Questions about the Bidirectional Influence between Smart Home Companions and Users - Projekt 2

Nora Berg

October 4, 2017

1 Introduction

The term Companion Technology describes devices that assist users in handling mundane tasks for them and increasing the users overall comfort. While there exist a range of companions in health care, cars, or other areas of life, this work is mainly concerned with companions in smart homes. i.e. households with one resident. As Weiser [16] had predicted at the beginning of the 90s, companions become more and more included into everyday devices and computer become more and more invisible. It's the object itself that matters, not the computer in it. Companions are aimed to adapt to the users current lifestyle as much as possible, but the user is also influenced by the companion in two ways. The first way is that users learn how to *use* the object they included into their life to achieve the offered benefits. Additionally, companions have limitations how much information is retrieved and how it can be interpreted, and therefore the user also needs to learn to *administrate* the companion to provide additional information how the companion should act in borderline situations. The implementation approach to automate more and more of those meta-decisions can lead to "too independent" companions, which can also lead to decreased acceptance by users because of the feeling of lost control [13].

Companions impact the world of the user. The relationship between technical devices and human users have been subject to several studies in computer science, philosophy and sociology. It was shown that computers became part of our social fabric with influence on peoples lifes, capable of unpredictable emerging behavior [10, 14] and that trust relationships between human and computers are comparable to those between two humans [6].

In this work we deduce the questions from combining the bidirectional influence between technical companions and humans, and the technical implementation of companions. Our questions regard the impact of the companions, visible not only in the advantages but also in case companions fail in their functionality. Who feels responsible if companions fail or betray the trust of users? How could a good and emotionally healthy lifestyle for users of companions be ensured? How can be ensured that companions not only act well for a majority of people, but also for people with "other preferences"? How can users be safe from manipulation of manufacturers, especially since those devices are included in most private areas and equipped with a lot of sensors to pick up as much as information as possible?

This work is structured as follows. In section 2 an overview over the related papers is given, followed by an overview over companion technology and their technical implementation in section 3. Thereafter different approaches to analyze the relationship between humans and technology as well as their bidirectional influence is given in section 4. We conclude with several questions concerning the relationship between users and companions in smart homes in section 5, followed by an outlook for future work in section 6.

2 Related Work

The related work of companion systems can be categorized into two areas. The first area concerns companions system from a *technical* point of view. This includes specific ways to implement those systems. The *environmental* view investigates how companion systems influence environment and users. Between both views exists a bidirectional influence therein the environment defines the goals of the technical system and experiences its effects.

A technical overview gives [3]. It defines an companion system as a technical system with cognitive abilities and the main goal of increasing the comfort of the user. To create appropriate technical abilities research distinguishes between the fields of advanced human computer interaction (e.g. environment perception, emotion recognition), planning, and learning. To enhance the interaction between users and technical systems, life sciences as psychology and neurology play an important role. Furthermore, the paper gives an overview on current application areas as Robotics, Smart Homes, Health and Elderly Care, as well as Driving Assistance.

An Overview over the technical requirements and implementation is given in [4]. Analogous to [3], the goal of companion technology is to increase the users comfort. Increasing comfort is supposed to be achieved by supporting the user in its decisions and actions. To achieve this, companion systems should find solutions and plan actions adapting user specific requirements and needs. The paper emphasizes *availability*, *cooperativeness*, and *trustworthiness*, as well as *transparency* towards the user as requirement to increase comfort. The main tasks of companions include planning, decision making, interaction with the user, and integration of situational contexts as e.g. sensor data or emotional states of the user. The implementations of companion technology is described with models of the *system view* and the *user view* and their reciprocal requirements and expectations. Therein the *system view*, creates a model of the world with help of the previous tasks, sensory inputs and analysis of speech and psychological features. The *user view* describes perception and impact on the user. The dialog of the system has the task to mediate between those views to adapt the system view of the situation.

Recommender Systems are used in companion technology to recommend *things*, personalized to attributes of users. In traditional applications recommendations are made directly made for objects, e.g. as product advertisement in web shops or movie recommendations in streaming services. In companion technology, recommendation systems can be used to plan future actions and decide which action is the most appealing. The implementation of recommender systems in general depends on two parts, the attributes and the decision algorithms [1].

In [1] basics of context aware recommender systems are summarized. While traditional recommender systems use static user information to calculate a rating whether a specific item fits to a user, context aware recommender systems include time-dependent attributes as, e.g. location, weather, or companionship of users. Context-aware recommender systems can increase the precision of recommendations compared to standard recommender systems. But they also need to elicit contextual data on a regular basis, e.g. through UI or sensors.

