

# Validation and Verification of Agent Models for Trust: Independent Compared to Relative Trust

Mark Hoogendoorn<sup>1</sup>, S. Waqar Jaffry<sup>1</sup>, and Peter-Paul van Maanen<sup>1,2</sup>

<sup>1</sup> VU University Amsterdam, Department of Artificial Intelligence,  
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands  
{mhoogen, swjaffry}@cs.vu.nl

<sup>2</sup> TNO Human Factors, Department of Cognitive Systems Engineering,  
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands  
peter-paul.vanmaanen@tno.nl

**Abstract.** In this paper, the results of a validation experiment for two existing computational trust models describing human trust are reported. One model uses experiences of performance in order to estimate the trust in different trustees. The second model in addition carries the notion of relative trust. The idea of relative trust is that trust in a certain trustee not solely depends on the experiences with that trustee, but also on trustees that are considered competitors of that trustee. In order to validate the models, parameter adaptation has been used to tailor the models towards human behavior. A comparison between the two models has also been made to see whether the notion of relative trust describes human trust behavior in a more accurate way. The results show that taking trust relativity into account indeed leads to a higher accuracy of the trust model. Finally, a number of assumptions underlying the two models are verified using an automated verification tool.

## 1 Introduction

When considering relations and interaction between agents, the concept of trust is of utmost importance. Within the domain of multi-agent systems, the concept of trust has been a topic of research for many years (e.g., [14,13]). Within this research, the development of models expressing how agents form trust based upon direct experiences with a trustee or information obtained from parties other than the trustee is one of the central themes. Some of these models aim at creating trust models that can be utilized effectively within a software agent environment (e.g., [11]), whereas other models aim to present an accurate model of human trust (e.g., [10,3,6]). The latter type of model can be very useful when developing a personal assistant agent for a human with the awareness of the human's trust in different other agents (human or computer) and him- or herself (trustees). This could for example avoid advising to use particular information sources that are not trusted by the human or could be used to enhance the trust relationship with the personal assistant agent itself.

In order for computational trust models to be usable in real life settings, the validity of these models should be proven first. However, relatively few experiments have been performed that validate the accuracy of computational trust models upon empirical data.

For instance, in [9] an experiment has been conducted whereby the trends in human trust behavior have been analyzed to verify properties underlying trust models developed in the domain of multi-agent systems. However, no attempt was made to fit the model to the trusting behavior of the human.

In this paper, the results of a validation experiment for two computational trust models describing human trust are reported. A trust model taken from [11], which was inspired on the trust model described in [10], has been taken as a baseline model. This model uses experiences of performance in order to estimate the trust in different trustees and is an influential model in the domain of agent systems. The second model which is validated in this study is a model which also carries the notion of relative trust [6]. The idea of relative trust is that trust in a certain trustee not solely depends on the experiences with that trustee, but also with trustees that are considered competitors of that trustee. A comparison between the two models is also made to see whether the notion of relative trust describes human trust behavior in a more accurate way.

The validation process includes a number of steps. First, an experiment with participants has been performed in which trust plays an important role. As a result, empirical data has been obtained, that is usable for validating the two models. One part of the dataset is used to learn the best parameters for the two different trust models. Then these parameters are used to estimate human trust, using the same input as was used to generate the other part of the dataset. Finally, a number of assumptions underlying the two trust models are verified upon the obtained dataset using an automated verification tool. These assumptions are useful to verify whether the humans indeed exhibit the patterns that are used as a basis for the development of trust models.

This paper is organized as follows. First, the two trust models that have been used in this study are explained in Section 2. The experimental method is explained in Section 3. Thereafter, the results of the experiment in terms of model validation and verification are described in Section 4. Finally, Section 5 is a discussion.

## 2 Agent Models for Trust

In this section the two types of trust models which are subject of validation are described. In Section 2.1 a model is explained that estimates human trust in one trustee independent of the trust in other trustees. In contrast, in Section 2.2 a model is described for which this relative dependency actually is important.

### 2.1 Independent Trust Model

This section describes the independent trust model [11,10]. In this model trustees are considered rational and are therefore thought of having no bias to calculate trust. Trust is based on experiences and there is a certain decay of trust.

For the present study, it is assumed that a set of trustees  $\{S_1, S_2, \dots, S_n\}$  is available that can be selected to give particular advice at each time step. Upon selection of one of the trustees ( $S_i$ ), an experience is passed back indicating how well the trustee performed. This experience ( $E_i(t)$ ) is a number on the interval  $[-1, 1]$ . Hereby,  $-1$  expresses a negative experience,  $0$  is a neutral experience and  $1$  a positive experience. There is also a decay parameter  $\lambda_i$  in the model, for which holds that  $0 \leq \lambda_i \leq 1$ .

Given the above, trust now can be calculated by means of the following formula:

$$T_i(t) = T_i(t-1) \cdot \lambda_i + \left(1 - \left(\frac{E_i(t)+1}{2}\right)\right) \cdot (1 - \lambda_i)$$

The independent trust is calculated for each trustee. Note that the experience is mapped to the domain  $[0, 1]$  in this equation. Eventual reliance decisions are made by determining the maximum of the independent trust over all trustees. For more details on the rationale behind the formula, see [11,10].

