A Decision Making Framework for Robot Companion Systems Usable by Non-Experts

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Informatik und Angewandte Kognitionswissenschaft
der Universität Duisburg-Essen
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Jens Hoefinghoff
aus
Essen

1. Gutachter: Prof. Dr. Josef Pauli
2. Gutachter: Prof. Dr. Dirk Söffker
Tag der mündlichen Prüfung: 23.06.2015

Ackowledgement

First of all, I offer my sincerest gratitude to my supervisor Prof. Dr. Josef Pauli, who has supported me with his expertise and his valuable advice throughout my time as a PhD student and research assistant at Duisburg University. His guidance allowed me to constantly broaden my knowledge and experience while I enjoyed the excellent working conditions which enabled me to complete my thesis in an appropriate period of time. I also want to thank my second referee Prof. Dr. Dirk Söffker for always showing serious interest in my work and providing me with helpful hints, ideas and insights.

Furthermore, I am grateful for having been blessed with friendly and competent colleagues. Particularly, I would like to thank Fabian Bürger, Michael Korn, and Anastasia Noglik for the inspiring exchange of knowledge and for every kind of help with managing everyday work. Furthermore, Astrid Rosenthal-von der Pütten's and Laura Hoffmann's expertise in planning, realizing, and analysing studies has been a great benefit. I am also indebted to my former colleague Michael Müller, whose great supervision during my time as a student has aroused my interest for scientific working.

Additionally, I must not forget to acknowledge all the students who made contributions to my thesis. In this context, I especially thank Laura Steinert, Jens Kapitza and Tobias Schmidt for assisting me by doing a lot of programming. Apart from that, I would like to thank Alexandra Emmerich for proofreading my thesis, which has increased the linguistic quality a lot.

Moreover, it is important for me to say that I attribute the attainment of my doctor's degree to my family, especially to my parents Gabriele and Josef Hoefinghoff. My academic success is not at last a testament of their encouraging trust in me, their support of all my pursuits and their love. Words cannot express how grateful I am for everything you have done for me. To an equal extent, my sister Nadine Brands has been a great support during the last years.

I also owe a special thanks to Laura Pauline Flake, who has not only proofread most of my publications but also gave the best possible moral

support, when doubts and frustration arose. Your contribution to my success will never be forgotten.

Last but not least, I would like to thank my friends for sharing many good times with me and for faithfully accompanying me through all stages of my thesis and of life, while allowing me the time and room I needed for myself and my work.

Abstract

In the course of becoming an increasingly important part of society, robots have also found their way into private households. For now, most robots are designed to solve only one or a few specific tasks. In the (near) future, however, robots are supposed to become companions assisting humans in their everyday life. A serious problem lies in the fact that requirements made upon robots as well as their fields of duty are largely dependent on the individual demands of the user. Due to this reason, the behaviour and the possible applications of a robot companion need to be customizable. The aim of this thesis is to develop a decision making framework for robot companions which offers solutions for the previously described challenges in the creation of robot companion systems.

First of all, a suitable decision making algorithm that is applicable for variant tasks without a multitude of parameters having to be adjusted manually is created. This is important in order to give users without programming skills or technical expertise the possibility of enhancing the capabilities of their robot to a certain extent. The developed algorithm then is evaluated in a simulation in which human decisions are compared to decisions made by the algorithm.

In addition to the evaluation made in a simulation, the decision making algorithm is implemented on the humanoid NAO robot. A modular software architecture is used in order to ensure that enhancements/modifications can be implemented without huge effort. Furthermore, the implementation provides interfaces making it possible to create new applications without programming by an XML configuration file. Based on these interfaces a tool assisting users without technical expertise in the creation of new applications is developed. Moreover, a usability study is conducted to reveal how the tool can be enhanced.

Finally, the whole approach is evaluated via two human-robot interaction studies. Those studies aim at investigating how the participants perceive the robot's decision making behaviour and if they can imagine using such a robot at home.

Contents

Chapter 1

Introduction

Companion technology increasingly finds its way into human daily life. Generally, each kind of companion technology is supposed to assist humans in coping with everyday life. In doing so, the fields of application vary from private households to different kinds of businesses. The requirements such companion systems have to fulfil are manifold and still increase which makes it necessary to implement variant human-like competences, like the recognition of emotions or the adaptability to environments and situations. To close the gap between common cognitive systems and companion technologies many open issues have to be investigated. In Wendemuth and Biundo (2012) a research project, which addresses different topics that are of great importance for the creation of companion technology, is presented. Planning and decision-making, interaction and availability, and situation and emotion are specified therein as essential aspects. According to Wendemuth and Biundo, all these research issues have to be investigated from a system's as well as from a user's perspective. Thus, a cooperation between different disciplines, like psychology or computer science, is indispensable.

As robots are supposed to live side by side with humans in the (near) future, they are a prime example for companion technology. Needless to say, that the question as to how companion systems should be constructed also applies for robots. A study about the roles and acceptance of robot companions for the home can be found in Dautenhahn et al. (2005). The results show that most of the subjects (80%) liked the idea of having computing technology for the home but less than 40% liked the idea of having a robot companion. This may result from the fact that most people do not have any experience in the exposure to robots in contrast to common computer technology. However, even the acceptance of computers has needed time to grow, since the first personal computers were presented in 1977, 36 years after Konrad Zuse built the first programmable computer (Zuse, 1993, p. 62). The acceptance of

computers increased immensely when the Commodore 64 was offered at an affordable price of 599 US$ in 1982 (Laing, 2004, p. 86-89). It can be assumed that the acceptance of robot companions also increases when they become accessible to the populace.

Concerning the role of robot companions, the study of Dautenhahn shows that some roles, such as the friend role, are less desirable ($\approx 20\%$), while other roles, like an assistant, find higher acceptance ($\approx 95\%$). Furthermore, it has been revealed that the age of the subject had an impact on the results. This already leads to a high degree of possible customizations that are difficult to be entirely put into practice. These customizations would lead to immense costs in the production which, in turn, would result in high prices for the end customer.

Beside the possible roles of robot companions the appearance, behaviour and the communication capabilities of such systems also play a key role. Mori (1970) has shown that the acceptance does not increase linearly to the anthropomorphism of a robot but that the acceptance decreases strongly from a certain level of anthropomorphic appearance. From this point, the acceptance does not increase until a very accurate image of a human being is reached. The region between the decreasing and the increasing is what Mori has called the "Uncanny Valley". Investigations by Dautenhahn et al. (2005) have shown that only 29% of the subjects stated that they preferred a human-like appearance. In contrast to that a human-like communication was desired by 71% of the subjects.

In addition to the questions concerning the appearance of and the communication with robot companions, their behaviour is an essential issue. In Dautenhahn et al. (2005) most of the subjects stated to prefer a controllable (71%) and predictable (90%) behaviour. A human-like behaviour is stated as being less desired (36%) as well as a human-like appearance (29%). However, in contrast to appearance and communication, the definition of a human-like behaviour can be more idiosyncratic. Therefore, it can be assumed that a more specific breakdown to different human-like abilities would reveal that some human attributes, like the ability to learn, find higher acceptance than others, such as defiant behaviour.

On the whole, all these findings show that there are some properties in the creation of robot companions which require a customer-specific production and others which are preferred by a bigger part of the potential consumers. To increase the acceptance of robot companions, they have to be affordable. For this purpose, a mass product allowing individual configurations by the users is needed.

1.1 Tasks and Challenges

As mentioned in the introduction, many different aspects are important regarding the creation of robot companions. In fact, a robot companion can be seen as a special kind of an artificial intelligent system. A crucial difference between common artificial intelligent systems and robot companions is that the former are mostly constructed to solve one or only a few specific tasks while robot companions should be able to deal with a greater range of tasks. The tasks which are supposed to be performed by robot companions could be subjects to change and are dependent on the owner. In the same way it is possible that different robots are meant to act differently in the same situation. Therefore, the user needs the possibility of influencing the robot's behaviour.

In the following, a framework allowing users to configure their robot companion according to their own needs is presented. Regarding the popularization of robot companions, it is a major goal that customizations can be accomplished even by users without programming skills or huge technical expertise. Furthermore, the robot should be able to adapt its behaviour through feedback while acting with the environment.

There are already numerous decision making algorithms available for artificial intelligent systems, some of which are also used or generally usable for robot companions. In addition, a lot of research has been done in the field of human-robot interaction (HRI). The results have revealed that the robot's behaviour has an influence on aspects like acceptance, likeability and so on. Due to the reason that many HRI studies are conducted based on a robot that uses scripted behaviours, it is not possible to use the information gathered from human-robot interaction studies in order to modify the robot's decision making algorithm.

Although both disciplines, computer science and psychology, work together, there is often a broken connection between both research areas. Figure 1.1 shows some of the competences of both disciplines. In the field of computer science many works focus exclusively on the creation of decision making algorithms without any consideration of human users. Therefore, the evaluations are mostly based on specific tasks which the robot has to solve successfully. In many cases, those evaluations are made exclusively in simulations as there are no implementations on a real robot available. Even when a specific algorithm has been implemented on a real robot, the evaluations rather focus on task solving by the algorithm than on user dependent aspects like acceptance. While most of the works in the field of computer science stop at this specific point, the psychologists often start with the creation of interaction scenarios without using any or only very rudimentary learning frameworks. There-

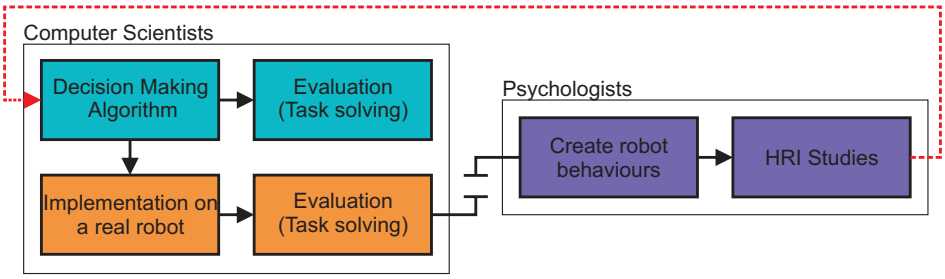

Figure 1.1: The different competences of computer scientists and psychologists in the development of robot companions.

fore, the interactions mostly consist of prewired behaviour sequences that do not include any intelligent adaption of the robot's behaviour. Of course, studies based on such interaction scenarios are important and can reveal useful information even for the creation of robot companions. However, due to the fact that a direct connection between the development of a decision making framework and HRI-studies only exists in rare cases, the results from these studies are hardly usable for adapting the algorithm. Moreover, it is absolutely essential to include the users' view of the system in order to create accepted robot companion systems.

In this thesis a decision making framework bridging the gap shown in figure 1.1 is presented. Additionally, this framework takes into account that the user himself should be able to not only adapt the robot's behaviour by feedback in given applications but to create individual applications (e.g. playing a card game) by enhancing the robot's capabilities as well (e.g. the capability of recognizing cards).

Although the usability of software for configuring robot applications is a topical field of research in the area of interactive systems, most of the created tools are developed for experienced users to support research in the field of robotics. In Kramer and Scheutz (2007) nine different robotic development environments have been evaluated in view of different aspects without including the use of the environment by inexperienced users as an aspect. The results show that for some development environments even experienced users had serious difficulties which began already at the point of the installation process and continued during the configuration of simple applications. With regard to the creation of robot companions it is important that even inexperienced users are able to handle the software for configuring robot applications.

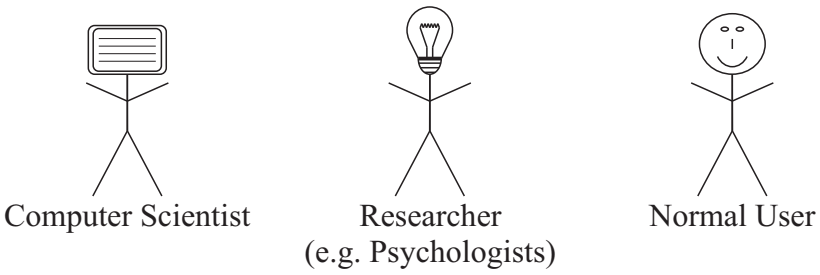

Figure 1.2: The different user types, which are able to work with different parts of the presented decision making framework.

In the following, the three user types shown in figure 1.2 will be distinguished. Each type of user should be able to perform enhancements and modifications to the decision making framework, but on different levels.

- **Computer Scientist:** representative for all academics of the computer science area or equivalent competences. Those users are able to do enhancements/modifications without any limitations.
- **Researcher:** representative for all academics outside of the computer science area (e.g. psychologists, engineer or sociologist). Those users are able to do enhancements/modifications but with minor limitations.
- **Normal User:** representative for all private users. Those users are able to do enhancements/modifications but with major limitations.

For the achievement of these aims each step of the decision making frameworks development needs to fulfil different criteria. Although there are a lot of factors which play an important role in the creation of robot companions, the main focus of this thesis is on *decision making*, *implementation architecture* and *human factors*.

The decision making behaviour is an essential part regarding the creation of robot companions. In general, a robot companion needs the ability to learn from made experiences in order to adapt its behaviour. The decision making algorithm has to find the user's acceptance and, in addition, to be able to deal with a large number of variant applications. Furthermore, it is important that new applications can be added easily without the necessity to adjust a multitude of parameters, especially as the adjusting of parameters supposes the user to understand the algorithm's internal computations. Such an assumption is contradictory to the goal that the whole framework should

be usable by users without programming skills or technical expertise (normal users). For these reasons, the algorithm has to be developed with respect to the requirements of the whole framework.

The implementation architecture has to fulfil the criterion of supporting users in enhancing e.g. the robot's behaviour and simultaneously supporting computer scientists and researchers in implementing new findings in terms of the design of robot companions. From the users' point of view the implementation architecture should offer tools to ensure a quick and easy customization of the robot. This could include the creation of new stimuli to which the robot should react or new actions which the robot is able to perform. So far, a multitude of user friendly tools are provided for different robotic platforms. Most of the tools are limited in such a way that the results are pre-wired behaviours and do not support an online adaptation of behaviour by default. Even if such mechanisms are existing, their use is often restricted to specific tasks or the configuration's effort is immense. From the researchers' points of view it is important that new findings e.g. about human-robot interaction can be integrated fast and easily. For this purpose a high grade of modularisation is necessary, as it allows computer scientists to replace specific parts when needed. Other disciplines, like psychology, would benefit from the possibility of creating new applications very fast and without being reliant on computer scientists.

Human factors play a decisive role in the creation of robot companions. Regarding technical aspects robot companion systems must be easy and intuitive to use. Additionally, properties such as likability and sympathy for the robot are further significant aspects which are not of any importance regarding the creation of common technical devices. With respect to the decision making framework it has be ensured that the users are able to comprehend and accept the robot's decisions.

1.2 Outline

In the following, a decision making framework for robot companions is described. The framework fulfils different criteria regarding to which the handling of the companion system by users without programming skills or technical expertise plays a decisive role.