The environmental view on companion raises the question in which ways companions impact peoples lives. The range of this question spreads from dialog designing in the field of human computer interaction to philosophical questions how the world changes with all its complex coherences due to the additional *things* introduced by technology. In its full extend the field includes aspects from computer science, psychology, sociology, neurology, art and philosophy.

The master thesis [13] examines companion systems with focus on how acceptance of technical systems can be increased. Thereby, perceived control of the user and reliability of companions is named as major influence on acceptance. In contrast, the thesis states that limitations of companion systems, i.e. in collecting and interpreting environmental data, as well as unpredictability of the user, lead to always imperfect companions. To increase acceptance it is examined why, when and how control can be implemented into companions. It emphasizes that control mechanisms via system feedback (information towards the user), user feedback (information towards the companion), user fallback (explicit preferences set by the user that overwrite all learned or "smart behavior"), and system fallback (recommendation of actions by the system in unexpected situations) oppose against complete automation which would take away the control from the user and would therefore reduce acceptance.

Another interesting aspect for the increasing pervasiveness of smart home companions concerns the *trust* users have in their systems. The reason is that companions are implemented in the users most private areas. Buecher, Simon and Tavani [6] investigate the trust relationship between humans and artificial agents from a philosophical standpoint. A trust relationship has two main aspects. The competence level for a given task, as well as moral considerations of both trustor and trustee. The main difference between a trust relationship among humans in comparison to trust relationships between humans and artificial agents is that artificial agents lack the control over the moral decisions a human can consider. Nevertheless, the competence level can be fulfilled by artificial agents.
For this work it is interesting which kind of moral standpoints are affected in smart home environments and how they affect the trust and acceptance of companions.

A more wholesome approach to understand the impact of increasingly pervasive computerized environment is taken in [10]. The main statement here is that the old understandings of relationships between persons and artifacts is heavily changed by the increasing intelligence, autonomy and pervasiveness of technical systems. Where in older times machines were used *by* people to accomplish tasks, machines nowadays make independent decisions. This turns the relationship between people and machines into bidirectional relationships. The new influence from technological artifacts on people emerges because machines and the network of machines are not understood in their completeness and operate on levels invisible to users. Technology in its whole becomes an ecosystems of machines in which humans do not control everything. Words like *emergence* and *control* emerge in discussions about technology and we see that the current technological environment is not a simple tool anymore. The book emphasizes the necessity to analyze the relations between technological artifacts themselves, between technological artifacts and humans, and the chains on influence between all those elements. This becomes important since the technological environment is a core part of current reality.

A similar approach is done in Actor Network Theory (ANT) [14]. The goal of this approach is to understand bidirectional influences between the technological world and the social world. For this the ANT gives guidelines how to survey relationships between objects and humans and the networks the create. In ANT everything is perceived as an actor, an object which can take any form, may it be a human, animal or technical devices or software. To achieve new insights about the reciprocal influence between actors, they should be abstracted from (nearly) all presumptions that might come from the learned understanding of what the actor actually is in real life. Instead actors are described by their set of tasks and the relationships to other actors. Characteristics of actors, network and relationships are said to give information about the overall state of the network. Interesting aspects include stability of the network, redistribution of tasks between the actors or how real life actors can be convinced to execute their ascribed tasks.

The influence from technical innovations is explored in the past on several occasions and depicted future living concepts pretty accurate. An predictive paper was from Mark Weiser at beginning of the 90s [16]. Therein he predicts the future of computers to be part of the every day environment. Computers disappear into the background and become part of everyday objects to enable the users to achieve the work without explicitly having to *use a computer*. He includes the overall monitoring of the environment, may it be people (i.e. coworkers, neighbors, family members), or things (i.e. coffee makers, written notes). This is necessary to help the user getting tasks done. He also mentions privacy and security concerns, as well as abuse of power of information by institutions, i.e. states or companies.

Domestication is the process how people learn how to use items, i.e. companion technologies, and change their lives while including those. An early well researched example of this process is given in the development of the *Frankfurter Küche* [11], a newly developed kitchen environment in Germany of the 1930s. The domestication of software- and hardware and a more abstract description of domestication is given in [2, 15]. They name five steps from *appropriation* of an item to *conversion* of the usage of the item, which consumers go through while implementing objects into their lives.

3 Companion Technology

Companion Technology is broadly defined as technology that supports humans in their daily lives by increasing the comfort for the user [3, 4]. The increase of comfort can take many forms due its application in various fields. The deployment of companion technology ranges from medical support in health and elderly care, over specific task that need to be accomplished, e.g. driving assistance, to everyday tasks as living at home. Beside the common goal, similarities between all those fields can be found in the implementation of companions in the environment. In this section an overview over the implementation of companions is given.