## 2.2 Relative Trust Model

This section describes the relative trust model [6]. In this model trustees are considered competitors, and the human trust in a trustee depends on the relative experiences with the trustee to the experiences from the other trustees. The model defines the total trust of the human as the difference between positive trust and negative trust (distrust) on the trustee. The model includes several parameters representing human characteristics including trust flexibility  $\beta_i$  (measuring the change in trust on each new experience), decay  $\gamma_i$  (decay in trust when there is no experience) and autonomy  $\eta_i$  (dependence of the trust calculation considering other options). The model parameters  $\beta_i$ ,  $\gamma_i$  and  $\eta_i$  have values from the interval  $[0, 1]$ .

As mentioned before, the model is composed of two models: one for positive trust, accumulating positive experiences, and one for negative trust, accumulating negative experiences. Both negative and positive trust are represented by a number between  $[0, 1]$ . The human's total trust  $T_i(t)$  in  $S_i$  is the difference in positive and negative trust in  $S_i$  at time point  $t$ , which is a number between  $[-1, 1]$ , where  $-1$  and  $1$  represent the minimum and maximum values of trust, respectively. The human's initial total trust in  $S_i$  at time point  $0$  is  $T_i(0)$ , which is the difference in initial trust  $T_i^+(0)$  and distrust  $T_i^-(0)$  in  $S_i$  at time point  $0$ .

As a differential equation the change in positive and negative trust over time is described in the following manner [8]:

$$\begin{aligned} \frac{dT_i^+(t)}{dt} &= E_i(t) \cdot \frac{(E_i(t)+1)}{2} \cdot \beta_i \cdot \\ &\quad \left( \eta_i \cdot (1 - T_i^+(t)) + (1 - \eta_i) \cdot \right. \\ &\quad \left. (\tau_i^+(t) - 1) \cdot T_i^+(t) \cdot (1 - T_i^+(t)) \right) - \\ &\quad \gamma_i \cdot T_i^+(t) \cdot (1 + E_i(t)) \cdot (1 - E_i(t)) \\ \frac{dT_i^-(t)}{dt} &= E_i(t) \cdot \frac{(E_i(t)-1)}{2} \cdot \beta_i \cdot \\ &\quad \left( \eta_i \cdot (1 - T_i^-(t)) + (1 - \eta_i) \cdot \right. \\ &\quad \left. (\tau_i^-(t) - 1) \cdot T_i^-(t) \cdot (1 - T_i^-(t)) \right) - \\ &\quad \gamma_i \cdot T_i^-(t) \cdot (1 + E_i(t)) \cdot (1 - E_i(t)) \end{aligned}$$

In these equations,  $E_i(t)$  is the experience value given by  $S_i$  at time point  $t$ .

Furthermore,  $\tau_i^+(t)$  and  $\tau_i^-(t)$  are the human's relative positive and negative trust in  $S_i$  at time point  $t$ , which is the ratio of the human's positive or negative trust in  $S_i$  to the average human's positive or negative trust in all trustees at time point  $t$  defined as follows:

$$\tau_i^+(t) = \frac{T_i^+(t)}{\left(\frac{\sum_{j=1}^n T_j^+(t)}{n}\right)}$$

and

$$\tau_i^-(t) = \frac{T_i^-(t)}{\left(\frac{\sum_{j=1}^n T_j^-(t)}{n}\right)}$$

Finally, the total change in trust can be calculated as follows (using which the new trust value can easily be calculated):

$$\frac{dT_i(t)}{dt} = \frac{dT_i^+(t)}{dt} - \frac{dT_i^-(t)}{dt}$$

Similarly as for the independent trust model, the trustee with the highest trust value is relied upon.

### 3 Method

In this section the experimental methodology is explained. In Section 3.1 the participants are described. In Section 3.2 an overview of the used experimental environment is given. Thereafter, the procedure of the experiment is explained in four stages: In Sections 3.3, 3.4, 3.5 and 3.6, the procedures of data collection, parameter adaptation, model validation and verification are explained, respectively. The results of the experiment are given in Section 4.

#### 3.1 Participants

18 Participants (eight male and ten female) with an average age of 23 ( $SD = 3.8$ ) were found to be willing to participate in the experiment as paid volunteers. Non-color blinded participants were selected. All were experienced computer users, with an average of 16.2 hours of computer usage each week ( $SD = 9.32$ ).

#### 3.2 Task

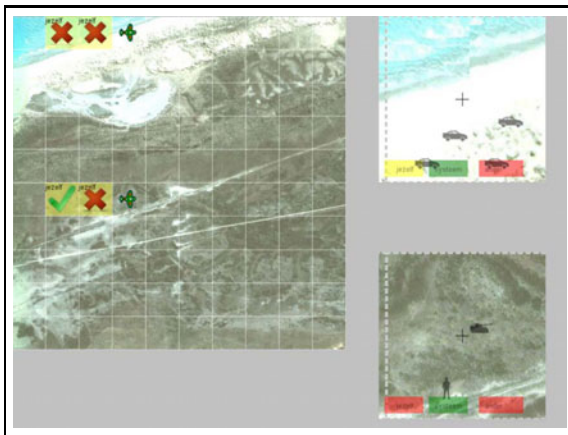
The experimental task was a classification task in which two participants on two separate personal computers had to classify geographical areas according to specific criteria as areas that either needed to be attacked, helped or left alone by ground troops. The participants needed to base their classification on real-time computer generated video images that resembled video footage of real unmanned aerial vehicles (UAVs). On the camera images, multiple objects were shown. There were four kinds of objects:

civilians, rebels, tanks and cars. The identification of the number of each of these object types was needed to perform the classification. Each object type had a score (either  $-2$ ,  $-1$ ,  $0$ ,  $1$  or  $2$ , respectively) and the total score within an area had to be determined. Based on this total score the participants could classify a geographical area (i.e., attack when above 2, help when below  $-2$  or do nothing when in between). Participants had to classify two areas at the same time and in total 98 areas had to be classified. Both participants did the same areas with the same UAV video footage.

