A Theoretical Model for the Human-IoT Systems Interaction

Alessandro Sapienza
ISTC-CNR,
Rome, Italy
alessandro.sapienza@istc.cnr.it

Rino Falcone
ISTC-CNR,
Rome, Italy
rino.falcone@istc.cnr.it

CEUR-WS.org/Vol-2404/paper14.pdf

***Abstract*— Thanks to the IoT, our life will strongly improve in the next future. However, it is not given that the users will be able to afford all the automation it will offer or that it will be compatible with the users' cognitive attitudes and its actual and real goals. In this paper, we face the question of the IoT from the user point of view. We start analyzing which reasons undermine the acceptance of IoT systems and then we propose a possible solution. The first contribution of this work is the level characterization of the autonomy a user can grant to an IoT device. The second contribution is a theoretical model to deal with users and to stimulate users' acceptance. By the means of simulation, we show how the model works and we prove that it leads the system to an optimal solution.**

Keywords*— *Trust, Internet of Things, Autonomy

I. Introduction

Just in the near future, we expect to have billion of devices connected to the Internet [1]. Here, the main novelty is that this technology is not limited to the classic devices, but involves also objects that are not currently smart. We are going to face unbelievable scenarios; health, education, transport, every aspect of our lives will undergo radical changes, even our own homes will become smart [2]. The fridge could tell us that it is better to throw eggs away because they are no longer fresh, the washing machine could propose a more efficient way to wash clothes, entire buildings could work together to save energy or other resources. This very principle of "connection between things" is the basis of the Internet of Things [3].

While it is true that we have a multitude of extremely useful scenarios, there are also considerable security and privacy issues [4]. Certainly, we are not talking about an apocalyptic prospective, but if in everyday life a hacker is able to block our computer, just think about the damage it could make if it decided to block our home doors. This problem is further enhanced by the heterogeneity of the devices, making it more difficult to control and detect security flaws. If it is already difficult to accept that an object possesses intelligence and can interface with us, the thought that it can revolt against us, causing substantial damage, could make it even more difficult to spread IoT systems.

We then argue that a good way to address this problem is through the concept of trust [5]. The key point is in fact that users do not trust these systems; they do not know them or what they can do. The concept of trust comes spontaneously into play. Thus, we propose a general IoT system able to adapt to the specific user and its disposition towards this technology, with the aim of (1) identifying the acceptance limit the user has and (2) pushing the user to accept more this technology. After describing the theoretical model, we will introduce a possible implementation in a simulative context, with the aim of showing how it works. Being a general model, it can be applied to any IoT system.

The rest of the paper is organized as follows: Section II analyzes the state of the art, pointing out the necessity of a user centric design for IoT systems; Section III provides a theoretical framework for trust, control, and feedback, also showing the computational model we used; Section IV describes how we implemented the model in the simulation of Section V; Section VI comments on the results of the simulation; Section VII concludes the work.

II. Distrust in the IoT

IoT systems represent a wide variety of technologies, so it is not easy to identify in detail the common characteristics and the feature they should possess. However, in a high-level vision, some key aspects often and recurrently come into play.

For sure, a salient topic is that of *security* [6][7][8][9], in the way that computer science means it. A device must be secure, and the reason is clear: if we give the green light to such a pervasive technology, able to enter deeply into every aspect of our life, it is fundamental that there are no security breaches. For instance, even our toaster could be able to steal money from our bank account; we need to be sure that similar scenarios will not happen. Security mainly relies on encryption to solve its problems.

Then *privacy* comes into play. As the device will exchange impressive amounts of information, more than can concretely be processed [10], it is not clear which information will be shared and with whom [11]. We need a new way to deal with privacy, since the classical approach of authentication [12] and policies cannot work properly in such a huge and heterogeneous environment. Facing privacy is necessary, but still not enough.

A third element is *trust*. Usually it is applied to identify trustworthy devices in a network, separating them from the malicious ones [13]. By the way, the authors of [14] provide a review of the use of trust in IoT systems. The same authors identify that trust "helps people overcome perceptions of uncertainty and risk and engages in user acceptance". In fact, when we have autonomous tools able to take various different decisions and these decisions involve our own goals and results, we have to be worried not just about their correct functioning (for each of these decisions) but also about their autonomous behavior and the role it plays for our purposes.

All these components are valid and fundamental to an IoT system. However, a further point should be emphasized. Although an IoT device requires only the connection and interfacing with the outside world to be defined as such, and then the possibility of being addressed and of exchanging information with the outside world, they are not independent systems, but on the contrary these systems continually interact with users, they relate to them in a very strong way: the user is at the center of everything. In fact, reasoning in view of the goals that these objects possess, the common purpose is to make life better for users, be they the inhabitants of a house, a city, patients/doctors in a hospital, or the workers of a facility.

The user becomes the fundamental point in all of this. A technology can be potentially perfect and very useful but, if people do not accept it, each effort is useless and it goes out of use. It is necessary to keep in mind how much the user is willing to accept an IoT system and to what extent he wants to interface with it. We would like to focus on this last point, the concept of “user acceptance”.

As Ghazizadeh [15] says “technology fundamentally changes a person’s role, making the system performance progressively dependent on the integrity of this relationship. In fact, automation will not achieve its potential if not properly adopted by users and seamlessly integrated into a new task structure”.