3.1 Parts of Companions

The setup on which companions work differ due to the area of application, Nevertheless, on an abstract level the same basic tasks have to be fulfilled by all companions. Those tasks entail abstract *parts* of companions, each solving one (or a set of) task.

Tasks and parts of companions can be observed by the looking at the interplay between the companion and environment. As in Fig. 1 companions perceive *environmental information* to *plan actions* and then *interact* with the environment or the user to achieve a *goal*. The companions themselves differ due to their task in their implemented sensors, intelligence and the way of interaction between user and system.

The User is the center of the companion technology development process. The reason is that a system that isn't accepted by the user, for whatever reason, is useless. The *acceptance* of companion technology is influenced not only by pure informational facts that are exchanged with the user and the goal that can be achieved, but also by more psychological questions, as how the system is perceived, can be controlled and whether

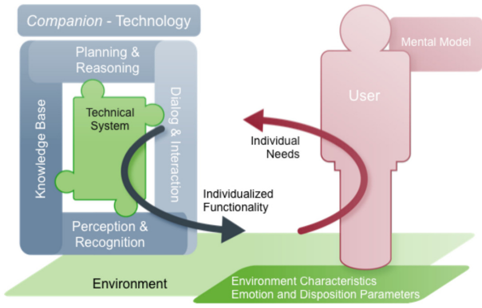

Figure 1: The interaction between user and system is influenced by the knowledge of the system over the situation and the assumptions an user has over the system. (Fig. from [5])

the system is predictable [13], or more abstractly, how it fits to the mental model the user has in mind [5].

Environmental Information or *context* is a data set to describe the environment to the system. It can be retrieved in several ways like sensor measurements, direct user input, and indirect conclusions. Also data about users are part of the environmental information, since users are part of the systems environment.

All informations are stored in a *knowledge base* and are the input for decision making processes while planning actions. A set of environmental information is defined as *states*. They need to be put in an abstract form to allow computational processing [9]. The information in the knowledge base can differ in several aspects, e.g. in temporal validity, abstraction level, update behavior, or in the relevance to the decision making algorithms. For example, a value for temperature measurement has limits in its temporal validity. The need to update such data is due to the context. For a heating control system there is a need to update regularly, while a temperature measurement of a medical thermometer is mostly needed on demand.

Dialogs are responsible to communicate between user and system. Within dialogs preferences can be either set directly and states can be determined in a direct or indirect way. For example, a user can set a time in which she will not be disturbed or the dialog can deduce that from the fact that she is at work. The dialog setup is important for the acceptance of companions. Confusing, overly interrupting, incomprehensible or just unattractive dialog designs can lead to rejection of companions no matter how good the remaining parts of the companion are implemented. On the other hand, a good design and usability can overcome (some) technical deficiencies. How much dialogs influences the usage of a companion also depends on the needs a user has to have a task fulfilled. A user might be willing to click through long dialogs and reveal a large amount of personal information to get achieved something important rather than to work through the same dialog for something unimportant.

Planning The planning component is the core of a companion device and requires a lot of artificial intelligence. It allows to switch from the current state to another more desired ($\approx$ more comfortable) state [3]. These two states together describe the *planning problem* [9]. The knowledge base of an intelligent device includes *actions*. An action consist of a set of contextual variables that must be fulfilled and a description which

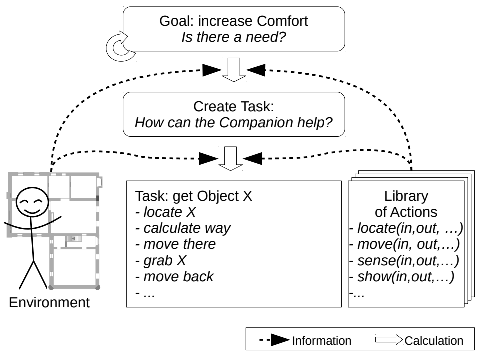

Figure 2: Information flow of the planning component

state would be the outcome of the action. Actions can either be executed automatically by the device or require user input or intervention. A *task* is defined as a sequence of those actions. Tasks can be used to overcome the gap between the current state and the desired state and therefore solve the planning problem. The calculation of tasks is the job of the planning component. Besides an initial computation of tasks, the companion is required to repeatedly check the current state of user and environment, and include emerging changes into their calculations.