During the time a UAV flew over an area, three phases occurred: The first phase was the advice phase. In this phase both participants and a supporting software agent gave an advice about the proper classification (attack, help, or do nothing). This means that there were three advices at the end of this phase. It was also possible for the participants to refrain from giving an advice, but this hardly occurred. The second phase was the reliance phase. In this phase the advices of both the participants and that of the supporting software agent were communicated to each participant. Based on these advices the participants had to indicate which advice, and therefore which of the three trustees (self, other or software agent), they trusted the most. Participants were instructed to maximize the number of correct classifications at both phases (i.e., advice and reliance phase). The third phase was the feedback phase, in which the correct answer was given to both participants. Based on this feedback the participants could update their internal trust models for each trustee (self, other, software agent).

In Figure 1 the interface of the task is shown. The map is divided in  $10 \times 10$  areas. These boxes are the areas that were classified. The first UAV starts in the top left corner and the second one left in the middle. The UAVs fly a predefined route so participants do not have to pay attention to navigation. The camera footage of the upper UAV is positioned top right and the other one bottom right.

The advice of the self, other and the software agent was communicated via dedicated boxes below the camera images. The advice to attack, help, or do nothing was communicated by red, green and yellow, respectively. On the overview screen on the left, feedback was communicated by the appearance of a green tick or a red cross. The



**Fig. 1.** Interface of the task

reliance decision of the participant is also shown on the overview screen behind the feedback (feedback only shown in the feedback phase). The phase depicted in Figure 1 was the reliance phase before the participant indicated his reliance decision.

### 3.3 Data Collection

During the above described experiment, input and output were logged using a server-client application. The interface of this application is shown in Figure 2. Two other client machines, that were responsible for executing the task as described in the previous subsection, were able to connect via a local area network to the server, which was responsible for logging all data and communication between the clients. The interface shown in Figure 2 could be used to set the client's IP-addresses and ports, as well as several experimental settings, such as how to log the data. In total the experiment lasted approximately 15 minutes per participant.

The screenshot shows a software interface with four tabs: 'Connect', 'Start', 'Tune', and 'Validate'. The 'Connect' tab is active. Below the tabs are several input fields and checkboxes:

Field Name	Value
Participant 1 number	1
Participant 2 number	2
Number of operators	2
Model frequency	100
Tuning increment	0.01
Gamemaker frequency	100
Client 1 hostname/ip	localhost
Client 2 hostname/ip	localhost
Client 1 port (in)	5402
Client 2 port (in)	5404
Client 1 port (out)	5403
Client 2 port (out)	5405
Dump comments in console	<input checked="" type="checkbox"/>
Dump comments in file	<input checked="" type="checkbox"/>
Use dummy data	<input type="checkbox"/>
Generate Trace	<input type="checkbox"/>

**Fig. 2.** Interface of the application used for gathering validation data (Connect), for parameter adaptation (Tune) and validation of the trust models (Validate)

Experienced performance feedback of each trustee and reliance decisions of each participant were logged in temporal order for later analysis. During the feedback phase the given feedback was translated to a penalty of either 0, .5 or 1, representing a good, neutral or poor experience of performance, respectively. This directly maps to the value  $\frac{E_i(t)+1}{2}$  in the trust models. During the reliance phase the reliance decisions were translated to either 0 or 1 for each trustee  $S_i$ , which represented that one relied or did not rely on  $S_i$ .

### 3.4 Parameter Adaptation

The data collection described in Section 3.3 was repeated twice on each group of two participants, called condition 1 and condition 2, respectively. The data from one of the conditions was used for parameter adaptation purposes for both models, and the data from the other condition for model validation (see Section 3.5). This process of parameter adaptation and validation was balanced over conditions, which means that condition 1 and condition 2 switch roles, so condition 1 is initially used for parameter adaptation and condition 2 for model validation, and thereafter condition 2 is used for parameter adaptation and condition 1 for model validation (i.e. cross-validation). Both the parameter adaptation and model validation procedure was done using the same application as was used for gathering the empirical data. The interface shown in Figure 2 could also be used to alter validation and adaptation settings, such as the granularity of the adaptation.

The number of parameters of the models presented in Section 2 to be adapted for each model and each participant suggest that an exhaustive search [8] for the optimal parameters is feasible. This means that the entire parameter search space is explored to find a vector of parameter settings resulting in the maximum accuracy (i.e., the amount of overlap between the model's predicted reliance decisions and the actual human reliance decisions) for each of the models and each participant. The corresponding code of the implemented exhaustive search method is shown in Algorithm 1.