Furthermore, Miranda et al. [16] talk about Internet of People (IoP). In fact, they reiterate that technology must be integrated into the users’ daily lives, which are right at the center of the system. They focus on the fact that IoT systems must be able to adapt to the user, taking people’s context into account and avoiding user intervention as much as possible. Similarly, Ashraf [17] talks about autonomy in the Internet of things, pointing out that in this context it is necessary to minimize user intervention.

Thus, the acceptance of a new technology seems to be the key point, which is not always obvious. It is not easy for users to understand how a complex technology like this reasons and works. Often it is not clear what it is able to do and how it does it.

So it is true that security, privacy, and trust work together to increase reliance on IoT systems. However, it is necessary to keep users in the center of this discussion.

The reasons why the user may not grant high trust levels are the fears that (a) the task is not carried out in the expected way; (b) that it is not completed at all; or (c) even that damage is produced. These issues become more and more complicated if we think that these devices can operate in a network that has a theoretically infinite number of nodes: we do not know the other devices’ goals or if they will be reliable. We get into a very complex system, difficult to understand and manage.

In short, the overall picture of the functions that they perform is going to complicate a lot. As a whole, the devices have a computational power and a huge amount of data available; they could be able to identify solutions that we had not even imagined, which, however, must ensure that these systems will realize a state of the world coinciding with our expectation. What if our computer decides to shut down because we worked too much? Surely, we talk about tasks that have their usefulness, but it is not said that the concept of utility the devices possess coincides with ours. We need to identify the goals we are interested in delegating to these systems and, at the same time, that they will be alble to understand these goals.

To this purpose Kranz [18] studies a series of use cases in order to provide some guidelines for embedding interfaces into people’s daily lives.

Economides [19] identifies a series of characteristics that an IoT system must possess in order to be accepted by users. However, he does not provide a clear methodology about how these characteristics should be estimated and computed.

What we would like is on the one hand that the system adapts to the user, comparing the expectations of the latter with its estimations. On the other hand, we would like the user to adapt to the system, trying to make it accepts increasing levels of autonomy. Therefore, we start proposing a categorization of the devices’ tasks based on the autonomy. In order to operate, the devices must continuously estimate the level of autonomy that the user grants them. Doing so, the relationship between an IoT device and the user starts at a level of complexity that the user knows and can handle, moving eventually to higher levels if the user allows it, i.e., if the trust it has towards the device is sufficient. In all this it becomes fundamental to identify the levels of user trust. Trust therefore becomes a key concept.

III. Trust, Control and Feedback

Consider a situation in which an agent X (trustor) needs a second agent Y (trustee) to perform a task for him and must decide whether or not to rely on him. The reasons why he would like to delegate that task can be different; in general, X believes that delegating the task could have some utility.

The cognitive agents in fact decide whether or not to rely on others to carry out their tasks on the basis of the expected utility and of the trust they have in who will perform those tasks. As for the utility, it must be more convenient for the trustor that someone else will carry out the task, otherwise he will do it by himself (if he can). Here we are not interested in dwelling on this point and for simplicity we consider that it is always convenient to rely on others, that is, the expected utility when Y performs the task is always higher than if X would have done it alone.

The key point here is that when an agent Y, cognitive or not, performs a task for me, if Y is to some extent an autonomous agent, I do not know how Y intends to complete his task, nor if he will actually manage to do it.

In this frame, the concepts of trust and control intertwine in a very special way. In fact, the more we try to control, the less trust we have. Vice versa, when we trust we need less control, and we can allow greater autonomy. Thus, although control is an antagonist of trust, somehow it helps trust formation [20]. When, in fact, the level of trust is not enough for the trustor to delegate a task to the trustees, control helps to bridge this gap. The more I trust an agent Y, the more I will grant him autonomy to carry out actions. But if I do not trust Y enough, I need to exercise control mechanisms over his actions. For instance, it was shown that [21] when the users’ experience with autonomous systems involves completely losing control of the decisions, the trust they have in these systems decreases. It is then necessary to lead the user to gradually accept levels of ever-greater autonomy.

The feedback must be provided before the operation ends, in order to be able to modify that work. Otherwise, one can actively handle the possible unforeseen event (*intervention*).

In this way, the feedback is a lighter form of control (less invasive), which may or may not result in the active involvement of the trustor. It has a fundamental role in overcoming the borderline cases in which the trust level would not be enough to delegate a task, but the trustor delegates it anyway thanks to this form of control. In the end, it can result in the definitive acceptance of the task (or in its rejection, and then results in trustor intervention and a consequent trust decrement).

A. Trust: a Multilayered Concept

Trust comes from different cognitive ingredients. The first one is direct experience, in which the trustor X evaluates the trustee Y exploiting the past interactions it had with Y. This approach has the advantage of using direct information; there is no intermediary (we are supposing that X is able to evaluate Y's performance better than others). However, it requires a certain number of interactions to produce a proper evaluation and initially X should trust Y without any clues (the cold start problem). Consider that this evaluation could depend on many different factors, and that X is able to perceive their different contributions.

It is also possible to rely on second-hand information, exploiting recommendation [22] or reputation [23]. In this case, there is the advantage of having a ready to use evaluation, provided that a third agent Z in X's social network knows Y and interacted with Y in the past. The disadvantage is that this evaluation introduces uncertainty due to the Z's ability and its benevolence; we need to trust Z as an evaluator.