If changes happen during execution of actions, an alternative goal or plan is calculated according to the new situation. In the end the solution to the planning problem is a task that fits best to the user preferences. Each action in this task has to have fulfilled prerequisites, otherwise it is not executable. Each task can be either a set of instructions to the user, executed automatically by the device or consist of both options. The stages of task creation is shown in Fig. 2. If a need of the user is detected, e.g. user needs object X, then the companion determines whether it should do something. Therefore it includes all available information about the environment, the user, and its own possibilities in form of actions. If the presumed outcome of the calculated task is good enough, the execution of the task is triggered. In this case it would include getting object X to the user. Besides the execution of tasks, the companion needs to be able to alter or stop its own tasks if the situation requires it. For this, the first stage is always running parallel to the other stages and checks the current state of environment. To change its own behavior, the companion needs see itself and the current running tasks as part of the environment, so that another task could also be `stop_other_task(Y)`.

There are plenty algorithms and optimization possibilities in the area of planning algorithms. Which one is useful depends on the device and the kind of planning problem.

For example, the breadth-first search [8] can be used to find a solution for a creating a task by combining actions. Starting at the goal state, we build a search tree by repeatedly choosing all actions that fulfill a non empty subset of prerequisites of the goal state or the preceding intermediate actions. These actions are added as leaves to the search tree until a way is found that fits to the users preferences. A way is found when a state is added as leaf, which has all prerequisites fulfilled in the current state.

An example for the search process is shown in Fig. 3. In this scenario a companion recognizes the need to have an object X at a specific location Y. The outcome of an action for moving objects (*move*) would fulfill this action, so it is chosen as possible action and added as the first leave. For a breadth-first-search, all possible actions that fulfill the new prerequisite would be added as leaves to the respective state. All those actions which do not fulfill the prerequisite of the examined step would be ignored. In the next step, all new leaves are are looked at (here this is stage with action *move*),

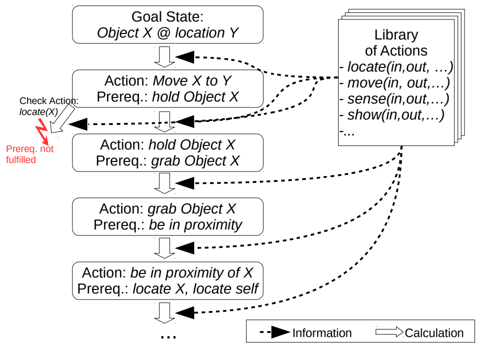

Figure 3: the planning components tries to find a sequence of actions to fulfill a goal

and their prerequisites become goal of the next step. So all actions are chosen as new leave which would fulfill the required prerequisite of *holding an object*. All other actions, e.g. the *show* action which do not fulfill the prerequisite are therefore discarded for this step. The choosing of new actions goes on until one or more actions are found, those prerequisites are all fulfilled at the current state of the companion. The path from the last leaf, i.e. a current possible action, to the root of the search tree, i.e. the goal state, describes a task in form of a sequence of actions and solves the planning problem. It depends on the application, how many tasks are searched for, how many action they include, when the search considered finished or aborted, the reasons for either, and which task is finally executed.

The choosing of goal states and best fitting tasks is another function of the planning component. Here, context aware recommender systems [1] come to good use. Context aware recommender systems have the ability to rate specific actions to in their fitting the current situation. In literature these functions most of the time map abstract *items* to a user, but note that in case of the planning component context aware recommender systems would rate fully generated plans, actions, and assumed outcomes to a user and a situation. Nevertheless, in the following paragraphs the name *item* is adopted.

Traditional recommender systems are represented by the function $R : U \times I \rightarrow Rating$ that maps attributes of *User* (U) and *Item* (I) to a rating $r_{u,i} \in Rating$. U and I are sets of attributes that describe the specific user u or item i. The rating $r_{u,i}$ is element of the totally ordered set *Rating* which describes how good an item fits to the user, e.g. on a scale from one to ten as in $R_{MusicEquipment}(John, guitar\ strings) = 9$.

Context aware recommender systems additionally include contextual information of typically limited temporal and/or spatial validity. The recommendation function is therefore described by $R : U \times I \times C \rightarrow Rating$. Set C consist of contextual attributes, as e.g. temporal (weekend, 12-24-2023), spatial (London, home) or otherwise user related (in company, ill, happy). If the planning component chooses a goal state and a task, context aware recommender system can rate goals and tasks to find the task that has the best outcome for the user. In a simplified example, a smart heating system would reduce room temperature at night if the user is healthy. Then the goal would be *increase sleep comfort by cold and fresh air*, and the action would be *reduce temperature*. Although, if the context variables of the user would say *high body temperature* or *shivering*, the intelligent heating system would consider changing the plan, since the user is ill.