In chapter 2 a suitable decision making algorithm is presented and evaluated. It is important that the method does not have any sensitive parameters,

particularly to ensure that the framework is useable by non-experts. The presented approach can be categorized into a reinforcement learning method. Its computations were inspired by a psychological theory in which the importance of emotions for the human decision making process is pointed out.

The results of the algorithm were evaluated by a comparison of human results with the results of the algorithm solving a popular gambling task. In order to gather further information about the algorithm's decisions a subsequent study which will be presented in chapter 3 was conducted. The study included a rating of decisions made by human subjects. Furthermore, the study was supposed to reveal if subjects were able to correctly identify presented decisions as having been made by a human or by the decision making algorithm.

Subsequent to the explanation of the decision making algorithm and its evaluation, the implementation of the same on the humanoid NAO robot is presented in chapter 4. Especially the description of the modular software architecture plays a decisive role in this chapter to ensure that new findings can be implemented easily. Furthermore, the interface which can be used to configure applications will be described.

Subsequent to the description of the implementation, a configuration tool is presented in chapter 5. The tool has been developed to give users without programming skills or technical expertise the possibility of configuring new applications for the robot. In order to evaluate the tool regarding usability aspects, a study with subjects from the age group 40 or older was conducted.

In addition to the evaluation of the configuration tool, also studies concerning human robot interaction aspects were conducted. These studies, presented in the chapters 6 and 7, mainly focused on revealing if the subjects were able to recognize a learning process of the robot.

Finally, the conclusion is presented in chapter 8 which consists of a summary, discussion and an outlook for further research.

Chapter 2

Decision Making Based on Artificial Somatic Markers

The first step in the development process is the creation of a decision making algorithm suitable for robot companions. In particular this means that the algorithm has to be applicable on variant applications without any sensitive parameters having to be adjusted manually. The learning algorithm presented in this chapter is a reinforcement learning method. The used computations are influenced by Damasio's Somatic Marker Hypothesis (SMH). Due to that reason, the SMH is described at the beginning of this chapter in order to provide a basis for the subsequent description of the algorithm. At the end of this chapter, the evaluation of the algorithm is presented. For evaluation purposes the Iowa Gambling Task (see section 2.2) has been used, allowing a comparison of the results of human players with the results of the algorithm. Some of the parts presented in this chapter have also been published in Hoefinghoff and Pauli (2012, 2013).

2.1 Somatic Marker Hypothesis

The Somatic Marker Hypothesis (SMH) proposed by Damasio (1994) describes the influence of emotions on the human decision making process. A multitude of research in psychology and artificial intelligence systems is based on the SMH or at least includes some parts of it. Some examples for research on different topics in the field of psychology in which the SMH has some relevance can be found in Bechara et al. (2000a, 2005); Brand et al. (2007); Ko et al. (2010); Stolper et al. (2011); Noël et al. (2013). Works that address the topic of artificial intelligent systems are for example Breazeal (1999, 2003); Thomaz et al. (2005); Hoogendoorn et al. (2009); Pimentel and Cravo (2009);

Harrington et al. (2011). These examples show that the SMH finds acceptance in research. In this thesis the SMH is used as an inspiration for the creation of a decision making algorithm which is presented in the following. Due to that reason, some key aspects of the SMH are described in this section.

In accordance with James (1884), Damasio has proposed that emotions are just bodily changes while a feeling results from becoming aware of these changes. A prominent example in this context is that humans are sad because they cry and not that they cry because they are sad. For one specific emotion several characteristic bodily changes can exist which at large are called a somatic state.

In contrast to James, whose theory does not include any mental evaluation of the situation, such an evaluation takes place in Damasio's SMH. A further difference is that James has postulated that the body is always required for the generation of feelings, while Damasio has described an additional mechanism in which the body is bypassed.

Damasio differentiates between primary emotions and secondary emotions. Primary emotions (early emotions) are inbred and can trigger very fast prewired reactions to specific stimuli, like fleeing from huge animals. Immediately after a huge animal is recognized, bodily changes which are characteristic for the emotion fear begin. This might lead to flight behaviour. After becoming aware of the bodily changes, the feeling fear arises (Damasio, 1994, p. 131-134).

Damasio's description of primary emotions corresponds to the mechanisms James has described. An essential brain region for such inbred emotions is the amygdala, which receives information about the current stimulus at first and reacts with a fast appropriate response.

As already mentioned Damasio has introduced the term secondary emotions (adult emotions) for those emotions which are learned during one's life course (Damasio, 1994, p. 134-139). Secondary emotions are grounded on personal emotional experiences and can therefore be very idiosyncratic. For example, a person makes himself looking ridiculous at a performance. From then on, the thought of entering the stage again is connected to a negative emotion for this person.

Damasio has described that an incoming stimulus is processed via the amygdala in order to create a fast response when existing (primary emotion). The situation is not further analysed within the amygdala. Simultaneously to that process, the stimulus is also processed via the ventromedial prefrontal cortex (VMC) which performs an analysis of the situation under consideration of made experiences (secondary emotions). The result of the VMC's analysis is sent back to the amygdala.

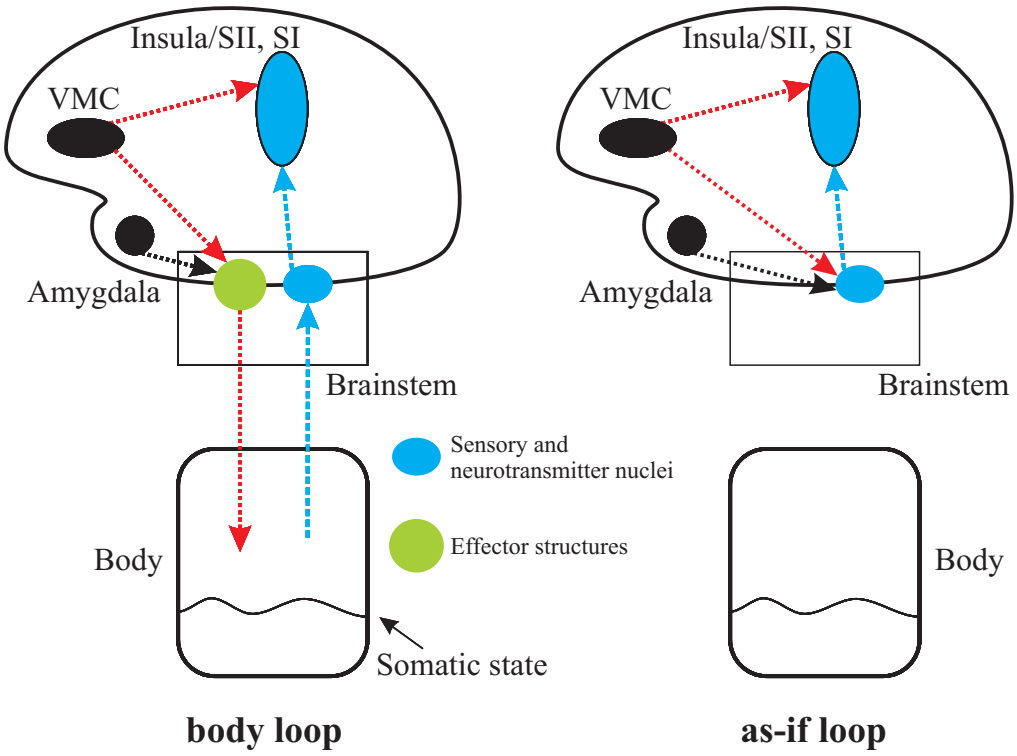

Figure 2.1: The figure shows the models which Damasio has introduced (Bechara and Damasio (2005)). The *body loop* is shown on the left hand side and the *as-if loop* on the right hand side.

In order to clarify the role of primary and secondary emotions the following example can be used. If someone opens a door on the other side of which is standing a person/friend, the amygdala creates a very fast response letting the person who opened the door wince. The VMC simultaneously analyses the stimulus with the result that no danger emanates from this person. This leads to a relaxation of the muscles that are responsible for the wince.

As mentioned before, Damasio has postulated that the inclusion of the body is not mandatory for the creation of emotions and feelings. In his opinion the brain is able to learn the fainter image of an emotional body state (Damasio, 1994, 155-158). This allows the decision making process to bypass the body as its inclusion is energy consuming and takes longer to come to a decision. Due to that reason Damasio has presented two models, the *body loop* and the *as-if loop*. Figure 2.1 shows both models.

As already explained, the amygdala is mainly responsible for primary emotions while the ventromedial prefrontal cortex manages the secondary emotions. The somatosensory cortices consisting of insula, SII and SI are able

to receive information about somatic states. If the *body loop* is active, the body gives feedback about the somatic state via the sensory and neurotransmitter nuclei to the somatosensory cortices. Due to made experiences, it is possible that the body can be bypassed which means that the *as-if loop* becomes active. In such a case, the bodily changes are going to be simulated by the VMC which sends information about the somatic state directly to the somatosensory cortices. With this mechanism the body does not have to be involved which leads to a faster processing and consumes less energy.

Up to this point, the terms primary emotion, secondary emotion, feeling and somatic state have been discussed without clarifying what a somatic marker is. Damasio has introduced the term somatic marker for those somatic states which are generated through experiences (secondary emotions) and stored by the VMC. According to Damasio, somatic markers are defined as follows:

> [...] somatic markers are a special instance of feelings generated from secondary emotions. Those emotions and feelings have been connected, by learning, to predicted future outcomes of certain scenarios. (Damasio, 1994, p. 174)

Basically, Damasio divides the human decision making process into an emotional part with a subsequent rational part. The emotional decision making part is based on the emotional memory which is consisting of the somatic markers. Every time when a specific situation is present that calls for a decision, the emotional memory is used in order to filter out options which promise to lead to a negative outcome. Damasio has described the role of somatic markers as follows:

> Somatic markers do not deliberate for us. They assist the deliberation by highlighting some options (either dangerous or favorable), and eliminating them rapidly from subsequent consideration. (Damasio, 1994, p. 174)

Therefore, the output of the emotional selection is a subset which contains only actions that promise to lead to a positive outcome. Those actions are considered in the subsequent rational decision making part. Finally, one of the remaining actions is chosen based on rational criteria. The reward that is obtained for the chosen action is used in order to update the emotional memory.

In this section some important aspects of Damasio's SMH have been presented. In summary, it can be noted that emotions play a decisive role

beside rationality in the human decision making process. Damasio has defined emotions as bodily changes (like facial expression or increasing heart rate) arising in consequence of a specific stimulus. Becoming aware of these bodily changes, finally leads to a feeling (like fear). Furthermore, it is possible that the body is bypassed and that emotions are simulated in the brain. It is distinguished between primary emotions and secondary emotions. Primary emotions are inbred and can create predefined responses to specific stimuli. From a computer scientist's view, such mechanisms can be realized by using e.g. a decision tree that is hard-coded into an artificial intelligent system. Secondary emotions are created through experiences and can be stored in the brain in the form of somatic markers. As somatic markers are updated based on experiences, static decision trees would not be suitable in order to implement this mechanism into an artificial intelligent system. Due to that reason, learning methods for artificial intelligent systems (like reinforcement learning) must be used in order to ensure that the artificial intelligent system is able to adapt its behaviour based on experiences.

Before the decision making approach based on artificial somatic markers is described, the results of the experiment which Damasio has used to support his SMH are presented in the next section. This experiment is also used in the following to evaluate the behaviour of the artificially intelligent agent.

2.2 Iowa Gambling Task

For evaluation purposes Damasio used the Iowa Gambling Task (IGT) (see Bechara et al. (1994); Damasio (1994)). Brand et al. (2006) have proposed, that two different types of decision making situations can be distinguished, which are decisions under ambiguity and decisions under risk. Decisions under ambiguity take place when there is no information, like probabilities, available in order to predict the consequences of a decision. In contrast to that, decisions under risk are made in situations which provide information allowing for the anticipation of the consequences.

The IGT is used to examine decision making in ambiguous situations as the task does not allow the subjects to predict the rewards. In this experiment a subject is given $2000 (play money, but looking like real money) and sits in front of four decks of cards. In each turn the subject can take one card from an arbitrary deck. Each card gains a benefit (positive value) but some cards also lead to a penalty (negative value). The possible amounts of the benefits or penalties are unknown to the subject. The task's goal is to increase the given amount. After every turn a visual and auditive feedback is given and the subject's amount is updated. Generally, the subjects do not have

any information about winning probabilities or loosing probabilities. This procedure is repeated 100 times which also is unknown to the subjects.

As the subjects are not allowed to take notes during the task, they are not able to calculate the net gains or losses of each deck. Accordingly, they have to rely on an emotional decision making process, based on somatic markers, in order to create an estimation as to which decks are risky and which are profitable. Every deck is prepared in a special way. The following list shows an exemplary configuration of the decks, according to Damasio's rules:

- **Deck A**: Every card gives a benefit of **$100** and **five** out of ten cards additionally have a penalty of **-$250**.
- **Deck B**: Every card gives a benefit of **$100** and **one** out of ten cards additionally has a penalty of **-$1250**.
- **Deck C**: Every card gives a benefit of **$50** and **five** out of ten cards additionally have a penalty of **-$50**.
- **Deck D**: Every card gives a benefit of **$50** and **one** out of ten cards additionally has a penalty of **-$250**.

Using this configuration, deck A and B are disadvantageous and deck C and D are advantageous decks. This is due to the reason that ten drawn cards from a disadvantageous deck lead to a net loss of $250, while ten drawn cards from an advantageous deck lead to a net gain of $250. The differences within the advantageous and disadvantageous decks are the penalties' frequency of occurrence and their magnitudes. An exemplary configuration is shown in table 2.1.

The results of the IGT presented in Damasio (1994) show that subjects search for patterns by sampling from all decks at the beginning. In the early phases of the experiment, the subjects often tend to choose from the disadvantageous decks A and B which can be explained by the immediate higher rewards. After obtaining some punishments they mostly switch to the advantageous decks C and D within the first 30 decisions. Only subjects which stated to be high-risk players made a few decisions for the disadvantageous decks in later phases of the IGT.

The overall results presented in Damasio (1994) show that people, who have no damages in regions of the brain which are responsible for creating somatic markers, avoid the disadvantageous decks and rather choose cards from the advantageous decks. A control group with ventromedial frontal patients shows a preference for cards of the disadvantageous decks. In the course of further investigations, Damasio has suggested that ventromedial

	Deck A	Deck B	Deck C	Deck D
1	100	100	50-50=**0**	50
2	100	100	50	50
3	100-250=**-150**	100	50	50
4	**-150**	100	50	50-250=**-200**
5	100	100	**0**	50
6	100	100	50	50
7	**-150**	100-1250=**-1150**	**0**	50
8	**-150**	100	**0**	50
9	100	100	**0**	50
10	**-150**	100	50	50
11	**-150**	**-1150**	50	50
12	100	100	50	50
13	100	100	50	50
14	**-150**	100	50	50
15	**-150**	100	**0**	50
16	**-150**	100	**0**	50
17	**-150**	100	50	50
18	100	100	**0**	50
19	100	100	**0**	50
20	100	100	**0**	**-200**
⋮	⋮	⋮	⋮	⋮

Table 2.1: Exemplary card decks for the IGT.

frontal patients are still sensitive to punishments but are only able to use this information for a short period of time. Therefore, they often tend to choose cards from disadvantageous decks, although they have obtained punishments for choosing this deck previously.