---

#### Algorithm 1. ES-PARAMETER-ADAPTATION( $E, R_H$ )

---

```

1.  $\delta_{\text{best}} \leftarrow \infty$ 
2.  $X \leftarrow \mathbf{0}$ 
3. for all parameters  $x$  in parameter vector  $X$  do
4.   for all settings of  $x$  do
5.      $\delta_X \leftarrow 0$ 
6.     for all time points  $t$  do
7.        $e \leftarrow E(t)$ 
8.        $r_M \leftarrow R_M(e, X)$ 
9.        $r_H \leftarrow R_H(e)$ 
10.      if  $r_M \neq r_H$  then
11.         $\delta_X \leftarrow \delta_X + 1$ 
12.      end if
13.    end for
14.    if  $\delta_X < \delta_{\text{best}}$  then
15.       $X_{\text{best}} \leftarrow X$ 
16.       $\delta_{\text{best}} \leftarrow \delta_X$ 
17.    end if
18.  end for
19. end for
20. return  $X_{\text{best}}$ 

```

---

In this algorithm,  $E(t)$  is the set of experiences (i.e., performance feedback) at time point  $t$  for all trustees,  $R_H(e)$  is the actual reliance decision the participant made (on either one of the trustees) given a certain experience  $e$ ,  $R_M(e, X)$  is the predicted reliance

decision of the trust model  $M$  (either independent or relative) given an experience  $e$  and candidate parameter vector  $X$  (reliance on either one of the trustees),  $\delta_X$  is the distance between the estimated and actual reliance decisions given a certain candidate parameter vector  $X$ ,  $\delta_{\text{best}}$  is the distance resulting from the best parameter vector  $X_{\text{best}}$  found so far. The best parameter vector  $X_{\text{best}}$  is returned when the algorithm finishes. This parameter adaptation procedure was implemented in C#.

If for Algorithm 1 the number of parameters is  $\mu$ ,  $\Gamma$  the granularity for each parameter,  $N$  the number of trustees and  $B$  the number of reliance decisions (i.e., time points) made by the human, then the worst case complexity of the algorithm is expressed as  $O(10^{\mu\Gamma}BN)$ . The complexity also depends on  $N$ , since  $R_M(e, X)$  results in a calculation of trust values over all trustees. For the independent trust model it holds that  $\mu = 1$  (i.e., the parameter  $\lambda_i$ ) and for the relative trust model  $\mu = 3$  (i.e., the three parameters  $\beta_i$ ,  $\gamma_i$  and  $\eta_i$ ). In the current experiment it furthermore holds that  $\Gamma = 2$  (i.e., steps of .01),  $N = 3$  (the two humans and the software agent) and  $B = 98$  (the total of classified geographical areas). This means that  $2.94 \cdot 10^4$  computation steps are needed for the independent trust model and  $2.94 \cdot 10^8$  for the relative trust model, which took on average 31 milliseconds for the first, and 3 minutes and 20 seconds computation time for the second model.<sup>1</sup>

### 3.5 Validation

In order to validate the two models described in Section 2, the measurements of experienced performance feedback were used as input for the models and the output (predicted reliance decisions) of the models was compared with the the actual reliance decisions of the participant. The overlap of the predicted and the actual reliance decisions was a measure for the accuracy of the models. The results are in the form of dynamic accuracies over time, average accuracy per condition (1 or 2) and per trust model (independent or relative). A comparison between the averages per model and the interaction effect between condition role allocation (i.e., parameter adaptation either in condition 1 or 2) and model type, is done using a repeated measures analysis of variance (ANOVA).

### 3.6 Verification

Next to a validation using the accuracy of the prediction using the models, another approach has been used to validate the assumptions underlying existing trust models. The idea is that properties that form the basis of trust models are verified against the empirical results obtained within the experiment. In order to conduct such an automated verification, the properties have been specified in a language called Temporal Trace Language (TTL) [1] that features a dedicated editor and an automated checker. The language TTL is explained first, followed by an expression of the desired properties related to trust.

---

<sup>1</sup> This was on an ordinary PC with an Intel(R) Core(TM)2 Quad CPU @2.40 GHz inside. Note that  $31 \cdot \frac{2.94 \cdot 10^8}{2.94 \cdot 10^4}$  milliseconds = 5.17 minutes  $\neq$  3.33 minutes computation time. This is due to a fixed initialization time of on average 11 ms for both models.



**Temporal Trace Language (TTL).** The predicate logical temporal language TTL supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to states of the world, time points and traces, i.e., trajectories of states over time. In addition, dynamic properties are temporal statements that can be formulated with respect to traces based on the state ontology *Ont* in the following manner. Given a trace  $\gamma$  over state ontology *Ont*, the state in  $\gamma$  at time point  $t$  is denoted by  $\text{state}(\gamma, t)$ . These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate  $\models$ , i.e.,  $\text{state}(\gamma, t) \models p$  denotes that state property  $p$  holds in trace  $\gamma$  at time  $t$ . Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\forall$  and  $\exists$ . For more details on TTL, see [1].

**Properties for Trust Models.** Within the literature on trust, a variety of properties have been expressed concerning the desired behavior of trust models. In many of these properties, the trust values are explicitly referred to, for instance in the work of [10] characteristics of trust models have been defined (e.g., monotonicity and positive trust extension upon positive experiences). In this paper however, the trust function is subject of validation and hence, cannot be taken as a basis. Therefore, properties are expressed on an external basis, solely using the information which has been observed within the experiment to see whether these behaviors indeed comply to the desired behavior of the trust models. This information is then limited to the experiences that are received as an input and the choices that are made by the human that are generated as output. The properties from [7] are taken as a basis for these properties. Essentially, the properties indicate the following desired behavior of human trust:

1. Positive experiences lead to higher trust
2. Negative experiences lead to lower trust
3. Most trusted trustee is selected

As can be seen, the properties also use the intermediate state of trust. In order to avoid this, it is however possible to combine these properties into a single property that expresses a relation between the experiences and the selection (i.e., the above items 1 + 3 and 2 + 3). Two of these properties are shown below. In addition, a property is expressed which specifies the notion of relativity in the experiences and the resulting selection of a trustee. The first property expresses that a trustee that gives the absolute best experiences during a certain period is eventually selected at least once within, or just after that particular period, and is shown below.