In general, contextual variables can be added to the traditional recommender calculations in three different ways. *Pre-filtering* filters possible goals or tasks according to the context before actual rating them. *Post-filtering* first calculates all goals or tasks, and thereafter eliminates the ones that do not fit to the context. *Contextual model-*

ing includes contextual information directly into the rating function. There are several approaches for it, e.g. using distance measures in multidimensional spaces which are spanned by the context variables for a nearest neighbor approach, or diverse machine learning algorithms (for more overview see [1]).

On a further note, we see that with a high number of different information the quality of goals and tasks can be improved. But with the increasing amount of contextual information the interpretation of environment and user becomes a increasing complex and complicated endeavor. Wrong interpretations of situations cause effects on environment and user. Further, privacy and trust come into play if a system collect a lot of personal data. From this point we enter the more ethical discussion about trust and risks of intelligent systems.

4 Resciprocal Influence between Companion Technology and People

To establish increased comfort, companions need to be accepted by the user. Acceptance is a state of mind of the user to willingly use a technical system. Thereby, benefits of the system exceed possible drawbacks and the users overall comfort is increased. In this section two influences on acceptance are described. On the one hand *control* (section 4.1) over a technical system contributes to acceptance by the user. Since not everything can be controlled by the user we also look at *trust* (section 4.2) and how it relates to acceptance and comfort. Further, trust and control interact with each other (section 4.3), and we see how new items are included into users daily life and which kind of influence *domestication* (section 4.4) has on the user.

4.1 Control and Acceptance

The perceived control over a technical system influences how well acompanion is accepted by the user [13]. Control itself includes observability, influencability, understanding and predictability of the system. If control over a system decreases, the acceptance decreases as well. Nevertheless, different users have different demands on control over the system. While some may accept a more autonomously acting companion, others may feel decreased control over it and therefore reject the usage. This is why more automation does not automatically lead to increased comfort. The users demand on control is important to include in the companion, not only to increase the acceptance of the system, but also because technical companions influence their environment. If malfunctioning, they can cause damage in the "real world".

Another important factor on acceptance concerns dynamical changing behavior of the user while using the companion long-term. As the user is considered part of the environment to the companion, the dynamical change can occur on two levels. The first concerns unexpected changes in the sensor measured view. For example, users suddenly do not follow daily routines, because of an illness. In this case the companion system considers the preferences of the user to come up with a new plan. One step further, those preferences also change over time and need to be considered. For example, users would feel the need to control the companion before some kind of trust is established, and the user would be annoyed after a while by the repeating task of user intervention, which exists to establish control. We see that that nearly every information of the knowledge base has some kind of time constrain until user interaction is necessary. Therefore, the comfort and acceptance of the user depends on a constant balancing of assumptions and guesses of the companion system.

4.2 Relationship between trust and comfort

Comfort as used in the field of companion technology is the psychological ease that comes when mundane tasks are done and specific needs are fulfilled. Technological companions can increase comfort by undertaking these tasks and fulfill needs to an extend. Buechner, Simon and Tavani [6] state that the execution of personal tasks for somebody is basis for trust relationships among humans and is transferable to relationships between technical companion systems and users as follows.

The expectation of fullfiling a task would naturally fall on a person (this might be the user itself or someone else) is put onto technical devices. With this, it can be suggested that people react to behavior, successes or failures of a technical systems according to those of a trusted person. In general, there exist several level of trust [6], from basic initial trust in unfamiliar persons to an overall trust to something/somebody the users life depends on. These levels differ in their intensity of the included *needs* and *expectations* and *acceptance of outcome*. For example, if a person (trustor) shares a secret with another person (trustee), the trustor has the need of sharing, she has the expectation that the trustee will not give the information to any third person, and the decision to share is also influenced by the expected outcomes over the question how the trustee reacts or what the outcome of the scenario if the trustee betrays the trust and tells a third person. Furthermore, the trust relationship between humans is reciprocal. So the trustee, trusts the trustor insofar as the tasks of being a trustee, here not-third-person-sharing and reacting thoughtfully, can be accomplished by the trustee. If the trustee is, for example, put into a critical position by the trustor, by, e.g. sharing the knowledge of a crime, the trustee may feel betrayed because the accomplishment of those tasks puts the trustor in a moral or emotional dilemma. Therefore, a successful trust relationship is based on the ability of trustor to estimate risks and outcomes of trust task and the impact on the trustee, as well as ability of the trustor to accomplish this task.