Subsequent studies show that the configuration of the decks has an influence on the performance of ventromedial frontal patients solving the IGT. In Bechara et al. (2000b) an alternative version of the IGT was used, with the result that no significant differences between healthy patients and ventromedial frontal patients could be observed. Numerous studies that were conducted used the IGT in order to test the performance of different patient groups like drug addicts (see Fishbein et al. (2005)) or subjects with schizophrenia (see Bark et al. (2005)). An overview of several studies can be found in Dunn et al. (2006). Later in this chapter, the IGT is used in order to evaluate the decision making algorithm which is presented in the following.

2.3 Goals and Design Criteria for the Decision Making Algorithm

Before the decision making algorithm is described in the following section, some further details about the goals and design criteria of its development

are presented in this section. Furthermore, similarities as well as differences to already existing algorithms are presented.

In this thesis the focus is on the creation of a decision making algorithm for robot companions. From a user's perspective it is important that the robot's decisions are comprehensible and acceptable. Based on the assumption that humans are generally able to comprehend decisions made by other humans (at least to a certain extend), a human-like decision making approach seems to be favourable.

Certainly, it is necessary that the decision making algorithm can be used for a broad range of tasks without a multitude of parameters having to be modified each time: otherwise users without programming skills or technical expertise are not able to do any customizations.

Summarized, the aim is to develop a decision making approach which is easily adaptable to different applications and which leads to decisions that are comprehensible and accepted by the user.

The question as to whether the inclusion of artificial emotions is advantageous for decision making algorithms has not yet been utterly solved. Picard's discussion regarding the pros and cons of generating artificial emotions in robots points out that the expectable benefit strongly depends on the application (Picard (1995)). Results from Dautenhahn et al. (2005) show evidence that most users do not prefer a human-like behaviour for robot companions. Are the findings of Dautenhahn et al. (2005) and the cons in Picard (1995) inconsistent with the assumption that a human-like decision making approach seems to be favourable for robot companions? This is not the case, as it can be assumed that there are some human-like properties which are preferred and others which are not. Obvious examples for preferred human-like abilities are the possibilities of learning and adapting the behaviour to different situations. Due to this reason the mentioned findings of Picard and Dautenhahn mainly call the idea into question that robot companions profit from the implementation of e.g. anger which might lead to acts of defiance.

There is a multitude of decision making approaches based on artificial emotions. Many of these approaches have some properties that make them unusable for the purpose of customizable robot companions, such as a high amount of a priori knowledge, a multitude of user-given parameters or a lack of an available implementation on a real robot. Some examples are given in the following listing:

1. **Often a high amount of a priori knowledge about the task and its environment is necessary.**

   Therefore, these algorithms can only be used for a limited range of tasks or environments. Especially, when the algorithm allows a high

resolution of emotions, which means that the agent is able to simulate different emotions like fear or happiness, it is necessary to define which events in the environment trigger which emotions. These definitions have to be given to the system in advance which could lead to a huge effort for each application.

In Velásquez (1998) an approach implemented on a robot dog named "Yuppy" is presented. Yuppy can have different emotions like happiness or fear which influence its decisions. Emotions are generated when specific events in the environment occur. To give an example, Yuppy becomes happy when it finds a synthetic bone or becomes fearful when it encounters darkness. All these triggers of emotions have to be defined and implemented in advance.

As a robot companion should be able to perform a huge number of tasks, the effort of using such an approach would be immense. Moreover, the needed expertise to implement the a priori knowledge is unacceptable for most users.

Further examples for algorithms which need different a priori knowledge can be found in Poel et al. (2002); Burghouts et al. (2003); Mata and Aylett (2005).

2. **The decision making algorithm itself needs a multitude of user-given parameters.**

   An example for such a parameter could be a specific threshold or a weighting parameter. Every user-given parameter is prejudicial to the creation of a user friendly framework. This especially counts for parameters with significant influence on the resulting behaviour, as they require the user to systematically estimate the consequences of choosing specific values for them which is not possible without a profound comprehension of the algorithm's computations. Furthermore, it is possible that the choice of a parameter's value is directly connected to the given task or a specific situation within the task. Accordingly, the choice of appropriate values for parameters can quickly reach an unacceptable effort.

   In Hoogendoorn et al. (2009) and Pimentel and Cravo (2009) two different decision making algorithms are presented which are also based on Damasio's SMH. Both algorithms use fixed user-given values as thresholds for the action selection. The choice of a suitable value depends on the application and requires knowledge about the decision making algorithm.

3. **There are no implementations available on a real robot.**

   Due to that reason, the evaluations of such approaches as well are only based on simulations and do not include any human robot interaction. This is sufficient to reveal if the artificial agent is able to adapt its behaviour as desired, but does not lead to any conclusion on the question whether the decisions are comprehensible by human subjects. Neither does it offer valuable clues to the issue of human subjects liking and accepting or rather disapproving the decision making behaviour. Furthermore, a lot of frameworks are developed to be used for virtual agents exclusively.

   Beside these user aspects there are also technical aspects often avoided in simulations that have to be considered in a real robot application. In many cases the evaluation scenarios follow a strictly sequential process. For example, at first the recognition of a stimulus is started, followed by the execution of a chosen action, ended with the reception of any kind of feedback. Such a sequential process leaves many open questions for a real robot application such as: how to handle incoming stimuli while a decision making process is already in progress or how to handle the execution of a new action while another action is currently being executed? Solutions for these and other problems are essential for the creation of robot companions.

   Additionally, some approaches are hard to realize at all on a physical machine. To give an example, in Mata and Aylett (2005) an action selection approach for virtual animals is presented. The approach uses artificial pheromones which are used to trigger specific emotions like fear. Therefore, the modelled animals have a virtual nose to detect pheromones and are also able to emit pheromones.

   Examples for frameworks which are evaluated exclusively in a simulation or which focus on virtual agents can be found in Poel et al. (2002); Burghouts et al. (2003); Tsankova (2009); Salichs and Malfaz (2012).

The decision making algorithm described in the following sections offers solutions for dealing with the previously listed problems. This gives users without programming skills or technical expertise the possibility of customizing their robot. As already discussed previously in this section, a high resolution of emotions is not constructive for this challenge. Therefore, the following algorithm only distinguishes between good and bad emotions but with different intensities. Basically, the algorithm can be categorized into a reinforcement learning method.

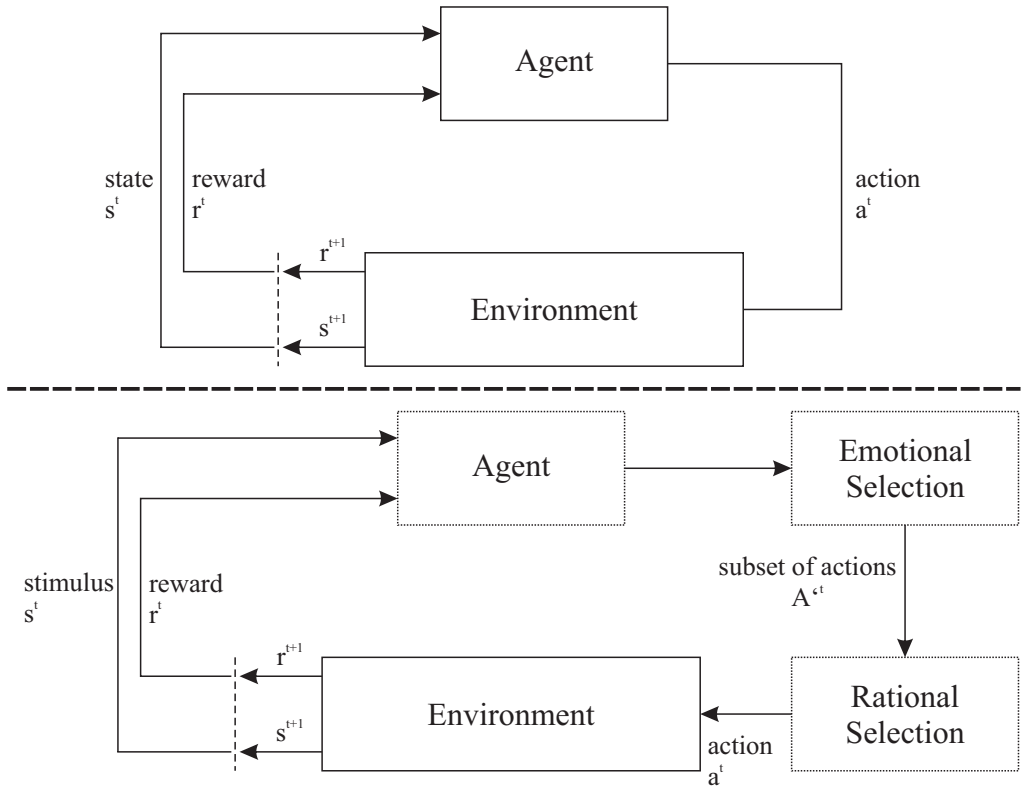

Figure 2.2: The agent-environment interface by Sutton and Barto (1998) (top) and the modified version inspired by the Somatic Marker Hypothesis (bottom).

A well-known description of reinforcement learning can be found in Sutton and Barto (1998), where the agent-environment interface shown in figure 2.2 (top) is presented. According to their model, the agent is able to receive a state $s^t \in S$ from the environment. S represents the set of all possible states. Subsequently, the agent chooses an action $a^t \in A(s^t)$, the set $A(s^t)$ thereby containing all available actions for this specific state. Finally, the agent obtains a reward $r^{t+1} \in R$ and reaches a new state s^{t+1}. Every time the agent has to make a decision the probability of a certain action being chosen is defined by the policy π^t. It applies that $\pi(s, a)$ is the probability that $a^t = a$ if $s^t = s$. The reinforcement learning method determines how the policy is changed based on experiences.

Based on Sutton's and Barto's agent-environment interface, a new agent-environment interface is developed which is inspired by Damasio's SMH (see figure 2.2 (bottom)). The modified interface divides the decision making process according to the SMH into an emotional selection part and a rational

selection part. Both are elements of the agent. The output of the emotional selection is a subset A'^t, which is selected based on the information given by the somatic markers. This step is performed by the decision making algorithm which is described in the following. In emotional inspired frameworks or in psychology the term state is often replaced by stimulus. Henceforward, the term stimulus is used. Subsequent to the emotional decision making part, one final action is chosen from the subset by further rational criteria. The rational decision part is not considered in the following. Instead, an action is randomly chosen from the subset.

Just to give an exemplary scenario for a possible inclusion of rational information, a task in which an agent is supposed to get a specific type of object can be assumed. The kind of object exists several times but is kept in different places. In this case, the emotional selection part might take into consideration all these objects as possible options, for the reward would be the same regardless of which object is chosen. At this point, the subsequent rational analysis could e.g. include the path lengths and choose the closest object.

This raises the question where emotion is included in the algorithm which is described in the following? On the one hand it becomes manifest in the implementation of the human emotional selection part according to the SMH: its output is not one single action but a set of actions promising to lead to a positive outcome. On the other hand, it is the adaption of the algorithm's parameters in order to reach a human-like behaviour.

2.4 Creation of Artificial Somatic Markers

The focus of this section lies on the computations used by the agent to process its obtained reward sequences. The resultant accumulated rewards are used in order to reflect how promising the choice of an action in consequence of a present stimulus is. In Sutton and Barto (1998) the term *return* is used for those accumulated rewards.

Based on the SMH, one somatic marker can be seen as a rating value which represents the expected outcome when an action is executed in the wake of a present stimulus. Therefore, a somatic marker is defined as the *return* of the human emotional decision making part. An artificial agent that is able to recognize different stimuli and is able to execute a defined number of actions consists of the following:

1. A set $S = \{s_1, ..., s_m\}$ which contains all stimuli that can be recognized. A stimulus can be a single signal or sensory value, but also a combination of different inputs which describes a whole situation.

2. A set $A = \{a_1, ..., a_n\}$ which contains all actions that can be executed.

Based on S and A the agent creates a matrix M (eq. (2.1)) which contains a somatic marker $\sigma_{i,j}$ for each pair of a stimulus s_i and an action a_j. The matrix M can be compared to a Q-table (see Watkins and Dayan (1992)) but the computation of its values is inspired by the SMH. The question as to how such a somatic marker $\sigma_{i,j}$ is computed is discussed in the following.

$$M_{|S|\times|A|} = \begin{pmatrix} M_1 \\ \vdots \\ M_m \end{pmatrix} = \begin{pmatrix} \sigma_{1,1} & \cdots & \sigma_{1,n} \\ \vdots & \ddots & \vdots \\ \sigma_{m,1} & \cdots & \sigma_{m,n} \end{pmatrix} = (\sigma_{i,j}) \tag{2.1}$$

The computation of a somatic marker should include already gathered experiences as well as new ones. As the system is supposed to be able to perform an online adaption of its behaviour based on obtained rewards, the algorithm can be generally categorized into a reinforcement learning approach. In terms of using the algorithm in a real robot application it is likely that the rewards are given from a human with whom the robot is interacting. Due to this fact, the owner can affect the robot's behaviour which is, in accordance with the results of Dautenhahn et al. (2005), an important aspect regarding robot companion systems.

The chosen computations of a somatic marker should fulfil the criteria listed below. Clarifications about the importance of each of these criteria as well as their considerations in the chosen computations are presented in the following.

1. **The update process should include new knowledge[1] and already collected knowledge.**

2. **Frequently obtained rewards with a lower magnitude should have the same impact on the decisions as a single reward with a higher magnitude.**

3. **The weightings of new and collected knowledge should be adapted to the situation.**

4. **The agent should be able to perform reversal learning.**

[1]Knowledge is obtained from gathered rewards.

Criterion 1: The update process should include new knowledge and already collected knowledge. Needless to say that the human decision making process does not exclusively consider the last obtained reward but also considers a history of rewards.

Every time the agent obtains a reward $r_{i,j}$, for an executed action a_j in consequence of stimulus s_i, the corresponding somatic marker $\sigma_{i,j}$ is updated. It depends on the stimulus which rewards are even possible. This is to say that, the consequences of a decision concerning the choice of a job are different from e.g. the decision which potato chips to buy.