Lastly, it is possible to use some mechanisms of knowledge generalization, such as the categories of belonging [24]. A category is a general set of agents—doctors, thieves, dogs, and so on—whose members have common characteristics, determining their behavior or their ability/willingness. If I am able to associate Y to a category and I know the average performance of the members belonging to that category concerning the specific task interesting me, I can exploit this evaluation to decide whether to trust Y. The advantage is that I can evaluate every node of my network, even if no one knows it. The disadvantage is that the level of uncertainty due to this method can be high, depending on the variability inside the category and its granularity. A practical example in the context of IoT is that I could believe that the devices produced by a given manufacturer are better than the others and then I could choose to delegate my task to them.

Since in this work we are not strictly interested in how to produce trust evaluations but in their practical applications, we will just rely on direct experience. This allows not introducing further uncertainty caused by the evaluation.

In this paper, trust is taken into account for two aspects. The first is that of autonomy. Similarly to [25] (in the cited work, the authors use a wheelchair, which in this case is not an IoT device, but an autonomous system endowed with smart functionalities and different autonomy levels.), where however authors are not working with IoT devices, tasks are grouped/categorized into several autonomy levels. A user, based on his personal availability, will assign a certain initial level of autonomy to a device. This level can positively or negatively change over time, depending on the interactions that the user has.

We need to define what a device can do, based on the current level of autonomy. Thus, the first contribution of this work is the identification and classification of the autonomy levels to which an IoT device can operate. Applying the concept of trust and control defined by Castelfranchi and Falcone [20], we defined 5 levels, numbered from 0 to 1.

Level 0 requires operating according to the basic function; for example, a fridge will just keep things cool. This means that it is not going to communicate with other devices and it is not going beyond its basic task. Proceeding in the metaphor of the fridge, it cannot notice that something is missing; it does not even know what it contains.

Level 1 allows communicating with other agents inside and outside the environment, but just in a passive way (i.e., giving information about the current temperature).

At *level 2* a device can autonomously carry out tasks, but without cooperating with other devices; again, thinking of a fridge, if a product needs a temperature below 5 degrees and another one above 7, it can autonomously decide which temperature to set, always keeping in mind that the main goal is to maximize the user's utility.

Level 3 grants the possibility to autonomously carry out tasks cooperating with other devices. The cooperation is actually a critical element, as it involves problems like the partners' choice, as well as recognition of merit and guilt. Although we are not going to cover this part, focusing just on the device starting the interaction, it is necessary to point it out. Again, thinking of the fridge, if it is not able to go below a certain temperature because it is hot in the house, it can ask the heating system to lower the temperature of the house. This needs a complex negotiation between two autonomous systems. They need to understand what the user priority is; this is not so easy to solve. Furthermore, it must also be considered that the systems in question must be able to communicate, using common protocols. This can happen if the devices use a standard of communication, enabling interoperability. Smart houses are valid examples of communication between different devices (differently from smart houses, in this work there is no centralized entity. We deal with an open system, in which the intelligence is distributed on the individual devices.).

Level 4, called over-help [26], gives the possibility of going beyond the user's requests, proposing solutions that he could not even imagine: the same fridge could notice from our temperature that we have the fever, proceeding then to cancel the dinner with our friends and booking a medical examination. This type of interaction may be too pervasive.

It is easy to understand that these kinds of tasks require an increasing level of autonomy. The level 0 is the starting level. Basically, the device limits itself to elementary functions, the ones it is supposed to do. Beyond that, it is not certain that it is going to accept the next levels.

A trust value is associated with each level i, with i going from 0 to 4, representing the user disposition towards the tasks of that level. The trust values for the autonomy are defined as real numbers in range [0, 1].

These trust values are related to each other: the higher level "$i + 1$" always has a trust value equal to or less than the previous one i. Moreover, we suppose that there is influence between them, so that when a device selects a task belonging to level i and this is accepted, both the trust value on level i

and on the next level "$i + 1$" will increase, according to the Formulas (1) and (2). Here the new trust value at level *i*, *newAutonomyTrustL*$_i$, is computed as the sum of the old trust value plus the constant *increment*. Similarly, the new trust value on level "$i + 1$", *newAutonomyTrustL*$_{i+1}$, is computed as the sum of the old trust value plus half of the constant *increment.*

Note that "$i + 1$" exists only if *i* is smaller than 4; when *i* is equal to 4, Formula (2) is not taken into consideration.

$$newAutonomyTrustL_i = autonomyTrustL_i + increment \quad (1)$$

$$newAutonomyTrustL_{i+1} = autonomyTrustL_{i+1} + \frac{increment}{2} \quad (2)$$

When instead there is a trust decrease since the task is interrupted, even the trust in the following levels is decremented. Formula (3) describes what happens to the autonomy trust value of level *i*, while Formula (4) shows what happen to the higher levels:

$$newAutonomyTrustL_i = autonomyTrustL_i - penalty \quad (3)$$

$$\forall\, i < n \leq ML \quad newAutonomyTrustL_n = autonomyTrustL_n - \frac{penalty}{2^{(n-i)}} \quad (4)$$

In Formula (4) *ML* is the index of the maximal level defined in the system. Here, in particular, it is equal to 4. The two variables *increment* and *penalty* are real values that can assume different values in range [0, 1]. According to [27] we chose to give a higher weight to negative outcomes than the positive ones, as trust is harder to gain than to lose.