**P1(min\_duration, max\_duration, max\_time): Absolute more positive experiences results in selection**

If a trustee  $a_1$  always gives more positive experiences than all other trustees during a certain period with minimal duration *min\_duration* and maximum duration *max\_duration*, then this trustee  $a_1$  is selected at least once during the period [*min\_duration*, *max\_duration* + *max\_time*].

Formal:

$$\begin{aligned} & P1(\text{min\_duration}:\text{DURATION}, \text{max\_duration}:\text{DURATION}, \text{max\_delay}:\text{DURATION}) \equiv \\ & \forall \gamma:\text{TRACE}, t_{\text{start}}, t_{\text{end}}:\text{TIME}, a:\text{TRUSTEE} \\ & [ [ t_{\text{end}} - t_{\text{start}} \geq \text{min\_duration} \ \& \ t_{\text{end}} - t_{\text{start}} \leq \text{max\_duration} \ \& \\ & \text{absolute\_highest\_experiences}(\gamma, a, t_{\text{start}}, t_{\text{end}}) \\ & \Rightarrow \text{selected}(\gamma, a, t_{\text{start}}, t_{\text{end}}, \text{max\_delay}) \end{aligned}$$

where

$$\begin{aligned} & \text{absolute\_highest\_experiences}(\gamma:\text{TRACE}, a:\text{TRUSTEE}, t_{\text{start}}:\text{TIME}, t_{\text{end}}:\text{TIME}) \equiv \\ & \forall t:\text{TIME}, r_1, r_2:\text{REAL}, a_2:\text{TRUSTEE} \neq a \\ & [ [ t \geq t_{\text{start}} \ \& \ t < t_{\text{end}} \ \& \ \text{state}(\gamma, t) \models \text{trustee\_gives\_experience}(a, r_1) \ \& \\ & \text{state}(\gamma, t) \models \text{trustee\_gives\_experience}(a_2, r_2) ] \Rightarrow r_2 < r_1 ] \end{aligned}$$

$$\begin{aligned} & \text{selected}(\gamma:\text{TRACE}, a:\text{TRUSTEE}, t_{\text{start}}:\text{TIME}, t_{\text{end}}:\text{TIME}, z:\text{DURATION}) \equiv \\ & \exists t:\text{TIME} [ t \geq t_{\text{start}} \ \& \ t < t_{\text{end}} + z \ \& \ \text{state}(\gamma, t) \models \text{trustee\_selected}(a) ] \end{aligned}$$

The second property, P2, specifies that the trustee which gives more positive experiences on average during a certain period is at least selected once within or just after that period.

**P2(min\_duration, max\_duration, max\_delay, higher\_exp): Average more positive experiences results in selection**

If a trustee  $a_1$  on average gives the most positive experiences (on average more than higher\_exp better than the second best) during a period with minimal duration min\_duration and maximum duration max\_duration, then this trustee  $a_1$  is selected at least once during the period [min\_duration, max\_duration+max\_delay].

Formal:

$$\begin{aligned} & P2(\text{min\_duration}:\text{DURATION}, \text{max\_duration}:\text{DURATION}, \text{max\_delay}:\text{DURATION}, \text{higher\_exp}:\text{REAL}) \\ & \equiv \forall \gamma:\text{TRACE}, t_{\text{start}}, t_{\text{end}}:\text{TIME}, a:\text{TRUSTEE} \\ & [ [ t_{\text{end}} - t_{\text{start}} \geq \text{min\_duration} \ \& \ t_{\text{end}} - t_{\text{start}} \leq \text{max\_duration} \ \& \\ & \text{average\_highest\_experiences}(\gamma, a, t_{\text{start}}, t_{\text{end}}, \text{higher\_exp}) \\ & \Rightarrow \text{selected}(\gamma, a, t_{\text{start}}, t_{\text{end}}, \text{max\_delay}) ] \end{aligned}$$

where

$$\begin{aligned} & \text{average\_highest\_experiences}(\gamma:\text{TRACE}, a:\text{TRUSTEE}, t_{\text{start}}:\text{TIME}, t_{\text{end}}:\text{TIME}, \text{higher\_exp}:\text{REAL}) \\ & \equiv \forall t:\text{TIME}, r_1, r_2:\text{REAL}, a_2:\text{TRUSTEE} \neq a \\ & [ t \geq t_{\text{start}} \ \& \ t < t_{\text{end}} \ \& \\ & [ \sum_{\forall t:\text{TIME}} \text{case}(\text{experience\_received}(\gamma, a, t, t_{\text{start}}, t_{\text{end}}, e), e, 0) > \\ & (\sum_{\forall t:\text{TIME}} \text{case}(\text{experience\_received}(\gamma, a, t, t_{\text{start}}, t_{\text{end}}, e), e, 0)) + \text{higher\_exp} * t_{\text{end}} - t_{\text{start}}) \\ & ] ] \end{aligned}$$

In the formula above, the case( $p$ ,  $e$ , 0) operator evaluates to  $e$  in case property  $p$  is satisfied and to 0 otherwise.

$$\begin{aligned} & \text{experience\_received}(\gamma:\text{TRACE}, a:\text{TRUSTEE}, t:\text{TIME}, t_{\text{start}}:\text{TIME}, t_{\text{end}}:\text{TIME}, r:\text{REAL}) \equiv \\ & [ \exists r:\text{REAL}, t \geq t_{\text{start}} \ \& \ t < t_{\text{end}} \ \& \ \text{state}(\gamma, t) \models \text{trustee\_gives\_experience}(a, r) ] \end{aligned}$$

The final property concerns the notion of relativity which plays a key role in the models verified throughout this paper. The property expresses that the frequency of selection of a trustee that gives an identical experience pattern during two periods is not identical in case the other trustees give different experiences.