Every obtained reward is a component of an infinite time series. Therefore, an obvious starting point for the computation of somatic markers is the exponential smoothing function. Equation (2.2) shows the computation of the exponential smoothing (Winters (1960)).

$$\overline{r_{i,j}^t} = \Delta \cdot r_{i,j}^t + (1 - \Delta) \cdot \overline{r_{i,j}^{t-1}} \ , \quad r_{i,j}^t \in R_{s_i}, \quad \Delta \in [0, 1] \tag{2.2}$$

This computation fulfils the combined request of including new knowledge $r_{i,j}^t$ and collected knowledge $\overline{r_{i,j}^{t-1}}$. There is also a weighting of new and collected knowledge but there is no adaption of the weighting parameter Δ. At the initial state, at which the agent does not have access to any knowledge, it would be preferable to weight new knowledge exclusively or at least to a higher extent than collected knowledge. In contrast to that, the weighting of collected knowledge should increase when the agent gathers reliable information in order to reduce the impact of outliers in the rewards. When the agent has reliable collected knowledge it is necessary to not exclusively consider collected knowledge but to take into account new knowledge as well. Otherwise, the agent is not able to relearn, because any new reward would be discarded.

In addition to the problem that a fixed weighting is used, the convergence of the resultant value can also be disadvantageous for the representation of knowledge. Under the assumption that a set of rewards R_{s_i} exists for every stimulus and that the initial value $\overline{r_{i,j}^{\text{init}}} \in [\min(R_{s_i}), \max(R_{s_i})]$, the upper bound for $\overline{r_{i,j}^t}$ is defined by $\max(R_{s_i})$, while the lower bound is defined by $\min(R_{s_i})$, when an exponential smoothing is used. Therefore, rewards with values lying in the interval $]\min(R_{s_i}), \max(R_{s_i})[$ are not able to reach the upper or lower bound.

Figure 2.3 illustrates an exemplary case with a constant obtained reward $r_{i,j}^t = 50$ and a set $R_{s_i} = \{-100, -50, +50, +100\}$. It is observable that the impact of the obtained rewards is small at the beginning and negligible after some iterations. Furthermore, the best case value +100 cannot be reached. Due to this fact, a single negative reward (e.g. at t=100) would have a major impact on the value $\overline{r_{i,j}^t}$, even if 100 positive rewards were obtained previously.

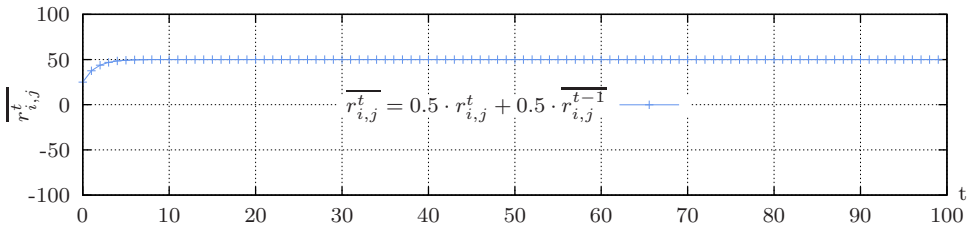

Figure 2.3: Exemplary case that shows the convergence problem when an exponential smoothing is used ($\overline{r^{\text{init}}_{i,j}} = 0$).

In the same way, frequently obtained negative rewards ($r^t_{i,j} = -50$) do not lead to the worst case value of -100.

The convergence is unproblematic in the case that a binary set of rewards is used ($R_{s_i} = \{-v, +v\}$). In such a case $-v$ is interpreted as worst case and $+v$ as best case. However, for human daily decisions there are often several advantageous and disadvantageous options and the positive or negative outcomes are of different quality as well. To express such different qualities R_{s_i} can be defined as follows: $R_{s_i} = \{-v_1, \cdots, 0, \cdots, +v_o\}$. It might happen that some actions won't ever lead to the maximal or the minimal reward that is possible. This fact leads to the SM not converging against the maximum or minimum value but against another level, even if the frequency of e.g. positive rewards is very high. Dependent on the task, it can be important that even actions which lead to rewards with a small magnitude adapt the somatic marker in such a way that the best/worst case value can be reached. Otherwise, single outliers in the rewards have too much influence on the computation of collected knowledge. To construct an example, the following configuration is assumed:

- $S = \{s_1\}$
- $A = \{a_1\}$
- $R_{s_1} = \{-1000, -50, +50, +1000\}$
- $\overline{r^{\text{init}}_{i,j}} = 0$

Figure 2.4 shows an example in which the best case (+1000) and the worst case (-1000) never occur. The obtained rewards can be found on top of the figure. Until $t = 18$ the agent has obtained only positive rewards (+50) followed by two negative rewards (-50) at $t = 19$ and $t = 20$. In order to show the influence of the fixed weighting on the result, three different weightings

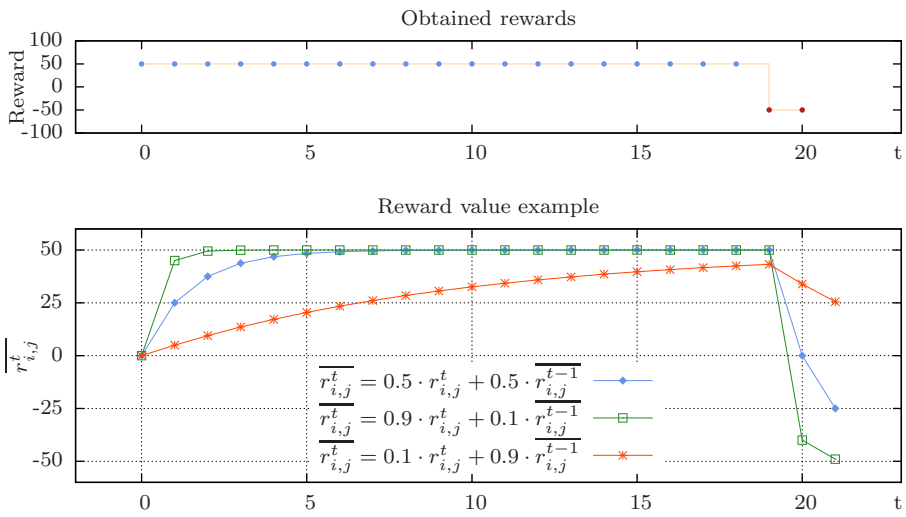

Figure 2.4: Exemplary case that shows the influence of different weightings when an exponential smoothing is used.

are used for the computations. The used weightings are an equal weighting of new knowledge $r_{i,j}^t$ and collected knowledge $\overline{r_{i,j}^{t-1}}$, a weighting with a high consideration of new knowledge and a weighting with a high consideration of collected knowledge. It can be noticed that even when 19 positive rewards are obtained in a row the best case value +1000 is not reached due to the convergence. In consequence, the two negative rewards at $t = 19$ and $t = 20$ affect the value so much that the option is marked as being disadvantageous (negative value). The case in which new knowledge is weighted with 0.1 is an exception. However, a low weighting of new knowledge leads to the problem that the agent is not able to adapt its behaviour to new situations quickly.

In summary, the influence of the history of rewards is not sufficient to prevent that a few outliers change $\overline{r_{i,j}^t}$ drastically. A high weighting of collected knowledge counteracts this problem but leads to further problems. The modifications of the computations presented in the next paragraph allow that $\overline{r_{i,j}^t}$ does not converge against 50 but against 1000 in such a case. This decreases the influence of outliers.

Furthermore, the example shows that the used weightings have a major influence on the result. As discussed previously, there are always cases in which a chosen fixed weighting is unfavourable. Therefore, the following modifications also include an adaptive weighting.

Case	$\overline{r_{i,j}^{t}}$
Unreliable knowledge	$1 \cdot r_{i,j}^{t} + 0 \cdot \overline{r_{i,j}^{t-1}}$
Reliable knowledge	$1 \cdot r_{i,j}^{t} + 1 \cdot \overline{r_{i,j}^{t-1}}$
Learning period	$w \cdot r_{i,j}^{t} + \hat{w} \cdot \overline{r_{i,j}^{t-1}}, \quad w + \hat{w} \in]1;2[$

Table 2.2: Desired weightings for new and collected knowledge.

Criteria 2 and 3: The consideration of the rewards' frequency and adaptive weightings are important to overcome the previously discussed problems. As shown before, the exponential smoothing does not fulfil all desired criteria which are listed at the beginning of this section. Therefore, a modification is necessary, especially to consider a higher influence of frequently obtained rewards. Furthermore, the example shown in figure 2.4 reveals that the used weightings have a major influence on the result.

Therefore, the question is how to set the weighting parameters, especially as a fixed weighting can be unfavourable at some situations. For example, there is no need to include collected knowledge at the beginning, unless some prior knowledge is given manually to the system. To solve this problem the weightings have to depend on the current situation or more precisely on the reliability of collected knowledge. Generally it can be distinguished between three different cases:

1. The agent does not have any collected knowledge or collected knowledge is unreliable.

2. The agent is sure about its behaviour due to the made experiences which makes collected knowledge most reliable.

3. The agent is in the learning phase, which means that the agent is not absolutely sure whether the collected knowledge is reliable. Nonetheless, the made experiences can be used in order to adapt its behaviour.

In addition to an adaptive weighting, the new computation should give frequently obtained rewards with smaller magnitudes the possibility of leading to the maximum or minimum value. Under consideration of the three different cases listed before, the computation is modified to realize the weightings that are shown in table 2.2. Here, w is defined as the weighting of new knowledge and $\widehat{w}$ is defined as the weighting of collected knowledge.

In contrast to the exponential smoothing for which the sum of the weightings is 1, the sum of the weightings for the new computation lies in the interval $[1;2]$. Consequently, the resultant value can increase/decrease continuously. A closer look at the different cases makes the chosen computation

more perspicuous. In the case that collected knowledge is unreliable, it is very obvious that only new knowledge is considered. It is rather uncommon that in the case that collected knowledge is reliable, both, new and collected knowledge, are weighted with the maximum value. Here, an exclusive consideration of collected knowledge would make reversal learning impossible as newly obtained rewards would not have any influence on the computation. Additionally, it is very likely that new knowledge and collected knowledge are corresponding when collected knowledge is rated as most reliable. If not, it depends on the reward's magnitude whether the somatic marker is affected much. When e.g. an action is marked as most promising, a small negative reward does not have a huge impact on the computation and the action still is marked as being advantageous. A striking example is that someone who always travels by plane won't stop doing this because the last flight had some minor turbulences (small negative reward). In contrast, the person may use the train for the next travel when this person survived a plane crash (high negative reward). Of course collected knowledge can be categorized neither as most reliable nor unreliable but something in between, which is denoted as learning period.

To reach the desired weightings shown in table 2.2, the exponential smoothing (see eq. (2.2)) is modified. In order to make the weightings dependent on the reliability of collected knowledge a value κ_i is introduced for each stimulus (see eq. (2.3)). The value κ_i defines the reliability of collected knowledge. The value c is a constant which defines the rewards' resolution. Collected knowledge is rated as unreliable when $\kappa_i = 0$ and as reliable when $\kappa_i = c \vee \kappa_i = -c$. In the following the meaning of the rewards' resolution as well as the computation for the reliability κ_i are presented.

$$\vec{\kappa} = \begin{pmatrix} \kappa_1 \\ \vdots \\ \kappa_m \end{pmatrix}, \quad \kappa_i \in \mathbb{Z} \wedge [-c, c], \quad c \in \mathbb{N}^+ \tag{2.3}$$

The rewards' resolution is important for the computation of the reliability and classifies different rewards into $2 \cdot c + 1$ equivalence classes. After every obtained reward the reliability is updated with the computation shown in eq.

class	rewards	class	rewards
-1	$[-100; 0[$	1	$]0; 100]$

class	reward
0	0

Table 2.3: Resultant equivalence classes when c=1 is used.

(2.4) in which $r_{max_i} = \max\left\{|r| \,\middle|\, r \in R_{s_i}\right\}$. To ensure that the new value lies in the interval $[-c, c]$, eq. (2.5) is added.

$$\widehat{\kappa_i^{t+1}} = \begin{cases} \kappa_i^t, & \text{if } r_{i,j}^t = 0 \vee r_{i,j}^{t-1} = 0 \\ \kappa_i^t + \left\lceil \frac{r_{i,j}^t}{r_{max_i}} \cdot c \right\rceil, & \text{if } r_{i,j}^t > 0 \\ \kappa_i^t + \left\lfloor \frac{r_{i,j}^t}{r_{max_i}} \cdot c \right\rfloor, & \text{if } r_{i,j}^t < 0 \end{cases} \tag{2.4}$$

$$\kappa_i^{t+1} = \begin{cases} -c, & \text{if } \widehat{\kappa_i^{t+1}} < -c \\ c, & \text{if } \widehat{\kappa_i^{t+1}} > c \\ \widehat{\kappa_i^{t+1}} & \text{else} \end{cases} \tag{2.5}$$

In order to consider a small history, the change of κ_i depends on the current reward and the last reward. Only if none of these are neutral (0), the value κ_i is changed based on the current reward. The basic idea behind this definition is that neutral rewards do not give sufficient information to draw conclusions about the collected knowledge's reliability. Therefore, κ_i is only changed when a positive trend (two consecutive positive rewards), a negative trend (two consecutive negative rewards) or an opposed trend (two consecutive rewards with opposed algebraic signs) occurs. The value of κ_i increases in case of a positive reward and decreases in case of a negative reward κ_i. Furthermore, the value by which κ_i is increased or decreased depends on the reward's magnitude but is at least 1 due to the rounding. In order to clarify the meaning of the constant c, the following two exemplary cases are assumed. Both share the same set R_{s_i} but use a different constant c.

1. $R_{s_i} = \{-100, -20, 0, 10, 25, 50\}, \quad c = 1$
2. $R_{s_i} = \{-100, -20, 0, 10, 25, 50\}, \quad c = 10$

Based on these configurations the rewards are categorized into equivalence classes which are defined as shown in table 2.3 and 2.4. It is observable that

class	rewards	class	rewards
-10	$[-100; -90[$	10	$]90; 100]$
-9	$[-90; -80[$	9	$]80; 90]$
-8	$[-80; -70[$	8	$]70; 80]$
-7	$[-70; -60[$	7	$]60; 70]$
-6	$[-60; -50[$	6	$]50; 60]$
-5	$[-50; -40[$	5	$]40; 50]$
-4	$[-40; -30[$	4	$]30; 40]$
-3	$[-30; -20[$	3	$]20; 30]$
-2	$[-20; -10[$	2	$]10; 20]$
-1	$[-10; 0[$	1	$]0; 10]$

class	reward
0	0

Table 2.4: Resultant equivalence classes when c=10 is used.