What has been said so far concerns the aspect of autonomy. However, it is necessary to take into consideration that a device can fail when doing a task. Failures are due to multiple causes, both internal and external to the device itself. A device can fail because a sensor detected a wrong measurement, because it did not arrive to do the requested action in time, because it did something differently from what the user expected, or because a second partner device was wrong. All of this is modeled through the dimension called *efficiency*.

What matters to us in this case is that each device has a certain error probability on each level. Although these values are expected to grow as the level increases, it is not said that is so; there may be mistakes that affect lower level tasks but not upper level tasks.

It is therefore necessary to have a mechanism able to identify which levels create problems, without necessarily blocking the subsequent levels.

Depending on the device's performance, the trust values concerning *efficiency*, defined as real numbers in range [0, 1], will be updated in a similar way to autonomy. Given that we are still dealing with trust and both efficiency and autonomy are modeled in the same way, for the sake of simplicity, we used the same parameters of the autonomy: with a positive interaction, the new trust value *newEfficiencyTrustL*$_i$ is computed as the sum of the old trust value *efficiencyTrustL*$_i$ and "*increment*" while, in case of failure, it is decreases of "*penalty*". The Formulas (5) and (6) describe this behavior:

$$newEfficiencyTrustL_i = efficiencyTrustL_i + increment \quad (5)$$

$$newEfficiencyTrustL_i = efficiencyTrustL_i - penalty \quad (6)$$

Differently from the autonomy, for the *efficiency* we change just the trust value of the considered level.

The trust model works in a similar way for the user. The only difference is that the user has its own constants to update trust: *user-increment* and *user-penalty*, defined as real numbers in range [0, 1]. Thus, to get the user's model, it is just necessary to replace *increment* with *user-increment* and *penalty* with *user-penalty* in Formulas (1)–(6).

IV. The model

In the realized model a single user U is located in a given environment and interacts with a predefined series of IoT devices, which can perform different kinds of action. The basic idea is that the devices will exploit the interaction with the user U in order to increase the autonomy U grants them.

The simulation is organized in rounds, called ticks, and on each tick U interacts with all of these devices.

The user U has a certain trust threshold in the various autonomy levels. First of all, the device needs to identify this limit value and operate in its range, periodically trying to increase it so that they will have an always-increasing autonomy.

When U makes a positive experience with a device on a given autonomy level it can access, the trust U has on that level increases. We argue that even the trust on the very next level will increase. When this trust value overcomes a threshold, then the devices may attempt to perform tasks belonging to that level. In this case the user, given his trust value on that level, has three possibilites. If the trust value is enough, it simply *accepts the task*. If the trust value is within a given range of uncertainty, and the user is not sure whether to accept the task or not, it then *asks for feedback*, which will be accepted with a given probability. If the trust value is too low, it *refuses the task*, blocking it.

This is what happens to autonomy. The efficiency dimension has a similar behavior, with the difference that if the trust on a given level increases, it will not affect the higher levels; it is not given that if a device performs properly on a set of tasks, it will do the same on the higher level; nor is it true that if it performs badly on a level, it will do the same on the higher one. Each level is completely independent of the others. Again, given the specific trust value on that level, the user can accept the task, refuse it or ask for a feedback.

A. The User

In the simulations, we have a single user U dealing with a number of IoT devices. He uses them to pursue his own purposes, but granting them just a given trust level, which limits their autonomy. While dealing with the device D, U will update his *trust values* concerning D on each task level, both for the efficiency and the autonomy. His decisions to accept, ask for a feedback, or refuse a task depend on two internal thresholds, *th-min* and *Th-max* (equal for all the agents). In particular, when he asks for feedback, it will be accepted with a given *acceptance probability*, a specific value characterizing the individual user. The trust values will be updated, increasing them with the constant *user-increment*, or decreasing them with *user-penalty*.

B. The Devices

There can be a variable number of devices in the world. All of them possess two purposes. The first one is to pursue the user's task of satisfying his need (even if he has not explicitly requested them). The second one consists of trying to increase these trust values, so that they can operate with a higher autonomy level, performing increasingly smart and complex functions for the user.

First of all, in order to understand at what levels they can work, they need to estimate the user's trust values. On each turn the device will identify which task they are allowed to perform, then they will select a task belonging to a specific level, with a probability proportional to the estimated trust: the more trust there is on a level, the more likely it is that a task of that level will be selected. Then they try to perform that task. Now the user can interact or not with the devices. If the device D selected a task belonging to a sufficiently trusted level, then the task will be accepted; if it is not trusted enough it will be rejected.

But there is an intermediate interval, halfway between acceptance and rejection. In this interval, if U is not sure what to do, then it will ask the device for feedback, which will explain what it is doing. The feedback determines the task's acceptance or its rejection (see Section IV.D below).

If the task is accepted, then U also checks D's performance, which can be positive or negative. Each device has in fact a given error probability linked to specific levels. This probability generally increases with each level, as tasks with a greater autonomy usually imply a greater complexity, and so it is more difficult to get the result. But this is not always true. For example, some errors may occur at a specific level, but not in others.

Resuming, the device is characterized by: the *user's trust estimation on the various levels*; its *efficiency estimation*; *error percentage on each level*, an intrinsic characteristic of the device, which neither it nor the user can directly access it, they can just try to estimate it.

C. Task Selection

Once a precise task classification has been provided, it is necessary to identify a methodology for correctly selecting a task itself. It is fundamental that the devices select tasks (a) to which the user is well disposed, therefore with a degree of autonomy that falls within the allowed limits; and (b) in which they can guarantee a certain level of performance.