### **P3(interval\_length, min\_difference, max\_time): Relative trust**

If a trustee  $a_1$  gives an identical experience pattern during two periods  $[t_1, t_1 + \text{interval\_length}]$  and  $[t_2, t_2 + \text{interval\_length}]$  and the experiences of at least one other trustee is not identical (i.e., more than  $\text{min\_difference}$  different at each time point), then the selection frequency of  $a_1$  will be different in a period during, or just after the specified interval.

Formal:

$$\begin{aligned} & \text{P3}(\text{interval\_length}:\text{DURATION}, \text{min\_difference}:\text{REAL}, \text{max\_time}:\text{DURATION}) \equiv \\ & \forall \gamma:\text{TRACE}, t_1, t_2:\text{TIME}, a:\text{TRUSTEE} \\ & [ [\text{same\_experience\_sequence}(\gamma, a, t_1, t_2, \text{interval\_length}) \& \\ & \exists a_2:\text{TRUSTEE} \neq a \\ & [\text{different\_experience\_sequence}(\gamma, a, t_1, t_2, \text{min\_difference})] \\ & \Rightarrow \exists i:\text{DURATION} < \text{max\_time} \\ & \sum_{\forall t:\text{TIME}} \text{case}(\text{selected\_option}(\gamma, a, t, t_1 + i, t_1 + i + \text{interval\_length}), 1, 0) / \\ & (1 + \sum_{\forall t:\text{TIME}} \text{case}(\text{trustee\_selected}(\gamma, t, t_1, t_1 + i + \text{interval\_length}), 1, 0)) \neq \\ & \sum_{\forall t:\text{TIME}} \text{case}(\text{selected\_option}(\gamma, a, t, t_2 + i, t_2 + i + \text{interval\_length}), 1, 0) / \\ & (1 + \sum_{\forall t:\text{TIME}} \text{case}(\text{trustee\_selected}(\gamma, t, t_2 + i, t_2 + i + \text{interval\_length}), 1, 0)) \end{aligned}$$

where

$$\begin{aligned} & \text{same\_experience\_sequence}(\gamma:\text{TRACE}, a:\text{TRUSTEE}, t_1:\text{TIME}, t_2:\text{TIME}, x:\text{DURATION}) \equiv \\ & \forall y:\text{DURATION} [ y \geq 0 \& y \leq x \& \exists r:\text{REAL} \\ & [ \text{state}(\gamma, t_1 + y) \models \text{trustee\_gives\_experience}(a, r) \& \\ & \text{state}(\gamma, t_2 + y) \models \text{trustee\_gives\_experience}(a, r) ] ] \end{aligned}$$

$$\begin{aligned} & \text{different\_experience\_sequence}(\gamma:\text{TRACE}, a:\text{TRUSTEE}, t_1:\text{TIME}, t_2:\text{TIME}, x:\text{DURATION}, \\ & \text{min\_difference}:\text{REAL}) \equiv \\ & \forall y:\text{DURATION} [ y \geq 0 \& y \leq x \& \exists r_1, r_2:\text{REAL} \\ & [ \text{state}(\gamma, t_1 + y) \models \text{trustee\_gives\_experience}(a, r_1) \& \\ & \text{state}(\gamma, t_2 + y) \models \text{trustee\_gives\_experience}(a, r_2) \& \\ & |r_1 - r_2| > \text{min\_difference} ] ] \end{aligned}$$

$$\begin{aligned} & \text{trustee\_selected}(\gamma:\text{TRACE}, t:\text{TIME}, t_{\text{start}}:\text{TIME}, t_{\text{end}}:\text{TIME}) \equiv \\ & \exists a:\text{TRUSTEE} [ t \geq t_{\text{start}} \& t < t_{\text{end}} \& \text{state}(\gamma, t) \models \text{trustee\_selected}(a) ] \end{aligned}$$

## **4 Results**

In this section the validation and verification results are given in Sections 4.1 and 4.2, respectively.

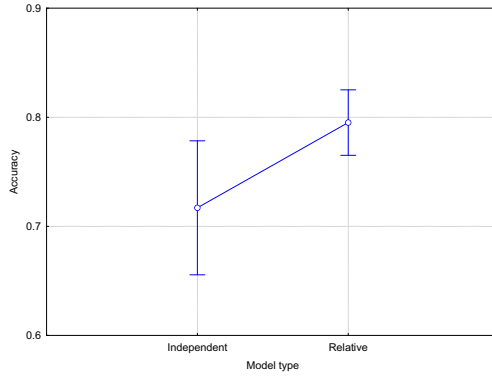


Fig. 3. Main effect of model type for accuracy

### 4.1 Validation Results

From the data of 18 participants, one dataset has been removed due to an error while gathering data. This means that there are 2 (condition role allocations, i.e., parameter adaptation either in condition 1 or 2) times 17 (participants) = 34 data pairs (accuracies for 2 models). Due to a significant Grubbs test, from these pairs 3 outliers were removed. Hence in total 31 pairs were used for the data analysis.