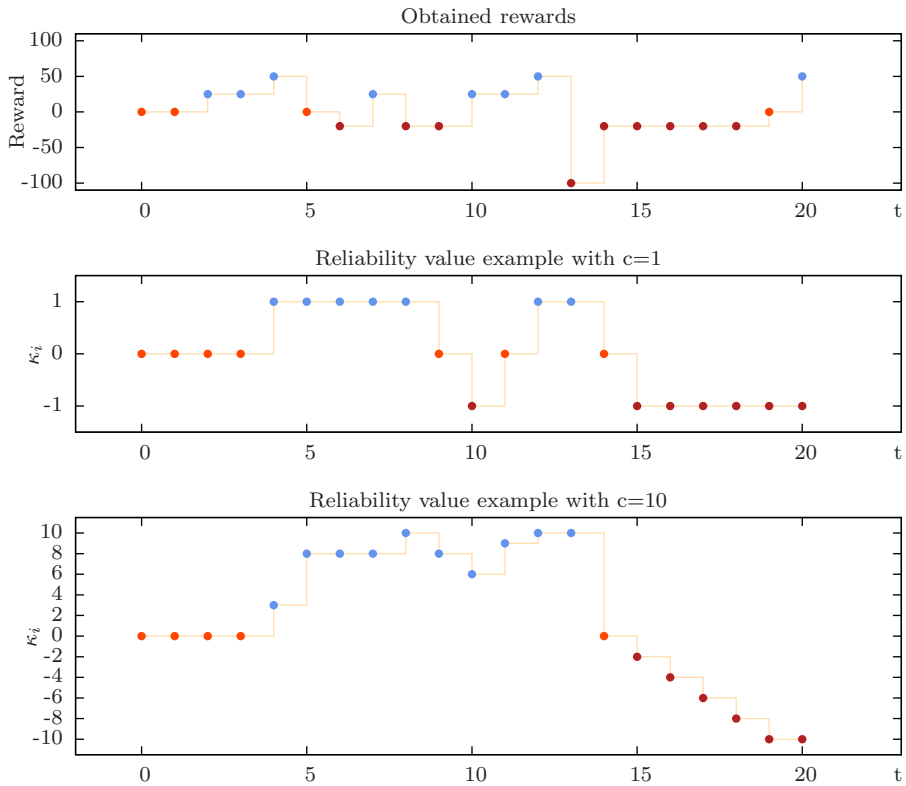

Figure 2.5: Exemplary case to show the meaning of the constant c for the computation of the reliability κ_i.

the higher value c=10 allows for a more detailed resolution of rewards, while the rewards' magnitudes have no meaning when c=1 is chosen.

An exemplary case for the given configurations is illustrated in figure 2.5. It is observable that the first change of κ_i occurs after the fourth reward has been obtained, because at this point the condition $r_i^t = 0 \vee r_i^{t-1} = 0$ is not fulfilled for the first time. Consequently, collected knowledge is marked as most reliable for the configuration with $c = 1$, while for the other configuration ($c = 10$) collected knowledge is indeed rated as more reliable but does not directly reach the maximum reliability value.

Another interesting incident is shown at t=8 and t=9. Here, the agent received two consecutive negative rewards with a small magnitude compared to the scaling value $r_{max_i} = 100$. Thus, when c=1 is used, collected knowledge is already marked as unreliable after the first negative reward. In contrast to that, collected knowledge is still marked as reliable when $c = 10$ is used. The second negative reward makes κ_i decrease to $-c$ when $c = 1$. Therefore, collected knowledge is now marked again as most reliable, for even negative experiences can lead to reliable knowledge. When $c = 10$ is used, κ_i decreases another time by the corresponding value which is defined through the equivalence classes (see table 2.4). In case that $c = 10$, the impact of the reward's magnitude is observable at t=13 and t=14. While a reward $r_i^{13} = -100$ changes κ_i by -10, the smaller reward $r_i^{14} = -20$ leads to a smaller change by -2.

It can be summarized, that a higher value for c makes the reliability of collected knowledge more stable against single outliers (see $t = 8$ and $t = 9$), while a smaller value might be advantageous in reversal learning cases requiring collected knowledge to be discarded quickly.

Based on the computed reliability the desired weighting, illustrated in table 2.2, is realized through a modification of the exponential smoothing function. The modified computation can be seen in equation (2.6) in which μ and λ are defined as two quadratic functions (2.7) (2.8) each of which is dependent on κ_i and c.

$$\overline{r_{i,j}^t} = \underbrace{(1 - (\mu \cdot \lambda)) \cdot r_{i,j}^t}_{\text{new knowledge}} + \underbrace{(\mu + (\mu \cdot \lambda)) \cdot \overline{r_{i,j}^{t-1}}}_{\text{collected knowledge}} \tag{2.6}$$

$$\mu = \frac{(\kappa_i)^2}{(c)^2}, \quad \kappa_i = [-c; +c] \tag{2.7}$$

$$\lambda = -\frac{(\kappa_i)^2}{(c)^2} + 1, \quad \kappa_i = [-c; +c] \tag{2.8}$$

The result for μ becomes 0.0, if collected knowledge is unreliable and reaches its maximum $\mu = 1.0$, when collected knowledge is rated as most

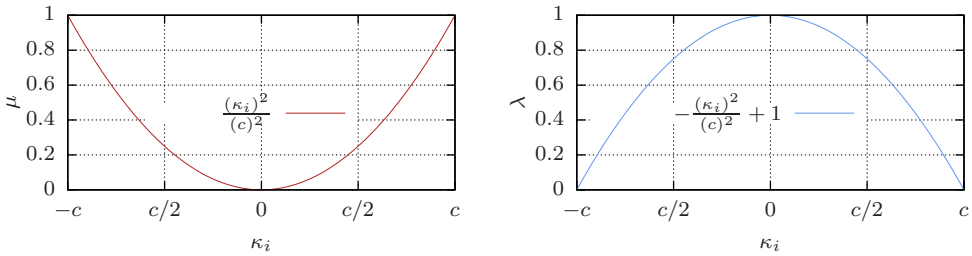

Figure 2.6: Quadratic functions (μ and λ) which are used to compute the weightings.

reliable ($\kappa_i = c \vee \kappa_i = -c$). In contrast to that, λ becomes 0.0, if collected knowledge is most reliable and becomes 1.0, when collected knowledge is unreliable. Furthermore, it applies that ($1 - \mu = \lambda$). Both functions are illustrated in figure 2.6. Due to the chosen combinations of μ and λ for the weightings, two biquadratic functions are created. The weighting function for new knowledge w can be seen in eq. (2.9), while the weighting function for collected knowledge $\widehat{w}$ can be seen in eq. (2.10). The addition of both weightings is shown in eq. (2.11).

Figure 2.7 shows a plot of each function and also the sum of the weightings. The newly created weighting functions fulfil all criteria of the desired weighting shown in table 2.2. It is observable that new knowledge is always considered with a different weighting, while the weighting of collected knowledge increases, when it is rated as more reliable. In the case that collected knowledge does not exist or is unreliable, new knowledge is used exclusively for the computation. When collected knowledge is rated as most reliable ($\kappa_i = -c$ or $\kappa_i = c$), the highest possible weightings are used for new knowledge as well as for collected knowledge. The preliminary decreasing of the weighting of new knowledge during the learning period counteracts the influence of single outliers in the rewards.

$$
\begin{aligned}
w = f_{new}(\kappa_i, c) &= 1 - (\mu \cdot \lambda) \\
&= 1 - ((\frac{(\kappa_i)^2}{(c)^2}) \cdot (-\frac{(\kappa_i)^2}{(c)^2} + 1)) \\
&= 1 - (\frac{(\kappa_i)^2}{(c)^2} - \frac{(\kappa_i)^4}{(c)^4}) \\
&= \frac{(\kappa_i)^4}{(c)^4} - \frac{(\kappa_i)^2}{(c)^2} + 1 \qquad (2.9)
\end{aligned}
$$

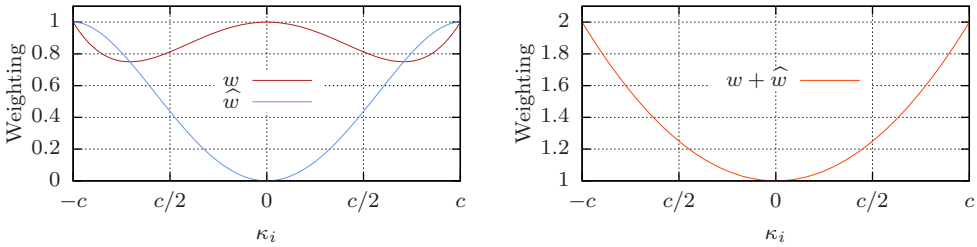

Figure 2.7: Weighting functions w, $\widehat{w}$ and the sum $w + \widehat{w}$.

$$\begin{aligned}
\widehat{w} = f_{old}(\kappa_i, c) &= \mu + (\mu \cdot \lambda) \\
&= \frac{(\kappa_i)^2}{(c)^2} + ((\frac{(\kappa_i)^2}{(c)^2}) \cdot (-\frac{(\kappa_i)^2}{(c)^2} + 1)) \\
&= \frac{(\kappa_i)^2}{(c)^2} + (\frac{(\kappa_i)^2}{(c)^2} - \frac{(\kappa_i)^4}{(c)^4}) \\
&= -\frac{(\kappa_i)^4}{(c)^4} + 2 \cdot (\frac{(\kappa_i)^2}{(c)^2})
\end{aligned} \tag{2.10}$$

$$\begin{aligned}
w + \widehat{w} &= f_{new}(\kappa_i, c) + f_{old}(\kappa_i, c) \\
&= (1 - (\cancel{\mu \cdot \lambda})) + (\mu + (\cancel{\mu \cdot \lambda})) \\
&= 1 + \mu \\
&= \frac{(\kappa_i)^2}{(c)^2} + 1
\end{aligned} \tag{2.11}$$

All these changes lead to the fixed weighting being replaced by an adaptive weighting which is based on the reliability of collected knowledge. Furthermore, the sum of the weightings lies now in the interval $[1; 2]$ which makes it possible that frequently obtained rewards with smaller magnitudes lead to a successive increasing of the resultant value.

Especially for the adaption of the weightings many justifications have been presented previously. At this point it must be noticed that there are of course different suitable weighting functions which might be applied. Alternative weighting functions could be used to create artificial agents with different characteristics, although some specific characteristics already can be modelled via the rewards' resolution c. However, it requires at least an implementation on a real robot followed by HRI studies to perform a significant evaluation of the extent to which a particular function is suitable for the application on a

robot companion in comparison to other functions. Due to these reasons, the implementation and evaluations of different weighting functions are a point of contact for further investigations.

To clarify the effects of adaptive weighting functions, the same example as shown in figure 2.4 is used. Figure 2.8 displays the performance of the adaptive weighting functions, including different values for the reward resolution (c). Although the obtained rewards in the example are either +50 or -50, the set of rewards still is defined as $R_{s_1} = \{-1000, -50, +50, +1000\}$. This is important for the definition of the equivalence classes.

In contrast to the results of the exponential smoothing, it is observable that the value $\overline{r_{i,j}^t}$ can increase continuously when the adaptive weighting functions are used. Furthermore, it can be seen that a higher value for the parameter c leads to more consecutive rewards being needed until the maximum reliability is reached. A maximum reliability, in return, results in the maximum weighting of new and collected knowledge ($1 \cdot r_{i,j}^t + 1 \cdot \overline{r_{i,j}^{t-1}}$). The effect of a higher rewards' resolution is that rewards with smaller magnitudes are categorized into an equivalence class closer to 0 than rewards with higher magnitudes. This results in a single reward with a lower magnitude not having a huge impact on the computation of the reliability κ_i and on the computed weightings respectively.

In the following, the hyperbolic tangent (tanh) is used additionally, in order to limit the function's result. The smoothed value is used as input ($\tanh(\overline{r_{i,j}^t})$). The function's result lies in the interval $]-1, +1[$, whereupon a negative value indicates a negative emotion, while a positive value indicates a positive emotion. There are also two additional characteristics of the hyperbolic tangent which are advantageous: first, the high gradient for inputs close to zero, allowing the agent to perform an explicit categorization in order to determine if a decision was good or bad even after only a few decisions. Second, the decreasing gradient for higher or lower inputs which ensures that consolidated knowledge is resistant to fluctuation.

Although the hyperbolic tangent is defined for inputs of the interval $[-\infty, \infty]$, it is advisable to limit the input in order to avoid computational errors. Therefore, each incoming reward is scaled as shown in equation (2.12). Due to the scaling, the best case is defined by $+\max\left\{|r| \;\middle|\; r \in R_{s_i}\right\}$ and the worst case is defined by $-\max\left\{|r| \;\middle|\; r \in R_{s_i}\right\}$. The reason why π is chosen, is that $\tanh(\pi) <\approx 1$. Here, also another value (preferable $> \pi$) could be used instead.

Due to the summation of new and collected knowledge it is possible that the hyperbolic tangent's input $\overline{r_{i,j}^t}$ lies outside the interval $[-\pi, \pi]$. Therefore, the computed value of $\overline{r_{i,j}^t}$ is limited as shown in equation (2.13). This ensures

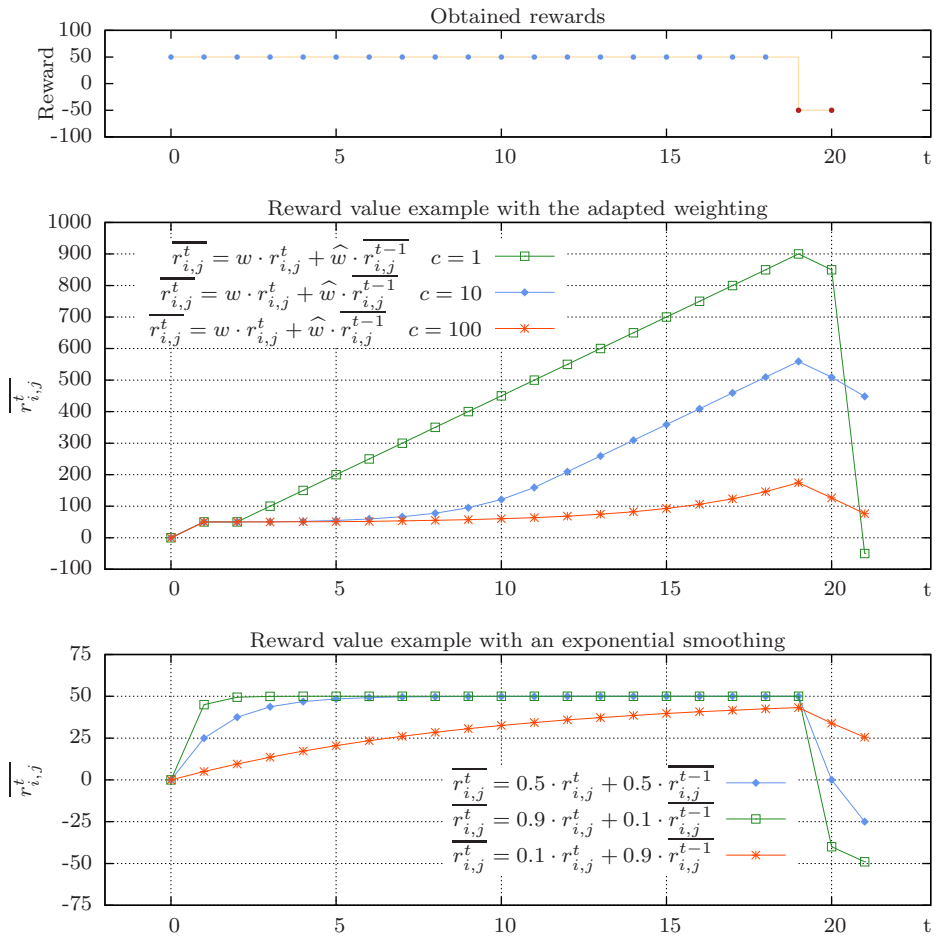

Figure 2.8: Performance of the adaptive weighting compared to the exponential smoothing with different values for the reward resolution (c).