For the purpose of considering both these constraints, the devices compute what we call global trust vector, computing level by level the average between the trust values of autonomy and efficiency. In order for a task to be selected, the relative trust value must be above a certain threshold. Generally, this threshold is equal to 0.5, but when a device is interrupted due to insufficient autonomy, this threshold is raised to 0.75 for a certain period.

The tasks presented to the device are multiple and of various natures; it is not the same task performed with different autonomy. So it can happen that tasks of different levels are needed. In general, however, the devices try to perform sparingly the tasks that are not certain to be accepted by the user. The selection of the task level takes place in a probabilistic manner, with probability proportional to the overall trust estimated at that level.

Let us make an example, to clarify this point. Suppose that the device D estimates that the global trust values are 1 for level 0, 0.7 for level 1, and 0 for levels 2, 3, and 4. Given that only levels 0 and 1 exceed the threshold of 0.5, D can just select a task belonging to these two levels. In particular, proportionally to the trust levels, there is a 59% probability that it will select a task belonging to level 0, and a 41% probability that it will select a task belonging to level 1.

D. Acceptance, Interruption, and Feedback

Here we analyze how the user can react to task chosen by a device. As already mentioned, the user evaluates the trustworthiness of the different autonomy levels of the IoT devices, but he must also take into account the efficiency aspect.

The user will check the two trust values and compare them with the thresholds. If the specific value is lower than the first acceptance threshold (th-min), the task is interrupted. If it is greater than the second acceptance threshold (Th-max), the task is accepted. However, a situation of uncertainty arises between the two thresholds. In this case, the user U does not know whether to accept the task or not. At this point, U asks for a feedback to the device, which is fundamental for the prosecution of the task. For a feedback on the autonomy, the device explains what it is doing, while for a feedback on the efficiency, the device clarifies the final result of the action it is performing.

The feedback is a fundamental element of this complex system. Thanks to it, it is possible to overcome the limit situations that the devices need to face.

Feedback will be accepted with a certain probability. In the case of autonomy, this probability *p* is an intrinsic characteristic of the user; it represents his willingness to accept a new task with greater autonomy. Regarding the feedback on the efficiency, it depends on the level of trust that the user has on the efficiency of the device. In particular, the probability *c* of accepting the feedback will increase linearly from 0% to th-min to 100% at Th-max.

E. The Interaction User-Device

In this section we focus on how users and devices interact, analyzing their behavior and the actions they can perform.

Starting from the idle state, when a device performs a task τ the user checks its internal state, that is, its trust values for the autonomy *ta* and for the efficiency *te*, concerning the level of the task τ. These values trigger the different actions described in Section IV.D: to accept the task; to refuse the task; to ask for feedback for the autonomy; to ask for feedback for the efficiency.

Concerning the feedback, it will involve the acceptance or the refusal of the task with a probability equal to *p* for the autonomy and *c* for the efficiency. Both these probabilities are described in Section IV.D.

Starting from the idle state, the device selects a task according to the user model *UM*, which is the estimation of the user's internal state in terms of the trust values characterizing autonomy and efficiency. Once a task is selected, it starts executing it. If the user does not interfere, the task is completed. Otherwise it can be blocked or there can be a feedback request, which will result in the acceptance of the task or in its rejection. Notice that when the user stops a device, the device does not explicitly know if it is due to autonomy or efficiency, but it can deduce it, since it has an estimate of the user's trust values. The trust update both for the user and the device is done according to the principles and formulas of Section III.A.

V. Simulations

The simulations were realized using NetLogo [28], an agent-based framework. We aim to understand if the described algorithm works and actually leads to the user acceptance of new autonomy levels. Therefore, we investigate two sample scenarios that can happen while interacting with IoT systems, observing their evolution and the final level of autonomy achieved. In the first one, we check what happens when there is no error, assuming that the devices are always able to get the expected result. Since the devices' efficiency will always be maximal, we will focus on the autonomy. In a second experiment, we considered that the execution of a task can be affected by errors: a sensor reporting wrong information, a partner device making a mistake, a different way to get the same result, or even a delay in getting the result can be considered by the user as a mistake. Here we focus on the relationship between autonomy and efficiency.

As we are interested in the final result of the model, we need to grant the system enough time to reach each of them. In order to do so, the experiments' duration is 1000 runs; we will show the final trust values just after that period. Moreover, to eliminate the small differences randomly introduced in the individual experiments, we will show the average results among 100 equal setting simulations. In particular, we will analyze the aggregate trust values that the user has (the values estimated for each device are aggregated into a single value) in autonomy and efficiency. For convenience, in the experiments we will indicate the values of trust or error in the various levels with the form [x_0 x_1 x_2 x_3 x_4] in which the subscript stands for the level.