In Figure 3 the main effect of model type (either independent or relative trust) for accuracy is shown. A repeated measures analysis of variance (ANOVA) showed a significant main effect ( $F(1, 29) = 7.60, p < .01$ ). This means that indeed the relative trust model had a higher accuracy ( $M = .7968, SD = .0819$ ) than the independent trust model ( $M = .7185, SD = .1642$ ).

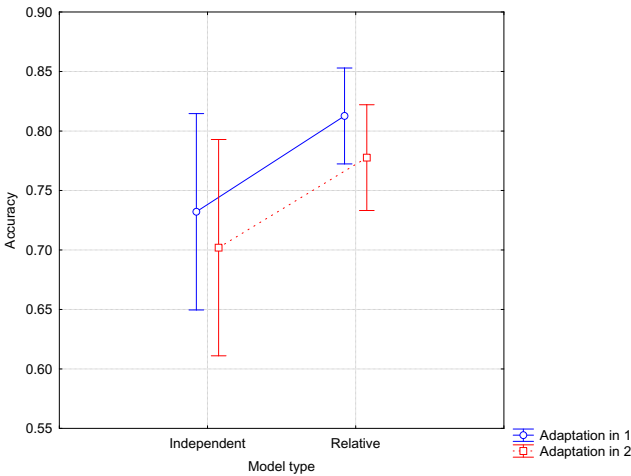


Fig. 4. Interaction effect between condition role allocation and model type on accuracy

Figure 4 shows the possible interaction effect between condition role allocation (parameter adaptation in condition 1 is referred to as adaptation 1 and parameter adaptation in condition 2 is referred to as adaptation 2) and model type (either independent or relative trust) on accuracy. No significant interaction effect was found ( $F(1, 29) = .01$ ,  $p = .93$ ). Hence, no significant learning effect between conditions was found. Cross-validation was not needed to balance the data, but the procedure still produced twice as much data pairs.

## 4.2 Verification Results

The results of the verification of the properties against the empirical traces (i.e., formalized logs of human behavior observed during the experiment) are shown in Table 1. First, the results for properties P1 and P2 are shown. Hereby, the value of `max_duration` has been kept constant at 30 and the `max_time` after which the trustee should be consulted is set to 5. The minimal interval time (`min_duration`) has been varied. Finally, for property P2 the variable `higher_exp` indicating how much higher the experience should be on average compared to the other trustees is set to .5. The results in Table 1 indicate the percentage of traces in which the property holds out of all traces in which the antecedent at least holds once (i.e., at least one sequence with the `min_duration` occurs in the trace). This has been done to avoid a high percentage of satisfaction due to the fact that in some of the traces the antecedent never holds, and hence, the property is always satisfied. The table shows that the percentage of traces satisfying P1 goes up as the minimum duration of the interval during which a trustee gives the highest experience increases. This clearly complies to the ideas underlying trust models as the longer a trustee gives the highest experiences, the higher his trust will be (also compared to the other trustees), and the more likely it is that the trustee will be selected. The second property, counting the average experience and its implication upon the selection behavior of the human, also shows an increasing trend in satisfaction of the property with the duration of the interval during which the trustee on average gives better experiences. The percentages are lower compared to P1 which can be explained by the fact that they might also give some negative experiences compared to the alternatives (whereas they are giving better experiences on average). This could then result in a decrease in the trust value, and hence, a lower probability of being selected.

**Table 1.** Results of verification of property P1 and P2

<code>min_duration</code>	% satisfying P1	% satisfying P2
1	64.7	29.4
2	64.7	29.4
3	86.7	52.9
4	92.3	55.9
5	100.0	58.8
6	100.0	70.6

**Table 2.** Results of verification of property P3

interval_length	% satisfying P3
1	0
2	41.1
3	55.9
4	67.6
5	66.7
6	68.4

The third property, regarding the relativity of trust has also been verified and the results of this verification are shown in Table 2. Here, the traces of the participants have been verified with a setting of `min_difference` to `.5` and `max_time` to `5` and the variable `interval_length` during which at least one trustee shows identical experiences whereas another shows different experiences has been varied. It can be seen that property P3 holds more frequently as the length of the interval increases, which makes sense as the human has more time to perceive the relative difference between the two. Hence, this shows that the notion of relative trust can be seen in the human trustee selection behavior in almost 70% of the cases.

## 5 Discussion and Conclusions

In this paper, an extensive validation study has been performed to show that human trust behavior can be accurately described and predicted using computational trust models. In order to do so, an experiment has been designed that places humans in a setting where they have to make decisions based upon the trust they have in others. In total 18 participants took part in the experiment. The results show that both an independent [11,10] as well as a relative trust model [6] can predict this behavior with a high accuracy (72% and 80%, respectively) by learning on one dataset and predicting the trust behavior for another (cross-validation). Furthermore, it has also been shown that the underlying assumptions of the trust models (and many other trust models) are found in the data of the participants.