that π is the upper bound and $-\pi$ is the lower bound for the hyperbolic tangent's input.

$$\widetilde{r^t_{i,j}} = \frac{r^t_{i,j}}{\max\left\{|r| \;\middle|\; r \in R_{s_i}\right\}} \cdot \pi \tag{2.12}$$

$$\overline{r^t_{i,j}} = \begin{cases} \pi & \text{, if } w \cdot \widetilde{r^t_{i,j}} + \hat{w} \cdot \overline{r^{t-1}_{i,j}} > \pi \\ -\pi & \text{, if } w \cdot \widetilde{r^t_{i,j}} + \hat{w} \cdot \overline{r^{t-1}_{i,j}} < -\pi \\ w \cdot \widetilde{r^t_{i,j}} + \hat{w} \cdot \overline{r^{t-1}_{i,j}} & \text{, else} \end{cases} \tag{2.13}$$

Finally, a somatic marker $\sigma_{i,j}$ is computed as shown in equation (2.14). To compute $\overline{r^{t-1}_{i,j}}$, the inverse hyperbolic tangent is used. The resultant

somatic marker $\sigma_{i,j}^{t+1}$ is used to assist the next decision making step as it gives information about the expected outcome for executing an action a_j in consequence of a stimulus s_i.

$$\begin{aligned}\sigma_{i,j}^{t+1} &= \tanh(\overline{r_{i,j}^{t}}) \\ &= \tanh(w \cdot \widetilde{r_{i,j}^{t}} + \hat{w} \cdot \overline{r_{i,j}^{t-1}}) \\ &= \tanh(w \cdot \widetilde{r_{i,j}^{t}} + \hat{w} \cdot \tanh^{-1}(\sigma_{i,j}^{t})) \end{aligned} \tag{2.14}$$

Summarized, somatic markers represent the accumulation of rewards. How these are used to influence the decision making process is shown in the next section.

Criterion 4: The agent should be able to perform reversal learning. In order to take into account that the environment in which a robot companion is operating as well as the desired behaviour of a robot companion are subjects of change, it is of great importance that the agent is able to solve reversal learning tasks. Some remarks concerning reversal learning abilities have already been made previously in this section. The most important aspects are elucidated again for a better comprehension, starting with the reliability κ_i and the used weightings based thereon. Due to the weightings shown in figure 2.7 it is assured that new rewards are always considered, even if collected knowledge is rated as most reliable. In doing so, the agent is able to adapt its behaviour when the common stimulus-response has changed.

During reversal learning tasks it is likely that the reliability κ_i reaches the value 0 more often than in common learning task. This is due to the reason that the reliability usually increases, when the agent starts learning, and decreases when the environment has changed, which results in negative rewards being obtained for an action that was rated positively before. This process is repeated every time when a change of the environment leads to opposed rewards. The closer the reliability κ_i comes to the value 0, the lower is the weighting of collected knowledge for the computation (see figure 2.7). This supports the robot's ability to relearn.

Furthermore, the reward resolution c plays a role, especially in reversal learning task. As shown in figure 2.5, reversal learning tasks might profit from a lower reward resolution. A lower value for c leads to rewards with lower magnitudes having a higher impact on the reliability κ_i. Consequently, collected knowledge can be discarded more quickly.

Summary of the creation of artificial somatic markers: At this point, the computation of somatic markers presented in this section is finished. As

has been pointed out, somatic markers represent the emotional memory of the agent in so far that they are used for the emotional decision making part according Damasio's SMH. One somatic marker exists for each pair of a stimulus s_i and an action a_j, thereby representing a value for the accumulation of obtained rewards based on the computation shown in equation (2.14).

All values needed for the computation, except for the reward resolution c, are adapted automatically. This is important with regard to the purpose of creating customizable robots. The influence of the rewards' resolution on the computation has already been discussed but is examined in more detail in the evaluation section at the end of this chapter.

Having completed the creation of somatic markers, it is still necessary to define how the information is used in order to assist the agent's decision making process. Therefore, the next section covers the selection mechanisms based on the somatic markers.

2.5 Deciding based on Artificial Somatic Markers

While the computation of somatic markers has been presented in the previous section, this section deals with the question as to they are used to assist the decision making. Based on the SMH, the output of the emotional selection is a subset $A' \subseteq A$ that contains all actions which promise to lead to a positive outcome. Therefore, a mechanism defining the resultant subset is needed. For that purpose, in the following a threshold is introduced, in order to determine which actions are selected for being included in the subset ($a \in A'$). In contrast to comparable algorithms (Hoogendoorn et al. (2009); Pimentel and Cravo (2009)), the threshold presented in this thesis is adaptive in order to work with different tasks without that a manual adjustment being required. An independent threshold θ_i is created for each stimulus s_i (eq. (2.15)). Each threshold is interpreted as a frustration level concerning a specific situation, whereupon a lower value represents a higher frustration.

$$\vec{\theta} = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_m \end{pmatrix}, \quad \theta_i \in]-1;1[\tag{2.15}$$

Before the computation of the thresholds as well as the selection rules based thereon are described, a short overview about the basic idea behind the adaptive threshold is given. Basically, one threshold expresses the frustration concerning a specific situation. Oversimplified it could be said, that when

humans try to solve a problem unsuccessfully, they might get frustrated and take into account further options. In contrast, if they find a solution, they often stick to their usual decisions because a further consideration of other options is not necessary.

For the agent this means, that if its made decisions in specific situations usually lead to positive rewards, the agent is less frustrated. A lower frustration results in a high threshold, as it is not necessary to take actions into account that are obviously less promising. However, if the usual decisions suddenly lead to negative rewards, the agent becomes frustrated which results in a low threshold to ensure that other actions are considered.

Generally, the frustration levels as well as the somatic markers are dependent on the obtained rewards. Therefore, the same computation as for the somatic markers is used for the frustration levels (see eq. (2.16)). The index $_j$ actually is not needed for the computation here but shows that the computation of a somatic marker and the frustration level are based on the same reward. In contrast to a somatic marker $\sigma_{i,j}$, which is only updated, when a combination of a specific stimulus s_i and an action a_j is present, the frustration level θ_i is updated every time, when the stimulus s_i occurs. Consequently, a somatic marker represents knowledge about the usefulness of executing a specific action in a specific situation while a frustration level represents the agent's success across all actions.

$$\begin{aligned} \theta_i^{t+1} &= \tanh(\overline{r_{i,j}^t}) \\ &= \tanh(w \cdot \widetilde{r_{i,j}^t} + \hat{w} \cdot \overline{r_{i,j}^{t-1}}) \\ &= \tanh(w \cdot \widetilde{r_{i,j}^t} + \hat{w} \cdot \tanh^{-1}(\theta_i^t)) \end{aligned} \tag{2.16}$$

The semantic meaning of a frustration level's value is shown in figure 2.9. A negative reward decreases the corresponding frustration level which means a higher frustration. Due to the resultant lower threshold it is more likely that options with a lower somatic marker are taken into account. A positive reward increases the frustration level which results in less frustration and a higher threshold. Consequently, only options with higher somatic markers are selected.

As the definition of the frustration levels which are used as thresholds is finished, the selection rules are presented next. There are three different cases that have to be considered in order to assure that the output of the emotional selection $A' \neq \emptyset$.

1. Case: The first case (default case) is present if there is at least one somatic marker $\sigma_{i,j}$ which is greater than or equal to the corresponding

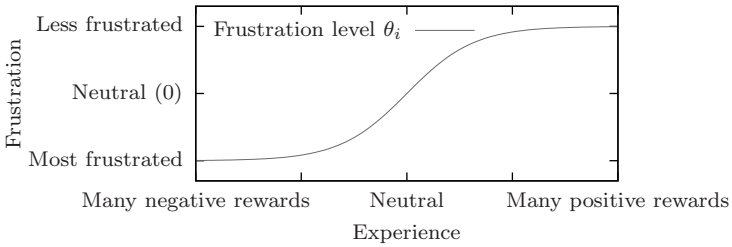

Figure 2.9: Meaning of the frustration level.

t	θ_1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	A'
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t+x$	0.13	-0.39	-0.99	0.13	0.14	$\{a_3, a_4\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 2.5: Exemplary selection of the subset A' for the default case.

frustration level θ_i. In such a case the output of the algorithm is defined as shown in equation (2.17). In consequence, each action with a somatic marker $\sigma_{i,j}$ greater than or equal to the corresponding threshold θ_i is included in the subset A'.

$$A' := \{a_j \in A \,\big|\, \sigma_{i,j} >= \theta_i\} \tag{2.17}$$

An example is shown in table 2.5. At a specific point in time $(t + x)$ the agent recognizes the stimulus s_1. Therefore, the agent starts the emotional decision making algorithm based on the gathered information. Here, the resultant subset A' contains the actions a_3 and a_4, as their corresponding somatic markers fulfil the condition shown in eq. 2.17.

Usually, all somatic markers and frustration levels are initialised with the value zero, which means that the agent does not have any knowledge. Generally, it is assured that $A' \neq \emptyset$, if there is at least one somatic marker which is greater than or equal to θ_i^0 at the initialisation. This is due to the reason that similar computations are used for somatic markers and the corresponding frustration level.

However, there are two additional cases which have to be considered in order to assure that $A' \neq \emptyset$, even if an arbitrary initialisation is used. Furthermore, it is possible that enhancements of the algorithm affect somatic markers as well as the frustration levels in so far that the following cases may occur.

t	θ_1	$\theta_i \cdot \max(M_i)$	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	A'
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t+x$	0.73	$0.73 \cdot 0.53 = 0.3869$	-0.39	0.39	0.53	0.14	$\{a_2, a_3\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 2.6: Exemplary selection of the subset A' for the second case.

2. Case: The second case is present, if the maximum somatic marker $\max(M_i)$ is smaller than the threshold θ_i but both values are greater than or equal to 0 (see eq. (2.18)). A solution for this case has to assure that at least the action corresponding to $\max(M_i)$ is contained in the subset A'. Additionally, actions with somatic marker values close to $\max(M_i)$ should have the possibility of being included in the subset A'.

The answer to the problem is an alternative threshold which is computed by the multiplication of the maximum somatic marker and the current threshold (see eq. (2.19)). Due to the fact that $\max(M_i)$ and θ_i lie within the interval $[0; 1[$, it is guaranteed that the resultant threshold is smaller than the maximum somatic marker. It is also possible that somatic markers which are close to the maximum lie above the alternative threshold, too.

$$\theta_i >= 0 \wedge \max(M_i) >= 0 \wedge \theta_i > \max(M_i) \tag{2.18}$$

$$a_j \in A' \Leftrightarrow \sigma_{i,j} >= \theta_i \cdot \max(M_i) \tag{2.19}$$

How much smaller the alternative threshold $(\theta_i \cdot \max(M_i))$ is compared to θ_i, is dependent on the difference between $\max(M_i)$ and θ_i: the greater the difference the smaller is the resultant alternative threshold. It has to be noticed that the value θ_i is not changed and is used as usual in the next steps, while the alternative threshold is discarded directly after the selection mechanism.

Table 2.6 shows an exemplary case in which $\max(M_i) = \sigma_{1,3} = 0.53$ is smaller than $\theta_1 = 0.73$. Therefore, the resultant subset A' would be empty when the default selection rule shown in equation (2.17) is used. Instead of using $\theta_1 = 0.73$ as threshold, an alternative threshold is computed based on eq. (2.19). Here, this leads to an alternative threshold's value of 0.3869. Consequently, the subset A' contains the actions a_2 and a_3 as their somatic markers' values lie above the alternative threshold.

3. Case: The third case is similar to the second case in so far, that no somatic marker value lies above the threshold. In contrast to the second case,

$\max(M_i)$ is negative which therefore also applies to all of the other somatic markers (see eq. (2.20)). The solution presented for the second case would fail here, as the result of the multiplication shown in eq. (2.19) always lies above the maximum value of a somatic marker. If the current frustration level is negative, the multiplication even increases the threshold instead of decreasing it.

The solution for case 3 has to lead to a consideration of those actions which are the least undesirable of all options. Just as for the second case, an alternative threshold is computed, but here it is based on the quotient of $\max(M_i)$ and $\min(M_i)$ (see eq. (2.21)). If the difference between the maximum and the minimum is small, for example 0, the threshold is -1 and all actions become available. This is the desired output, because all alternatives are rated equally bad. In contrast, a high difference between the maximum and the minimum ensures that only the maximum or values close to it will be taken into account.

$$\max(M_i) < 0 \wedge \theta_i > \max(M_i) \tag{2.20}$$

$$a_j \in A' \Leftrightarrow \sigma_{i,j} >= -(\frac{\max(M_i)}{\min(M_i)}) \tag{2.21}$$

An exemplary case is shown in table 2.7. At $t + x$ all actions are rated equally bad, which leads to an alternative threshold of -1. Consequently, all actions are contained in the subset A'. A further example in which some actions are evaluated as worse than others can be seen at $t + y$. Here, the actions a_3 and a_4 are contained in the subset A', as the values of their somatic markers lie above the alternative threshold -0.121.

Finally, all mechanisms for the emotional decision making part have been described. The algorithm's result is a subset $A' \subseteq A$. Based on this subset it would be possible to perform a further rational analysis according to the SMH. An example for such a rational aspect is given at the end of section 2.3. In this thesis the rational decision making part is not considered any further, instead an action is chosen randomly from the resultant subset.

2.6 Complete Decision Making Algorithm

In this section the single steps of the algorithm are summarized in order to get an overview before the evaluation is presented. Furthermore, a sample calculation is presented to make the single steps more comprehensible.

t	θ_1	$-(\frac{\max(M_i)}{\min(M_i)})$	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	A'
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t+x$	0.12	$-(\frac{-0.39}{-0.39}) = -1$	-0.39	-0.39	-0.39	-0.39	$\{a_1, a_2, a_3, a_4\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t+y$	0.12	$-(\frac{-0.05}{-0.41}) = -0.121$	-0.41	-0.73	-0.05	-0.12	$\{a_3, a_4\}$

Table 2.7: Exemplary selection of the subset A' for the third case.