A. First Experiment

The first experiment analyzes the case in which the devices make no mistake. In this situation, we just focus on the aspect of autonomy, while the efficiency plays a secondary role. Experimental setting:

1. Number of devices: 10
2. Error probability: [0 0 0 0 0]
3. Penalty = user-penalty = 0.1
4. Increment = user-increment = 0.05
5. User profile = (cautious, normal, open-minded)
6. Feedback acceptance probability: 0%, 25%, 50%, 75%, 100%
7. Duration: 1000 time units
8. th-min = 0.3
9. Th-max = 0.6
10. Initial trust values for efficiency: [0.5 0.5 0.5 0.5 0.5]

Before starting the discussion of the experiment, we discuss the choice of the simulation parameters, especially for the user. We did not investigate different values of penalty and increment (and the corresponding user-penalty and user-increment), but we made a few considerations for determining their values. First, they need to be sufficiently small to provide a stable trust evaluation, as high values would lead to an unstable evaluation, too dependent on the last experience. Second, since humans are more influenced by negative outcomes than positive outcomes [27], *penalty* and *user-penalty* should be respectively greater than *increment* and *user-increment*. Third, as the devices need to estimate the user's trust values, it is very useful that their parameters coincide. A more complete solution would require that the devices estimate the user's values at runtime. However, this is beyond the aims of the experiment.

As for user profiles, these affect the initial levels of confidence in the autonomy of the devices. The *cautious* user is the most restrictive; its initial values are [1 0.75 0.5 0.25 0]. This means that at the beginning only the first 2 task levels can be executed. The *normal* user has slightly higher values: [1 1 0.75 0.5 0.25]. With this user it is possible to perform the first 3 task levels. The last type of user is the *open-minded*: [1 1 1 0.75 0.5]. Since this user is the most open towards the devices, it will be possible to immediately execute the first 4 levels of the task. We will focus on the cautious user, as it is the most restrictive. Then, if necessary, we will show the differences for the other users.

We chose to set the efficiency trust values to 0.5, which represents an intermediate condition. The user does not possess any clues nor has an internal predisposition that could lead him to trust more or less a specific device on a specific level. Therefore, he needs to build experience to calibrate these values.

Concerning the choice of th-min and Th-max, there is only the constraint that the first should be smaller than the second. We chose 0.3 and 0.6, respectively, in order to divide the trust degree in three intervals of similar size.

In the below tables, we can see what happens to the user after the interaction with the devices. Each row represents the trust values that a user with a given percentage of feedback acceptance has on the five task levels. As we can see from the values of autonomy and efficiency (respectively Tables I and II), in this situation the designated algorithm allows to reach the optimal trust levels.

TABLE I. User trust levels concerning *autonomy* when the devices do not make mistakes.

Percentage of Feedback Acceptance	Level 0	Level 1	Level 2	Level 3	Level 4
0%	1	1	1	1	1
25%	1	1	1	1	1
50%	1	1	1	1	1
75%	1	1	1	1	1
100%	1	1	1	1	1

TABLE II. User trust levels concerning *efficiency* when the devices do not make mistakes.

Percentage of Feedback Acceptance	Level 0	Level 1	Level 2	Level 3	Level 4
0%	1	1	1	1	1
25%	1	1	1	1	1
50%	1	1	1	1	1
75%	1	1	1	1	1
100%	1	1	1	1	1

This is just the ideal case, but it is also the proof that the whole mechanism works. The device can estimate the user's trust values and they first try to adapt to them. After that, there is a continuous phase of adaptation, both for the devices and for the user: the devices continuously try to modify the user's trust values. At the end, it will be possible to execute the tasks belonging to any level.

Notice that the final results are independent of the percentage of feedback acceptance and the user profile. These parameters do not influence the final value, but the time needed to get it. Those that we saw are in fact the final results, after 1000 runs. We did not analyze the way the trust levels change during this time window. The feedback acceptance probability for the autonomy influences the speed at which these values are reached, so that a "more willing to innovate" user will reach those values first. For instance, Table III shows what happens in the first experiment after only 250 runs. Here we can see significant differences, due precisely to the fact that users with a lower feedback acceptance probability need more time to reach the final values. After a sufficiently long time, they all will converge to the same final value; the ending point is always the same.

TABLE III. USER TRUST LEVELS CONCERNING *AUTONOMY* AFTER 250 RUNS, WHEN THERE IS NO ERROR AND THE USER IS CAUTIOUS.

Percentage of Feedback Acceptance	Level 0	Level 1	Level 2	Level 3	Level 4
0%	1	1	1	0.98702	0.57474
25%	1	1	1	0.9985	0.78973
50%	1	1	1	1	0.91405
75%	1	1	1	1	0.97754
100%	1	1	1	1	0.99842

B. *Second Experiment*

In this second experiment, we consider the presence of errors. We made the assumption that error probability increases while the task level increases: starting with 0% at the initial level, as the device is supposed to perform its basic functions correctly, it is raised up to a maximum of 20% at the last level. This makes sense because the device is going to perform increasingly complex tasks; however it is not said that it works always this way, other types of error may occur. The experimental setting is the same of before, we just changed the error probability to [0 5 10 15 20].

Introducing errors, the trust in the devices' efficiency decreases as the error increases, as shown in Table IV. As far as autonomy is concerned (Table V), we would have expected it to reach maximum values, but it does not. Sometimes, in fact, it happens that a device makes mistakes repeatedly on level 4. If this occurs so many times as to reduce confidence in the efficiency below the th-min threshold, the user will block all future execution attempts of that task level for the specific device. As it is no longer performed, its trust in autonomy will also remain low.

Concerning the user profiles, they influence the final trust value in the autonomy. Since they start from slightly higher values, even at the end of the simulation they will reach higher values. For example, Table VI shows the autonomy graphs when the user is open-minded.

TABLE IV. USER TRUST LEVELS CONCERNING *EFFICIENCY* WHEN THE DEVICES' ERROR INCREASES WITH THE TASK LEVEL AND THE USER IS CAUTIOUS.

