.*, that the user's preferences are taken into account in evaluating the degree of fulfillment of an agent's promises. This is in contrast to, for example, the conclusions that could be obtained by performing a traditional statistical analysis of the frequency of delays such as those presented in (FlightStats 2008).

Discussion

In this section, we would like to discuss several aspects of our framework that we want to highlight, including several limitations that the reader should be aware of.

First of all, the work in this paper assumes that all promises considered have already been made, and therefore "agreed upon" by both parties, *i.e.*, the promise was proposed and accepted. This means that agents cannot simply make promises leaving a lot of room for possible failures (for instance, promising to land at 9AM instead of at 8:15AM), since this kind of behavior will likely not be accepted by the other agent. Furthermore, promises in this framework only involve one action, so a "complex" promise that requires several actions to be performed is actually regarded as a series of promises, each of which will have its associated degree of fulfillment. Lastly, we are focusing only on reasoning based on actions taken towards these promises, and not about beliefs regarding the capabilities of agents to fulfill the promises they have made.

We would like to discuss certain limitations that the framework exhibits. First of all, the axiomatization presented is intended to be a general set of properties that any system should exhibit. Even though this generality can be perceived as a weakness, it lays the groundwork for future research in which assumptions can be made in accordance with specific domains. Another important aspect to note is that the current presentation assumes that the reasoning agent evaluates degrees of fulfillment for *one agent at a time*. This means that, for instance, it will not reason about what the other agent did towards fulfilling its promises with other agents; if this were not the case, Axiom F2 would not always be a desirable property since no action might be preferable to actions benefiting others. Another aspect that may be perceived as a limitation is the fact that degrees of fulfillment are real numbers. This means that it is hard to make the distinction, for instance, between complete fulfillment of a strict subset of promises versus partial fulfillment of all promises in the same set (this is similar to the limitation exhibited by customer satisfaction ratings that merge ratings in different areas into one percentage value). Finally, the distance function between actions as defined here can only be evaluated for atoms that share the same action symbol, which does not allow agents to compare promises with respect to different actions, even though this may be desirable in certain situations.

Conclusions and Future Work

There are numerous applications where an organization or an individual wants to estimate the likelihood that a given organization or individual will fulfill a promise. Manufacturing companies wish to make such estimates in order to assign logistics assets and to plan accordingly. Consumers would like to decide whether one airline is more reliable than another or whether one politician is more likely to honor his promises than another.

In this paper, we have developed axioms that a notion of distance between actions, between promises and performed actions, and between sets of promises and sets of actions must satisfy. These axioms are generic and can be satisfied by many different specific distance functions. We provide an epistemic basis for these axioms and define some specific distance functions.

Based on these ideas, we propose a notion of fulfillment of promises that has many important features. In particular, it accounts for three phenomena not fully handled in previous works. First, we develop a notion of time in studies of promises. Our axioms allow us to penalize late (or early) fulfillments of promises if we so wish, though it does not require such penalties to be imposed. Second, we develop a notion of numeric quantities in promises: delivering 50 of a promised 100 units of a given item can be considered better than nothing and has an impact on our rating of the fulfillment of that promise. Third, we develop notions of replacability where an agent can accept actions in place of promises that are close enough (*e.g.* 50 red balls may be acceptable in place of 50 blue balls). Our framework is rich enough to support a variety of desires on the part of users to customize the notion of promise fulfillment to their needs.

We implemented two methods for using such fulfillment metrics in order to predict the likelihood of fulfillment of a promise in the future by a given agent, and tested them out on a database of flight on-time information for 8 major US airlines over the last 20 years. Our predictions, tested on the degrees of fulfillment for all flights in 2007 operated by these airlines, are highly accurate and can be computed within reasonable amounts of time.

Acknowledgements

The authors gratefully acknowledge funding support for this work provided by the Air Force Office of Scientific Research through the Laboratory for Computational Cultural Dynamics (LCCD) grants AFOSR grants FA95500610405 and FA95500510298, the Army Research Office under grant DAAD190310202, the National Science Foundation under grants 0540216 and 0705587, and ISF. Sarit Kraus is also affiliated with the University of Maryland Institute for Advanced Computer Studies. Any opinions, findings or recommendations in this document are those of the authors and do not necessarily reflect the views of sponsors.

Finally, the authors would like to thank the anonymous reviewers for their many helpful comments.

References

BTS. 2008. US DOT Bureau of Transportation Statistics: http://www.transtats.bts.gov/.

Dellarocas, C. 2006. Reputation mechanisms. *Handbook on Economics and Information Systems*.

Dix, J.; Kraus, S.; and Subrahmanian, V. S. 2006. Heterogeneous temporal probabilistic agents. *ACM Transactions on Computational Logic* 7(1):151–198.

Dondio, P., and Barrett, S. 2007. Presumptive selection of trust evidence. In *Proceedings of the Sixth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, 1–8. New York, NY, USA: ACM.

Ely, J.; Fudenberg, D.; and Levine, D. K. 2002. When is reputation bad? Harvard Institute of Economic Research Working Papers 1962, Harvard - Institute of Economic Research.

Falcone, R., and Castelfranchi, C. 2001. Social trust: a cognitive approach. In *Trust and deception in virtual societies*, 55–90. Norwell, MA, USA: Kluwer Academic Publishers.

FlightStats. 2008. FlightStats: http://www.flightstats.com/.

Sierra, C., and Debenham, J. K. 2007. Information-based agency. In Veloso, M. M., ed., *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 1513–1518.