In case of trust between humans and technical agents, the trustor is the user and the trustee is the technical agent. Thereby, the technical agent won't feel any betrayal, even if it can be programmed to show "emotions" that look like it. A typical trust scenario between human and technological agent can be found with a user surfing the web. The user has the need to find information about a specific topic. She trusts her browser to not share the information and she trusts other users of the same computer to not look into browser history. Furthermore, she estimates the outcome of failure of this task to be non dramatic. If the estimation of the outcome of trust betrayal changes, because maybe the user now searches for pornographic material instead of the latest news, she might consider to take further measurements in choosing a different agent, she trusts more. In this scenario, the user might choose web browser designed for privacy and anonymity.

In case of smart homes a variety of agents exists with very different tasks and with possibilities of knowledge of the most intimate areas of life. An intelligent mirror in the bathroom may process images, an intelligent bed or alarm clock evaluates sleeping habits and speech activated devices need to record and process every conversation in the room. So the user is in need to correctly estimate the trustworthiness of those devices to use them in a trusted way. If the user is not able to do that to an sufficient level she might worry over the the outcome if the trust is betrayed or react in other ways that are not beneficial in case of a comfortable feeling of "ease". One thing to increase the comfort despite this problem is to increase the control of the user [13] (see also section 4.3). In comparison to the web browser example users should have the possibility to use "another web browser" or "privacy mode", other, more "invisible" devices should at least have a power button to turn it off. But even if control is increased, the responsibility to ensure their privacy increases and comes with additional work for the user. In addition, many things happen in home environments in accidental ways. In this case, the user might not have the chance to turn devices off beforehand.

In a further step it would be interesting to find out whether always enabled high privacy mode changes the idea of being "watched" in users, so to speak, increases the trust, and in how far people which are aware of surveillance alter their behavior or even give up fighting for their own boundaries if they feel powerless over the established trust relationship.

It seems natural that a perfect technical companion would not betray trust, since this would lead to rejection or uncomfortable feelings for users. But technical companions are (up to this day) not able to understand the broad range of environmental influences as humans do and are therefore prone for failures and accidental betrayals. The reason is that all included informational sources for technical companions have to be pre-thought in some sense. Even self-learning algorithms need to have access to the needed information. So the trust that humans need to have in their devices is based on the amount and the intimacy of information they share with the device. The amount and intimacy of information is increased to increase acceptance and comfort, and with it increases the cost of betrayal for the user.

4.3 Relationship between Trust, Control and Perceived Responsibility

To achieve meaningful increase of comfort companions help out with tasks that are of *some* importance for the user. With the importance of the task, the cost of failure increases. Users expect that companions to execute their tasks sufficiently well, otherwise the companion would be useless for the user.

Trust and *control* [12] are two ways for the user, to reduce the perceived risk of failure of companion to which a task is assigned. Furthermore, trust and control don't exist solely side-by-side but influence each other [12] and are used by people if the respective other is not sufficiently available. Nevertheless, for a positive expectation a user need both, control and trust in a system.

As shown in [12], *Control* is defined as the positive expectation that comes from the influences a user has over a system. On the other hand, *trust* is the positive expectation of a user that comes from the assumption that the system act benevolently for the user. Nevertheless, control itself includes the trust, that the influence it pushes on the system is actually working, and therefore is also a trust in the bigger system how influence works. Between humans, trust can act as some kind of psychological influence, since it puts pressure onto the trustee to act according to the trustors expectations. Nevertheless, neither control nor trust can be reduced to the other.

Here we see some differences that comes with trust in technology to trust in humans similar to section 4.2. The pressure which is pushed to the trustee works for technology only indirectly. The companion itself is unable to recognize the level of trust a user has and act only according to its implemented rules. The pressure which comes through the trust works in a second step on the developers of the system. So besides the probably intimate user-companion relationship, there exist the customer-provider relationship to the developers of the system. The more intimate trust relationship to the companion can only be ensured by the manufacturer of the system (e.g. how well they implement control structures), which do not necessarily feel bound by the social pressure that comes from the shared trust between user and companion. So the risk lies with the user, and the control lies with the developer of the companion, which act in a completely different relationship. So the question is, how to establish a sustainable trust/control-relationship between companion and user.

4.4 Domestication

Domestication describes the process of adding an additional item to the life of a user. Thereby the domestication can be either seen as the item being domesticated by the user (by using it), but also by the item domesticating the user by implementing ways the item can used. Those ways of usage are implemented into the item and from an overall viewpoint, the user need to adapt to them in order to gain the advantages the item provides. The domestication process of items is studied in several papers.

The domestication of software was analyzed in [2], with the question to what extend the usage of the Whatsapp messenger changed the communication of students and how the students feel about the software. The domestication of an hardware device is analyzed in [15], by evaluating the domestication process of laptops for students on campus.