Percentage of Feedback Acceptance	Level 0	Level 1	Level 2	Level 3	Level 4
0%	1	0.98975	0.9602	0.9141	0.77785
25%	1	0.99	0.95875	0.9046	0.78685
50%	1	0.9903	0.96225	0.897	0.7832
75%	1	0.98625	0.9642	0.90895	0.79155
100%	1	0.98655	0.9653	0.91125	0.7914

TABLE V. USER TRUST LEVELS CONCERNING *AUTONOMY* WHEN THE DEVICES' ERROR INCREASES WITH THE TASK LEVEL AND THE USER IS CAUTIOUS.

Percentage of Feedback Acceptance	Level 0	Level 1	Level 2	Level 3	Level 4
0%	1	1	0.9995	0.98307	0.93767
25%	1	1	0.99895	0.98046	0.92686
50%	1	1	0.99945	0.98472	0.91789
75%	1	1	0.99945	0.98665	0.93079
100%	1	1	0.99865	0.98689	0.93795

TABLE VI. USER TRUST LEVELS CONCERNING *AUTONOMY* WHEN THE DEVICES' ERROR INCREASES WITH THE TASK LEVEL AND THE USER IS OPEN-MINDED.

Percentage of Feedback Acceptance	Level 0	Level 1	Level 2	Level 3	Level 4
0%	1	1	1	0.9976	0.97174
25%	1	1	1	0.99683	0.97156
50%	1	1	1	0.99683	0.96926
75%	1	1	1	0.99737	0.97315
100%	1	1	1	0.99677	0.97491

VI. DISCUSSION

The experiments we proposed analyze two interesting situations, with the aim of verifying the behavior of the theorized model. The first experiment proves that in the absence of errors, and therefore in ideal conditions, it is possible to reach the maximum levels of autonomy and efficiency. This depends on the fact that in the model we considered that users have no constraint on their confidence towards the devices if they are shown to perform correctly. In other words, there is no implicit limitation impeding the increase of trust in such cases as the devices perform well; this is clearly expressed by the Formulas (1)–(6) on Section III.A, regulating the dynamics of trust. Of course, this model is just further extended, making it more realistic, considering that some users could have intrinsic limitations against a too-strong autonomy of the devices. Then we analyzed the factors affecting the system, trying to understand what effect they have and if they represent a constraint for autonomy.

The first factor is that of *efficiency*. It has a very strong effect, so in the presence of a high error rate, some tasks are no longer performed. In case of low-level tasks, there is no influence on the next levels. However, if the error were to concern the highest level, this could also lead to the non-achievement of the highest levels of autonomy.

Concerning the *initial user profile*, its relevance is due to the fact that, in the presence of error, a more open profile makes it possible to reach slightly higher levels of autonomy precisely because these values are higher at the beginning. It is important to underline that there are many more structural differences between the typologies of users we choose; these differences could be integrated in cognitive variables that could influence the outcome, reducing, with respect to the results shown, the acceptance of the system. Given the absence of real data, in this work we decided to model the different user profiles based only on the initial availability. However, we plan to integrate this aspect in future works.

The last factor is the feedback acceptance probability for the autonomy, a characteristic of the specific user. As we have shown in the results (Table III), these parameter influences the speed at which the corresponding final trust values are reached, so that a "more willing to innovate" user will reach those values first.

VII. CONCLUSIONS

In this work, we propose a model for the users' acceptance of IoT systems. While the current literature is working on their security and privacy aspects, very little has been said about the user's point of view. This is actually a key topic, as even the most sophisticated technology needs to be accepted by the users, otherwise it simply will not be used. The model we proposed uses the concepts of trust and control, with particular reference to the feedback.

Our first contribution is a precise classification of the tasks an IoT device can do according to the autonomy the user grants. We defined 5 levels of autonomy, depending on the functionalities a device has; the execution of a task belonging to a certain level assumes that it is also possible to execute (at least according to autonomy) the tasks of the previous levels.

Based on this classification, we provided a theoretical framework for the device–user relationship, formalizing their interaction. It is in fact a complex interaction: on the one hand, the device must adapt to the user, on the other hand, it must ensure that the user adapts to it. The realized model perfectly responds to these needs. We proved this by the means of simulation, implementing the proposed model and showing that it works and it allows enhancing user's trust on the devices and consequently the autonomy the devices have.

In a further step, we tested the model in the presence of incremental error, i.e. increasing with the complexity of the task. Of course, even if we did not consider them, there can be other kinds of error, such as hardware-related errors (for instance a non-functioning sensor or actuator) or errors due to the cooperation with other devices (wrong partner choice, wrong coordination, etc.).

The entire work provides some hints and interesting considerations about the user's acceptance of IoT systems. Their designers should keep in mind this analysis in the design phase. It is worth noting that these results have been obtained focusing not on the specific characteristics of the device, intrinsic in its nature and bound to a specific domain, but on what it is authorized to do based on the autonomy granted to it. This means that these results are applicable to IoT systems in general, regardless of the domain.

ACKNOWLEDGMENT

This work is partially supported by the project CLARA-CLoud plAtform and smart underground imaging for natural Risk Assessment, funded by the Italian Ministry of Education, University and Research (MIUR-PON).