- **Step 1: A stimulus s_i is recognized.** The agent has a set S which contains all stimuli that are recognizable (see section 2.4). When a specific stimulus $s_i \in S$ is recognized the decision making process starts in order to react to the present situation.

- **Step 2: The subset A' is defined by the emotional selection.** Based on the equations (2.17), (2.19) and (2.21), which are described in section 2.5, those actions which promise to lead to a positive outcome are selected.

- **Step 3: The subsequent rational selection is performed in order to make a final decision.** As described in the previous section 2.4, no further analysis takes place in this thesis. Instead, an action a_j is chosen randomly and the execution of the action is started.

- **Step 4: A reward $r_{i,j}$ is obtained.** After or while executing an action, the agent is able to receive a reward out of the reward set R_{s_i}.

- **Step 5: Update of the related somatic marker $\sigma_{i,j}$.** When a reward is received, the agent updates its emotional memory. At first, the incoming reward is scaled with eq. (2.12) before being used for the update with the equations (2.13) and (2.14).

- **Step 6: Update of the related frustration level θ_i.** The frustration level is also updated after the reception of a reward. Just as in the update of a somatic marker, the reward is also scaled and limited (see equations (2.12) and (2.13)). Afterwards, the update is computed with eq. (2.16). The order of step 5 and step 6 is exchangeable, as they do not effect each other.

- **Step 7: Update of the reliability κ_i.** Based on the obtained reward, the reliability of collected knowledge increases or decreases (see

		Update reliability at last			**Update reliability before step 5 and 6**		
t	$r^t_{i,j}$	θ_1	$\sigma_{1,1}$	κ_1	θ_1	$\sigma_{1,1}$	κ_1
$t+x$	$\min(R_{s_i})$	0.99627	0.99627	10	0.99627	0.99627	10
$t+x+1$	$\min(R_{s_i})$	0	0	0	-0.99627	-0.99627	0

Table 2.8: Exemplary case when the reliability is updated before the somatic marker and before the threshold instead of afterwards.

equations (2.4) and (2.5)). In this step, the reward is not scaled. The reliability influences the computation of the weights for new and collected knowledge for the next iteration (see equations (2.9) and (2.10)).

It can be discussed, if the update of the reliability should be computed before steps 5 and 6 in order to completely use the available information for updating $\sigma_{i,j}$ and θ_i. However, an update of the reliability previous to steps 5 and 6 leads to the values of $\sigma_{i,j}$ and θ_i being changed drastically in some situations. Table 2.8 shows an exemplary case. In this example the action a_1 is rated as most promising at $t+x$ because the somatic marker $\sigma_{1,1}$ has reached the maximum possible value $\tanh(\pi) \approx 0.99627$. In order to clarify the consequences of the computations' order, it is assumed that at $t+x$ the agent obtains the negative reward $\min(R_{s_i})$. Further it applies that $|\min(R_{s_i})| = r_{max_i} = \max\left\{|r| \,\middle|\, r \in R_{s_i}\right\}$.

A closer look at table 2.8 reveals that all values become 0 when the update of the reliability is performed as the final step. This means, that the agent has a neutral rating for a_1 and that the frustration level θ_1 is neutral as well. The reliability value κ_i decreases to 0, which means that collected knowledge is unreliable and therefore not considered at the next computation. If the update is performed before steps 5 and 6, $\sigma_{1,1}$ and θ_1 take the lowest possible value $\tanh(-\pi) \approx -0.99627$. As in the other case, collected knowledge is rated as unreliable. Due to the fact that the frustration level reaches the lowest possible value, all actions are considered in the next iteration. This could be disadvantageous, as also actions with negative somatic markers are taken into account from then on, which is not the case when the reliability is updated as the final step.

In order to make the single steps more comprehensible, an exemplary case based on following configuration is used. This configuration represents the IGT that is explained in section 2.2.

- $S = \{s_1\}$
- $A = \{a_1, a_2, a_3, a_4\}$

t	θ_1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	A'	a_j	$r_{i,j}^{t}$	$r_{i,j}^{t-1}$	κ_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
44	0.13	-0.39	-0.99	0.13	0.14	$\{a_3, a_4\}$	a_4	-200	50	0
45	-0.49	-0.39	-0.99	0.13	-0.49	$\{a_1, a_3, a_4\}$	a_4	50	-200	-2

Table 2.9: Exemplary computations of the decision making algorithm.

- $R_{s_1} = \{-1150, -200, -150, 0, 50, 100\}$
- Reward resolution $c = 10$

Table 2.9 shows the values of the example at specific points in time. The stimulus s_1 is recognized at $t = 44$. Subsequently, the set A' is defined based on the current threshold $\theta_1 = 0.13$ and the somatic markers (see eq. (2.22)).

$$A' := \{a_j \in A \,\big|\, \sigma_{1,j} >= 0.13\} = \{a_3, a_4\} \tag{2.22}$$

The action a_4 is chosen randomly from the subset. Afterwards, the agent obtains a reward $r_{1,4}^{44} = -200$ which is used in order to update the somatic marker, frustration level and reliability. At first the weights of new (w) and collected ($\widehat{w}$) knowledge are computed with the equations (2.23) and (2.24). The weights are necessary for the update of the somatic marker and the threshold.

$$\begin{aligned} w = f_{new}(\kappa_i, c) &= \frac{(\kappa_i)^4}{(c)^4} - \frac{(\kappa_i)^2}{(c)^2} + 1 \\ &= \frac{0^4}{10^4} - \frac{0^2}{10^2} + 1 \\ &= 1.0 \end{aligned} \tag{2.23}$$

$$\begin{aligned} \widehat{w} = f_{old}(\kappa_i, c) &= -\frac{(\kappa_i)^4}{(c)^4} + 2 \cdot (\frac{(\kappa_i)^2}{(c)^2}) \\ &= -\frac{0^4}{10^4} + 2 \cdot (\frac{0^2}{10^2}) \\ &= 0.0 \end{aligned} \tag{2.24}$$

Subsequent to the computations of the weights, the somatic marker is updated. As a start the obtained reward is scaled as shown in eq. (2.25). It is not necessary to limit the input for the hyperbolic tangent with eq. (2.26) as $\overline{r_{1,4}^{44}}$ lies in the interval $[-\pi; \pi]$. Finally, the update process is shown in eq. (2.27).

$$\begin{aligned}\widetilde{r_{1,4}^{44}} &= \frac{r_{1,4}^{44}}{\max\left\{|r| \;\middle|\; r \in R_{s_1}\right\}} \cdot \pi \\ &= \frac{-200}{1150} \cdot \pi \\ &= -0.54 \end{aligned} \tag{2.25}$$

$$\overline{r_{1,4}^{44}} = \begin{cases} \pi & \text{, if } (1.0 \cdot -0.54) + (0.0 \cdot 0.14) > \pi \\ -\pi & \text{, if } (1.0 \cdot -0.54) + (0.0 \cdot 0.14) < -\pi \\ (1.0 \cdot -0.54) + (0.0 \cdot 0.14) & \text{, else} \end{cases} \tag{2.26}$$

$$\begin{aligned}\sigma_{1,4}^{45} &= \tanh(\overline{r_{1,4}^{44}}) \\ &= \tanh(w \cdot \widetilde{r_{1,4}^{44}} + \hat{w} \cdot \overline{r_{1,4}^{43}}) \\ &= \tanh(w \cdot \widetilde{r_{1,4}^{44}} + \hat{w} \cdot \tanh^{-1}(\sigma_{1,4}^{44})) \\ &= \tanh(1.0 \cdot -0.54 + 0.0 \cdot \tanh^{-1}(0.14)) \\ &= -0.49 \end{aligned} \tag{2.27}$$

The next step is the update of the frustration level which is similar to the update of the somatic marker. The only value that differs is $\overline{r_{1,4}^{43}}$ (see eq. (2.28)). Due to the fact that collected knowledge is unreliable and therefore is weighted with 0, the frustration level and the somatic marker share the same value in this example.

$$\begin{aligned}\theta_1^{45} &= \tanh(\overline{r_{1,4}^{44}}) \\ &= \tanh(w \cdot \widetilde{r_{1,4}^{44}} + \hat{w} \cdot \overline{r_{1,4}^{43}}) \\ &= \tanh(w \cdot \widetilde{r_{1,4}^{44}} + \hat{w} \cdot \tanh^{-1}(\theta_1^{44})) \\ &= \tanh(1.0 \cdot -0.54 + 0.0 \cdot \tanh^{-1}(0.13)) \\ &= -0.49 \end{aligned} \tag{2.28}$$

Finally, the reliability is updated based on the obtained reward (see equations (2.29) and (2.30)).

$$-2 = \widehat{\kappa_1^{45}} = \begin{cases} 0, & \text{if } -200 = 0 \vee 50 = 0 \\ 0 + \left\lceil \frac{-200}{1150} \cdot 10 \right\rceil, & \text{if } -200 > 0 \\ 0 + \left\lfloor \frac{-200}{1150} \cdot 10 \right\rfloor, & \text{if } -200 < 0 \end{cases} \tag{2.29}$$

$$-2 = \kappa_1^{45} = \begin{cases} -10, & \text{if} \quad -2 < -10 \\ 10, & \text{if} \quad -2 > 10 \\ -2 & \text{else} \end{cases} \tag{2.30}$$

The previously shown example has given an overview of all the computations that are performed for one single iteration. Furthermore, the meaning of the computations' order has been discussed. Table 2.9 shows that the reward of -200 at $t = 44$ led to the threshold decreasing from 0.13 to -0.49. Additionally, the somatic marker $\sigma_{1,4}$ decreased from 0.14 to -0.49. Consequently, the selected subset A' included the actions a_1, a_3 and a_4 at $t = 45$. A reward resolution $c = 10$ has been used for the example. The evaluation of the decision making algorithm following in section 2.7, among other things, focuses on the impact of the parameter c, as this is the only user-given parameter. For evaluation purposes the IGT is used (see section 2.2).

2.7 Evaluation of the Decision Making Algorithm

In the following the IGT is used for the evaluation of the decision making algorithm. Beside the impact of the chosen value for the reward resolution c, also the similarities and differences between the results of human subjects and the agent are of great interest.

The results presented in Bechara et al. (1994) show that healthy subjects, without any damage at the ventromedial prefrontal cortex are able to identify the advantageous decks. Therefore, they chose cards from the advantageous decks more often than the subjects with respective brain-damages from the control group. Needless to say that, regarding the decision making algorithm for a robot companion, one major evaluation criteria is, that the agent has to be able to identify the advantageous decks as well.

In order to create the same conditions for each participant, Damasio used the same placements of positive and negative rewards for each subject. However, to ensure that the agent is able to solve the IGT independently of the rewards' placements, the rewards within 10 cards were placed randomly for each run in the presented experiments. The configuration criteria of the different decks shown in section 2.2 remain unchanged (e.g. in deck B one card within 10 cards leads to a penalty of -1250). For the algorithm the IGT is described as follows:

- $S = \{\underbrace{triggerCard}_{s_1}\}$

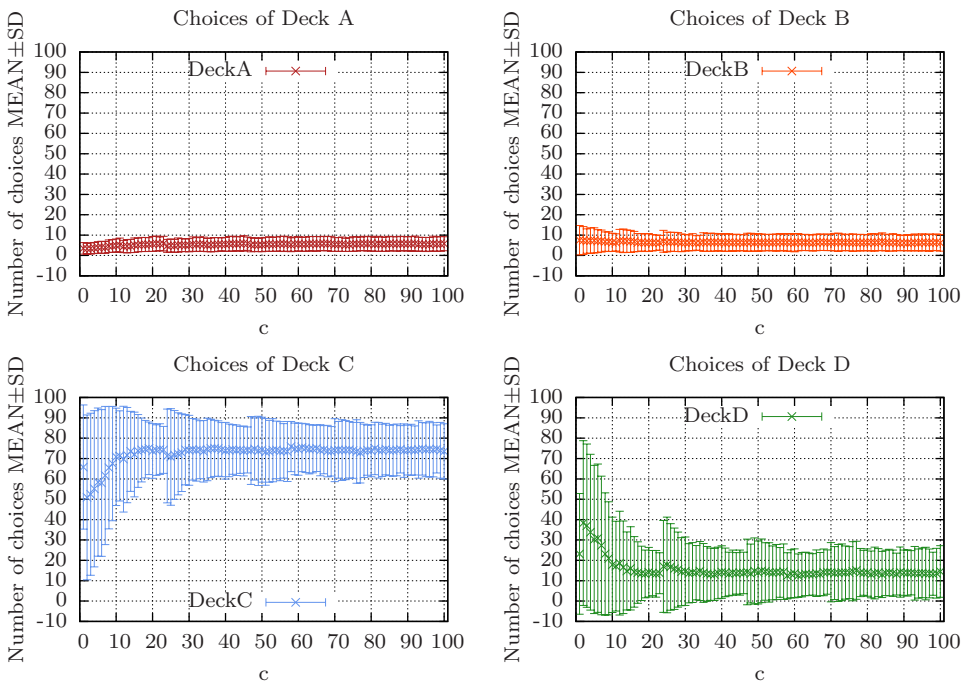

Figure 2.10: Results of the IGT which show the choices (MEAN±SD) for the different decks.

- $A = \{\underbrace{deckA}_{a_1}, \underbrace{deckB}_{a_2}, \underbrace{deckC}_{a_3}, \underbrace{deckD}_{a_4}\}$
- $R_{s_1} = \{-1150, -200, -150, 0, 50, 100\}$

As the reward resolution c is the only user-given parameter, the IGT was performed with different values for c in order to evaluate the impact of this parameter on the results. For the reward resolution all integer values from the interval $[1, 100]$ were used. 1000 runs per value were performed, each run thereby consisting of 100 turns. At the beginning of a run, the agent tried every action in random order once, before the decisions were made based on the decision making algorithm. All values were reset after each run. Additionally, the rewards' placements were created randomly for the next run.

The results are shown in figure 2.10. Regardless of the value given to parameter c, a significant trend of opting for the advantageous decks C and D instead of the disadvantageous decks A and B is observable. Especially deck C was chosen exceptionally often, which is due to the reason that the corresponding somatic marker cannot reach a negative value. This results from the fact, that for a chosen card from deck C, the agent either can obtain

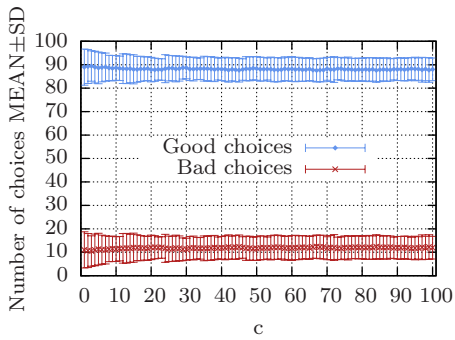

Figure 2.11: Results of the IGT when solely the distinction between good and bad choices is considered.

a positive reward of +50 or a penalty that leads to a reward of 0 (see section 2.2 table 2.1).