An historic example of an domestication process can be found in [11]. This paper describes the development and domestication of the first built-in kitchens ("Frankfurter Küche") in Germany in the thirties of the 20th century. The goal was to industrialize private kitchens to open up a lot of resources for the kitchen user as well as the builder of apartment blocks. Herein the domestication of the kitchen has similar goals as the technological companions and therefore the process of domestication for living environments can be compared. The kitchen of the time should shift from a center living room where eating, living and sleeping took place to a highly functional workstation.

In general, a domestication process includes several aspects the users go through until the item is successfully part of the users daily life [2, 15].

1. In the *Appropriation phase* users form knowledge and opinions about the item in question, and evaluate whether to obtain it. In the example of Whatsapp [2] the easiness of communication, the perceived user friendliness was named as reasons to use the software. Furthermore, the paper said that communication via Whatsapp is widely used (about 96 % of the students use it) and creates reliable bonds between people and a sense of *beloningness*. So we further assume that also the wish to take part in this form of social interaction is a valid reason for obtaining the Whatsapp messenger. In the example of the Frankfurter Küche [11] the manufacturers undertook an education campaign to advertise their products, as well as trying to "force" the people to use those kitchens by adjusting the ground plan of newly build apartments to not have enough space for the old kitchen styles. Similarly, the domestication process of the laptop [15] is kind of artificial placed upon the students by just "giving" the students the laptops. The decision making process as well as the work related to it, is not done by the users.

2. The *Objectification aspect* considers how an item is physically and symbollically placed in the users daily life. How it transforms from the initial obtained item to an item of the user, and further, what the personalized item means for the user ($\approx$ symbolical meaning). For example, in the Laptop [15] example it was analyzed how the laptop itself was personalized, i.e. which software was placed on the PC by the users to make it *their own* laptop. In the Whatsapp example [2] the feelings towards the messenger was surveyed, i.e. the symbolic meaning in the users life. Most of the students perceived the messenger as a "normal communication tool" rather than a "best friend" or "their life". Nevertheless, the paper does not state how important Whatsapp is in comparison to other communication tools, i.e. it was not asked if the students would be ok with changing to another communication tool. Such a question could have shown the perceived social consequences of not using Whatsapp and therefore the social importance of the messenger for the user.

3. The *Incorporation aspect* of the domestication process describes how the item is

used. This includes temporal and spatial aspects of usage. For example, the laptop from [15] diversified the students studying habits. It enabled them to learn and work at many places and at any times. Therefore, the learning process could be better adapted more flexible to the students lives in comparison to learning without a laptop. In the Whatsapp example [2] it was investigated which type of messages where send, which part of the software was mostly used and how long and often the software was used. In the Frankfurter Küche example [11] it was investigated how some the functionality of the kitchen was used. For example, it became apparent that specific kind of drawers were not used as intended (they disappeared in later kitchen furnishing), and further that some users did not use the new kitchen at all.

4. The *Conversion aspect* of domestication describes how the environment is changed by the item in further ways, e.g. how meanings and uses of the item are transformed by the user. In the Whatsapp example [2] these are questions about the usage and the effects, e.g. how much of a disruption the software is. In the Frankfurter Küche example [11] this would be the conversion from the kitchen from a living room where a lot of family life took place to a functional workplace.

For smart home companions it would be interesting, how those phases turn out. Which companions are included in the users lifes for what reasons? Which task are they supposed to fulfill and which do they fulfill? How do they influence or change the daily life of the user and how much do the user depend on the companion?

5 Conculsions

In this work we searched for questions that help to explore the effects of the interplay between companions and users. For this, we looked at the computational design of companions on the one hand, and the sociological characteristics of places in the environment that companions inhibit, on the other hand. We saw, that with increasing autonomy of companions, it is necessary to view companions not only as objects that behave in a foreseeable manner, but more like an actor with a set of tasks (similar to Actor Network Theory) in the users world. With the question, in which ways and how far the influence from the companion on the user implements in the users life.

We saw that limitations of the implementation of artificial intelligence affect users directly. With the limitation of the amount of information a companions processes, a companion is set up to make worse choices a human would make in the same situation. This leads to decreased acceptance of the companion and less comfort for the user. On the other hand, the more information is accumulated and processed, the trust relationship between user and companion is thickened. We think, this means failures of the companions are prone to be experienced more personal by the user, because his "personal life" is affected more severely.