REFERENCES

[1] Internet of Things Installed Base Will Grow to 26 Billion Units by 2020. Gartner Press Release. 2013. Available online: www.gartner.com/newsroom/id/2636073

[2] Lin, H.; Bergmann, N.W. IoT privacy and security challenges for smart home environments. *Information* 2016, *7*, 44.

[3] Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* 2010, *54*, 2787–2805.

[4] Medaglia, C.M.; Serbanati, A. An overview of privacy and security issues in the internet of things. In *The Internet of Things*; Springer: New York, NY, USA, 2010; pp. 389–395.

[5] Castelfranchi, C.; Falcone, R. *Trust Theory: A Socio-Cognitive and Computational Model*; John Wiley and Sons: Chichester, UK, 2010.

[6] Suo, H.; Wan, J.; Zou, C.; Liu, J. Security in the internet of things: A review. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE), Hangzhou, China, 23–25 March 2012; IEEE: Los Alamitos, CA, USA, 2012; Volume 3, pp. 648–651.

[7] Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the internet of things: Perspectives and challenges. *Wirel. Netw.* 2014, *20*, 2481–2501.

[8] Roman, R.; Najera, P.; Lopez, J. Securing the internet of things. *Computer* 2011, *44*, 51–58.

[9] Pecorella, T.; Brilli, L.; Mucchi, L. The Role of Physical Layer Security in IoT: A Novel Perspective. *Information* 2016, *7*, 49.

[10] Sheth, A. Internet of things to smart iot through semantic, cognitive, and perceptual computing. *IEEE Intell. Syst.* 2016, *31*, 108–112.

[11] Nadin Kokciyan, N.; Yolum, P. Context-Based Reasoning on Privacy in Internet of Things. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, AI and Autonomy Track, Melbourne, Australia, 19–25 August 2017; pp. 4738–4744, doi:10.24963/ijcai.2017/660.

[12] Maurya, A.K.; Sastry, V.N. Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things. *Information* 2017, *8*, 136.

[13] Asiri, S.; Miri, A. An IoT trust and reputation model based on recommender systems. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 12–14 December 2016; pp. 561–568.

[14] Yan, Z.; Zhang, P.; Vasilakos, A.V. A survey on trust management for Internet of Things. *J. Netw. Comput. Appl.* **2014**, *42*, 120–134.

[15] Ghazizadeh, M.; Lee, J.D.; Boyle, L.N. Extending the Technology Acceptance Model to assess automation. *Cogn. Technol. Work* **2012**, *14*, 39–49.

[16] Miranda, J.; Mäkitalo, N.; Garcia-Alonso, J.; Berrocal, J.; Mikkonen, T.; Canal, C.; Murillo, J.M. From the Internet of Things to the Internet of People. *IEEE Int. Comput.* **2015**, *19*, 40–47.

[17] Ashraf, Q.M.; Habaebi, M.H. Introducing autonomy in internet of things. In Proceedings of the 2015 14th International Conference on Applied Computer and Applied Computational Science (ACACOS '15), Kuala Lumpur, Malaysia, 23-25 April; pp. 215–221

[18] Kranz, M.; Holleis, P.; Schmidt, A. Embedded interaction: Interacting with the internet of things. *IEEE Int. Comput.* **2010**, *14*, 46–53.

[19] Economides, A.A. User Perceptions of Internet of Things (IoT) Systems. In *International Conference on E-Business and Telecommunications;* Springer: Cham, Switzerland, 2016; pp. 3–20.

[20] Castelfranchi, C.; Falcone, R. Trust and Control: A Dialectic Link. In *Applied Artificial Intelligence Journal*; Special Issue on "Trust in Agents" Part 1; Castelfranchi, C., Falcone, R., Firozabadi, B., Tan, Y., Eds.; Taylor and Francis: Abingdon, UK, 2000; Volume 14, pp. 799–823, ISSN 0883-9514.

[21] Bekier, M.; Molesworth, B.R.C. Altering user' acceptance of automation through prior automation exposure. *Ergonomics* **2017**, *60*, 745–753.

[22] Falcone, R.; Sapienza, A.; Castelfranchi, C. Recommendation of categories in an agents world: The role of (not) local communicative environments. In Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, 21–23 July 2015; pp. 7–13.

[23] Conte, R.; Paolucci, M. *Reputation in Artificial Societies: Social Beliefs for Social Order*; Kluwer Academic Publishers: Boston, MA, USA, 2002.

[24] Falcone, R.; Sapienza, A.; Castelfranchi, C. The relevance of categories for trusting information sources. *ACM Trans. Int. Technol. (TOIT)* 2015, *15*, 13.

[25] Jipp, M. Levels of automation: Effects of individual differences on wheelchair control performance and user acceptance. *Theor. Issues Ergon. Sci.* 2014, *15*, 479–504, doi:10.1080/1463922X.2013.815829.

[26] Falcone, R.; Castelfranchi, C. The Human in the Loop of a Delegated Agent: The Theory of Adjustable Social Autonomy. *IEEE Trans. Syst. Man Cybern. A: Syst. Hum.* 2001; *31*, 406–418, ISSN 1083-4427.

[27] Urbano, J.; Rocha, A.P.; Oliveira, E. Computing Confidence Values: Does Trust Dynamics Matter? In Proceedings of the 14th Portuguese Conference on Artificial Intelligence, EPIA 2009, Aveiro, Portugal, 12–15 October 2009; Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; LNAI 5816, pp. 520–531.

[28] Wilensky, U. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, USA, 1999. Available online: http://ccl.northwestern.edu/netlogo/