Of course, more work on the validation of trust models has been performed. In [9] an experiment has been presented to investigate human trust behavior. Although the underlying assumptions of trust models have to some extent been verified in that paper, no attempt has been made to fit a trust model to the data. Other papers describing the validation of trust models for instance validate the accuracy of trust models describing the propagation of trust through a network (e.g., [5]). In [12] a multidisciplinary, multidimensional model of trust in e-commerce is validated. The model includes four high-level constructs: disposition to trust, institution-based trust, trusting beliefs, and trusting intentions. The proposed model itself does however not describe the formation of trust on such a detailed level as the models used throughout this paper, it presents general relationships between trust measures and these relationships are subject to validation. Gefen and Straub [4] validate a four-dimensional scale of trust in the context

of e-Products and revalidates it in the context of e-Services which shows the influence of social presence on these dimensions of trust, especially benevolence, and its ultimate contribution to online purchase intentions. Again, correlations are found between the concepts of trust that have been distinguished, but no computational model for the formation of trust and the precise prediction thereof is prosed. Finally, in [2] a development-based trust measurement model for buyer-seller relationships is presented and validated against a characteristic-based trust measurement model in terms of its ability to explain certain variables of interest in buyer-seller relationships (long-term relationship orientation, information sharing, behavioral loyalty and future intentions).

Within the domain of agent systems, quite some trust models have been developed, see e.g. [14], [13] for an overview. Although the focus of this paper has been on the validation of two specific trust models, thereby also comparing relative with absolute trust, other trust models can also be validated using the experimental data obtained in combination with parameter estimation. This is part of the future work. Furthermore, other parameter adaptation methods will be explored or extended for the purpose of real-time adaptation, which accounts for human learning. In addition, a personal assistant software agent will be implemented that is able to monitor and balance the functional state of the human in a timely and knowledgeable manner. Also applications in different domains are explorable, such as the military and air traffic control domain.

## Acknowledgments

This research was partly funded by the Dutch Ministry of Defense under progr. no. V929. Furthermore, this research has partly been conducted as part of the FP7 ICT Future Enabling Technologies program of the European Commission under grant agreement no. 231288 (SOCIONICAL). The authors would like to acknowledge Francien Wisse for her efforts to gather the necessary validation data and implementing the experimental task. The authors would also like to thank Tibor Bosse, Jan-Willem Streefkerk and Jan Treur for their helpful comments.

## References

1. Bosse, T., Jonker, C., Meij, L.v.d., Sharpanskykh, A., Treur, J.: Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems* 18, 167–193 (2009)
2. da Costa Hernandez, J.M., dos Santos, C.C.: Development-based trust: Proposing and validating a new trust measurement model for buyer-seller relationships. *Brazilian Administration Review* 7, 172–197 (2010)
3. Falcone, R., Castelfranchi, C.: Trust dynamics: How trust is influenced by direct experiences and by trust itself. In: *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, New York, USA, pp. 740–747 (July 2004)
4. Gefen, D., Straub, D.W.: Consumer trust in b2c e-commerce and the importance of social presence: experiments in e-products and e-services. *Omega* 32, 407–424 (2004)
5. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: *Proceedings of the 13th International Conference on World Wide Web (WWW 2004)*, pp. 403–412. ACM, New York (2004)

6. Hoogendoorn, M., Jaffry, S., Treur, J.: Modeling dynamics of relative trust of competitive information agents. In: Klusch, M., Pěchouček, M., Polleres, A. (eds.) CIA 2008. LNCS (LNAI), vol. 5180, pp. 55–70. Springer, Heidelberg (2008)
7. Hoogendoorn, M., Jaffry, S., Treur, J.: Modelling trust dynamics from a neurological perspective. In: Wang, R., Gu, F. (eds.) Proceedings of the Second International Conference on Cognitive Neurodynamics, ICCN 2009, Advances in Cognitive Neurodynamics II, pp. 523–536. Springer, Heidelberg (2011)
8. Hoogendoorn, M., Jaffry, S.W., Treur, J.: An adaptive agent model estimating human trust in information sources. In: Baeza-Yates, R., Lang, J., Mitra, S., Parsons, S., Pasi, G. (eds.) Proceedings of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009), pp. 458–465 (2009)
9. Jonker, C.M., Schalken, J.J.P., Theeuwes, J., Treur, J.: Human experiments in trust dynamics. In: Jensen, C., Poslad, S., Dimitrakos, T. (eds.) iTrust 2004. LNCS, vol. 2995, pp. 206–220. Springer, Heidelberg (2004)
10. Jonker, C.M., Treur, J.: Formal analysis of models for the dynamics of trust based on experiences. In: Garijo, F.J., Boman, M. (eds.) MAAMAW 1999. LNCS, vol. 1647, pp. 221–232. Springer, Heidelberg (1999)
11. Maanen, P.-P.v., Klos, T., Dongen, K.v.: Aiding human reliance decision making using computational models of trust. In: Proceedings of the Workshop on Communication between Human and Artificial Agents (CHAA 2007), Fremont, California, USA, pp. 372–376. IEEE Computer Society Press, Los Alamitos (2007); Co-located with The 2007 IEEE IAT/WIC/ACM International Conference on Intelligent Agent Technology
12. McKnight, D.H., Choudhury, V., Kacmar, C.: Developing and validating trust measures for e-commerce: An integrative topology. *Information Systems Research* 13(3), 334–359 (2001)
13. Ramchurn, S., Huynh, D., Jennings, N.: Trust in multi-agent systems. *The Knowledge Engineering Review* 19, 1–25 (2004)
14. Sabater, J., Sierra, C.: Review on computational trust and reputation models. *Artificial Intelligence Review* 24, 33–60 (2005)