With the increasing amount of gathered information, the question of responsibility becomes apparent. We saw that trust is a bidirectional relationship, that comes with responsibility on both ends. In inter-human relationships, the trustee feels some level of obligation to fulfill the task he or she is trusted with. A companion is a proxy in this relationship, because the responsibility of the created relationship cannot experienced by a computer, but is implemented by developers. The developers or manufacturers become the trustee of this relationship, but do not have any personal relationship to the user. So the question from this is, how does this work out? Past and current discussions about misuse of information, privacy, or manipulation of users show that users concerns are ignored all too often. How can this be prevented? Is there a way to increase the control of the user over the companion so far, that the trust relationship between companion and users is fulfilled on the trustees end? Could such a way be ethical standards for during implementation of companions, some kind of "ethical design pattern"?

6 Future Work

In future work we would like to explore the idea of more ethical designs of smart home companions. Therein, we want to gain insights about the relationship between users and companions (and maybe developers) with focus on how to include responsibility and transparency, so that not only trust and control is increased, but also the risk and price of failures, misuse and manipulation is reduced. In such a re-thinking process we need to reduce common design characteristics back to there original tasks by approaching them in an ANT-like thinking [14], and explore possibilities to have similar effects and goals other ways (similar to the critical design approach [7]).

To develop more ethical design patterns the first necessary step is to investigate problems of current approaches, e.g. which forms of betrayal by companions users experience, or which design goals contradict each other not only through their implementation but through their inherent logic. The second step is to find patterns that increase *emotional* comfort for the user in addition to the comfort from outsourcing mundane tasks to companions. Namely, increasing trust without forcing trust from the user (e.g. by using large amounts of intimate information), and increasing control without including too much additional administrative work for the user (e.g. by letting the user set privacy behavior by hand and letting him guess the effects on the program). Patterns to achieve an increased emotional comfort would probably include the collection and usage of data, the placing of companions in the users home and life, and the one-on-one interaction between companion and user.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 191–226. Springer, 2015.

[2] Annie Dayani Ahad and Syamimi Md Ariff Lim. Convenience or nuisance?: The 'whatsapp'dilemma. *Procedia-Social and Behavioral Sciences*, 155:189–196, 2014.

[3] Susanne Biundo, Daniel Höller, Bernd Schattenberg, and Pascal Bercher. Companion-technology: An overview. *KI-Künstliche Intelligenz*, 30(1):11–20, 2016.

[4] Susanne Biundo and Andreas Wendemuth. Von kognitiven technischen systemen zu companion-systemen. *KI-Künstliche Intelligenz*, 24(4):335–339, 2010.

[5] Susanne Biundo and Andreas Wendemuth. Companion-technology for cognitive technical systems. *KI - Künstliche Intelligenz*, 30(1):71–75, 2016.

[6] Buechner, J., Simon, J., and Tavani, H.T. Re-Thinking Trust and Trustworthiness in Digital Environments. *Autonomous Technologies: Philosophical Issues, Practical Solutions, Human Nature: Proceedings of the Tenth International Conference on Computer Ethics: CEPE 2013*, pages 65–79, 2014.

[7] Anthony Dunne. Hertzian tales: Electronic products, aesthetic experience, and critical design. 2008.

[8] Wolfgang Ertel. *Grundkurs Kunstliche Intelligenz*. Springer, 2 edition, 2008.

[9] Frank Honold, Pascal Bercher, Felix Richter, Florian Nothdurft, Thomas Geier, Roland Barth, Thilo Hörnle, Felix Schüssel, Stephan Reuter, Matthias Rau, et al. Companion-technology: towards user-and situation-adaptive functionality of technical systems. In *Intelligent Environments (IE), 2014 International Conference on*, pages 378–381. IEEE, 2014.

[10] Erich Hörl. Die technologische bedingung. *Beiträge zur Beschreibung der technischen Welt*, 2011.

[11] Gerd Kuhn. Die "frankfurter küche". *Auszug aus: Wohnkultur und kommunale Wohnungspolitik in Frankfurt am Main*, 1998.

[12] Guido Möllering. The trust/control duality: An integrative perspective on positive expectations of others. *International sociology*, 20(3):283–305, 2005.

[13] Sebastian Rudolph. *Akzeptanz und Kontrollverlust in Smart Home Umgebungen.* PhD thesis, University of Applied Sciences Hamburg, 2016.

[14] Ingo Schulz-Schaeffer. Akteur-netzwerk-theorie: Zur koevolution von gesellschaft, natur und technik, 2000.

[15] Hanna Vuojärvi, Hannakaisa Isomäki, and Deirdre Hynes. Domestication of a laptop on a wireless university campus: A case study. *Australasian Journal of Educational Technology*, 26(2), 2010.

[16] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.