A lower value for c leads to higher standard deviations for the advantageous decks C and D. As both decks represent advantageous options, the higher standard deviations are insignificant. The standard deviations for the disadvantageous decks A and B are low compared to those from the advantageous decks and do not differ significantly independent of the reward resolution c. In conclusion, the identification of the advantageous decks is very reliable.

Figure 2.11 shows the overall results as well as figure 2.10 but aggregates the bad choices (deck A and deck B) and the good choices (deck C and deck D). This representation shows more clearly that on average, ≈ 90 choices were made on advantageous decks and that only ≈ 10 choices were made on disadvantageous decks. Furthermore, it can be observed that the parameter c does not have a major impact on the overall results for this task but influences the average choices of deck C and D. Additionally, the parameter c has an influence on single decisions which cannot be reflected by these figures.

In order to show the agent's ability to adapt its decision making behaviour, figure 2.12 shows the results of different phases in the game. For that purpose the game is divided into quarters. Considering the choices of the first quarter (choices 1-25) exclusively, it is observable that the average numbers of choices from the disadvantageous and the advantageous decks are close together. Furthermore, the standard deviations are comparatively high in contrast to the other quarters. In the second quarter (choices 26-50) the average values for the disadvantageous decks are remarkably lower than the average values for the advantageous decks. However, the standard deviations show that even in the second quarter disadvantageous options are taken into account

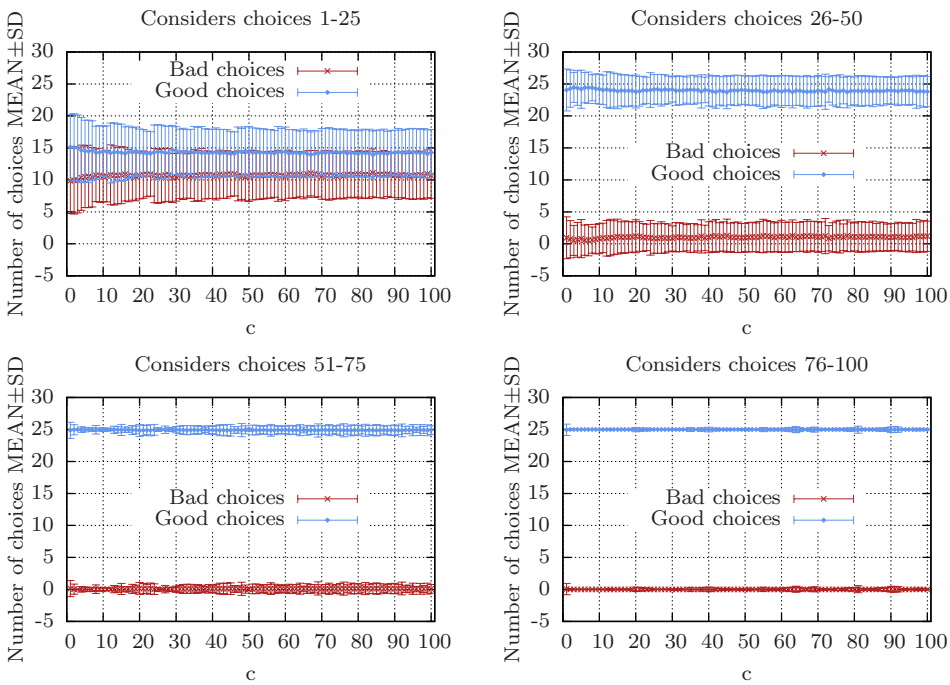

Figure 2.12: Results of the IGT at different quarters of the game.

from time to time. In the quarters 3 (choices 51-75) and 4 (choices 76 -100) almost every decision is made on an advantageous deck. The average value for advantageous choices is ≈ 25 and is ≈ 0 for disadvantageous choices with insignificant standard deviations.

Figure 2.13 shows three exemplary runs with 100 turns each. The first graph shows the decisions of a typical human subject (Bechara et al. (1994)), while the second and the third graph show exemplary runs of the artificial agent with different values for the parameter c.

Damasio has observed during his experiments, that healthy human subjects tend to choose from the disadvantageous decks at the early stages of the IGT before they change their behaviour. In addition to that the results reveal that in total the advantageous choices dominate for healthy human subjects. In the same way the results of the modelled agent show a tendency to opt for disadvantageous decks in early phases, together with an overall preference for advantageous choices especially in later phases of the game.

An obvious difference between the results of the agent and human subjects is the behaviour in later stages of the IGT: while human subjects still chose from the disadvantageous decks sometimes, the modelled agent did not. This can be explained by the fact that the algorithm does not model further

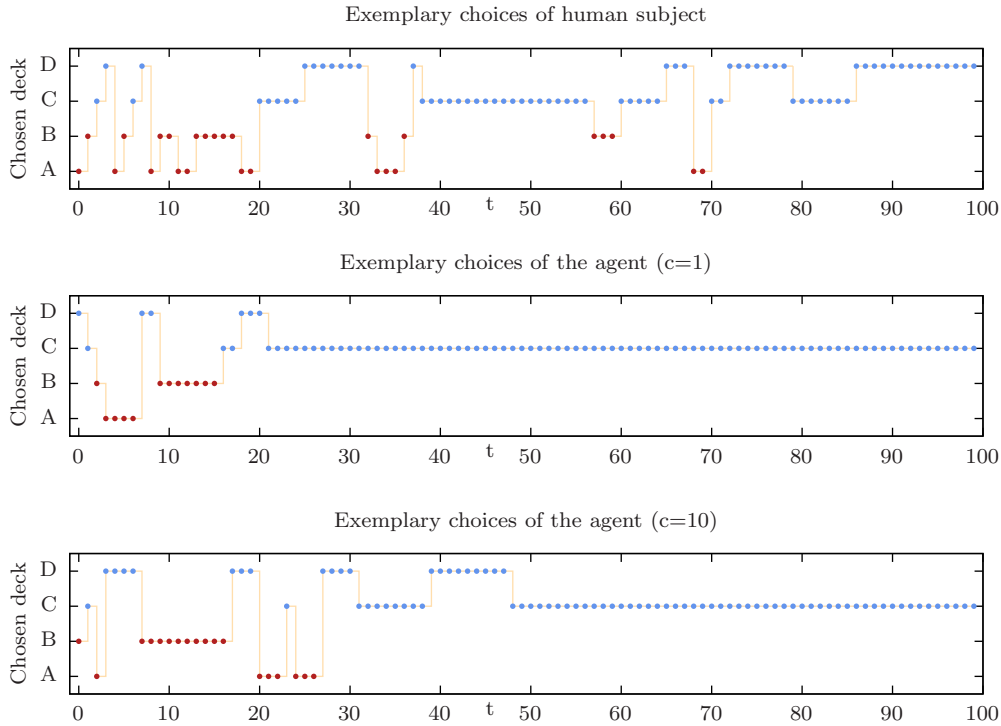

Figure 2.13: Exemplary runs of the IGT.

mechanisms which are part of the human decision making process such as personal characteristics (e.g. curiosity or risk taking). As the results of Dautenhahn et al. (2005) show that people do prefer a robot companion which is controllable (71%) and predictable (90%), it is questionable, if it is worthwhile to implement mechanisms to model personal characteristics. Regarding virtual agents the implementation of personal characteristics or further mechanisms could be desirable in order to reach a more accurate human behaviour (e.g. to simulate evacuation scenarios).

Figure 2.14 and figure 2.15 show the development of the somatic markers and the thresholds for two different reward resolutions ($c = 1$ and $c = 10$) thus giving insight into the way that the agent adapts its behaviour based on the somatic markers. The decisions corresponding to the values shown in the figures 2.14 and 2.15 are illustrated in figure 2.13.

At first, the exemplary case with $c = 1$ is examined. After the agent had tried all actions at the beginning, deck A was preferred for a short time. At $t = 5$ and $t = 6$ the agent obtained two consecutive negative rewards which led to a strong decrease of the somatic marker $\sigma_{1,1}$ and the corresponding frustration level θ_1.

At $t = 7$ the condition $\sigma_{i,j} >= \theta_i$ was fulfilled for every action a_j and therefore the subset contained every action ($A' = A$). The agent chose deck D at $t = 7$ and obtained a positive reward which led to a slight increase of the somatic marker $\sigma_{1,4}$ and the corresponding frustration level θ_1. At $t = 8$ the selected subset only contained the actions a_2, a_3 and a_4 as the somatic marker's value of a_1 lay below the threshold.

From $t = 9$ on the agent chose deck B until at $t = 15$ the agent obtained a high punishment of -1150 which decreased $\sigma_{1,2}$ and θ_1 strongly. Due to the low frustration level all options were available for the next decision. The agent chose deck C at $t = 16$. After that decision only the values of the somatic markers corresponding to deck C and D lay above the frustration level, which means that only advantageous options were taken into account.

From then on the subset did not contain any disadvantageous option until the end of the game, although there were still punishments within the advantageous decks (e.g. see $t = 20$). However, the negative rewards were not sufficient to decrease θ_1 so much, that it became equal to or smaller than a somatic marker of the disadvantageous decks A and B.

After $t = 20$, the agent chose exclusively deck C which led to an increase of $\sigma_{1,3}$ and θ_1 when a positive reward of +50 was obtained. In case of a punishment, which is a reward of 0 for this deck, neither $\sigma_{1,3}$ and θ_1 were changed. This is due to the reason that the chosen reward resolution $c = 1$ only allows two possible weightings: either $1 \cdot r_{i,j}^t + 0 \cdot \overline{r_{i,j}^{t-1}}$ if $\kappa_1 = 0$ or $1 \cdot r_{i,j}^t + 1 \cdot \overline{r_{i,j}^{t-1}}$ if $\kappa_1 = -1$ or $\kappa_1 = +1$. As a reward of 0 does not lead to any change of κ_1, it is very likely that the weighting $1 \cdot r_{i,j}^t + 1 \cdot \overline{r_{i,j}^{t-1}}$ is used very often, when deck C is chosen exclusively.

The exemplary case with $c = 10$ is illustrated in the figures 2.13 and 2.15. The agent tried every action once at the beginning, as in the previous case. Subsequently, deck D was preferred until $t = 7$. Then the agent switched to deck B. In the phases in which deck B is chosen consequently, it can be noticed, that the values $\sigma_{1,2}$ and θ_1 increased more slowly compared to the case in which $c = 1$ was used. Here, the effect of the higher reward resolution allowing for more than two different weightings in contrast to the example with $c = 1$ is observable.

Table 2.10 gives a more detailed view on the values for the exemplary case with $c = 10$. It can be observed that the obtained punishment at $t = 14$ led to the subset A' containing every action in the next step. Deck B was chosen again at $t = 15$ which led to a positive reward (+100). Due to the obtained punishment at $t = 14$ the reliability was decreased to 0 which resulted in an exclusive consideration of new knowledge for the computation at $t = 15$. Therefore, the values $\theta_1^{15} = -0.964$ and $\sigma_{1,2}^{15} = -0.964$ did not have any

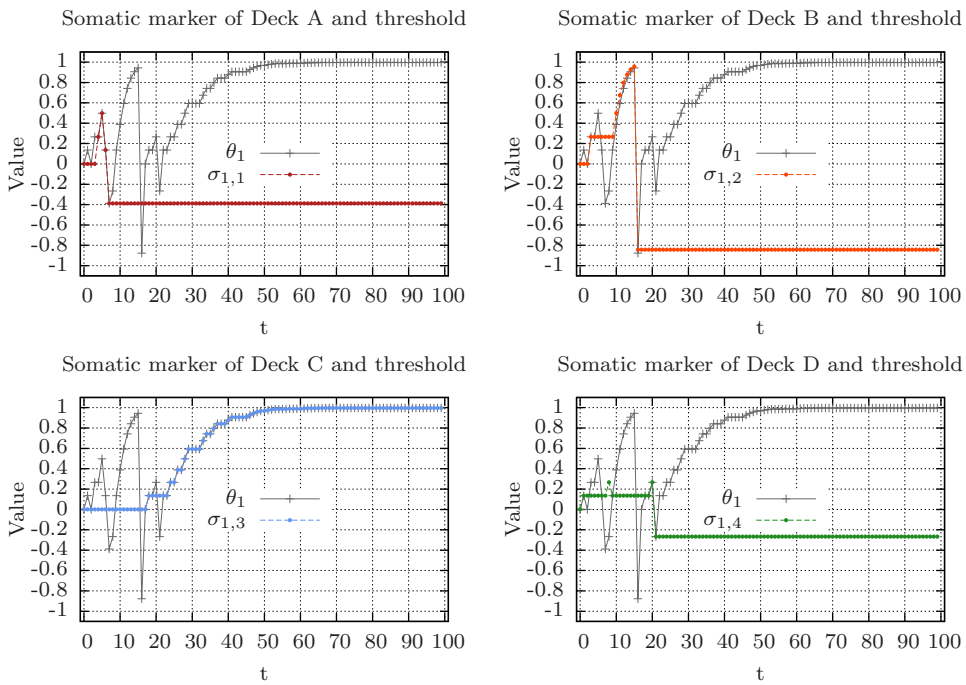

Figure 2.14: Exemplary course of the somatic markers and the threshold (c=1).

t	θ_1	$\sigma_{1,1}$	$\sigma_{1,2}$	$\sigma_{1,3}$	$\sigma_{1,4}$	A'	a_j	$r^t_{i,j}$	$r^{t-1}_{i,j}$	κ_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
13	0.697	−0.388	0.700	0.0	0.148	$\{a_2\}$	a_2	100	100	10
14	0.812	−0.388	0.814	0.0	0.148	$\{a_2\}$	a_2	−1150	100	10
15	−0.964	−0.388	−0.964	0.0	0.148	$\{a_1, a_2, a_3, a_4\}$	a_2	100	−1150	0
16	0.266	−0.388	0.266	0.0	0.148	$\{a_2\}$	a_2	−1150	100	1
17	−0.995	−0.388	−0.995	0.0	0.148	$\{a_1, a_2, a_3, a_4\}$	a_4	50	−1150	−9
18	−0.993	−0.388	−0.995	0.0	0.254	$\{a_1, a_3, a_4\}$	a_4	50	50	−8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2.10: Selected points in time from one run of the IGT. All values are rounded on three decimal places.

impact on the computation for the values θ_1^{16} and $\sigma_{1,2}^{16}$. At $t = 16$ deck B was chosen the last time as the punishment led to a reorientation, although it was possible to choose deck B again at $t = 17$. From $t = 27$ on, the agent chose exclusively from the advantageous decks C and D, whereupon the most decisions were made on deck C.

As mentioned before, deck C is the most chosen deck in the overall results as well as in the presented exemplary cases. In order to investigate if this effect

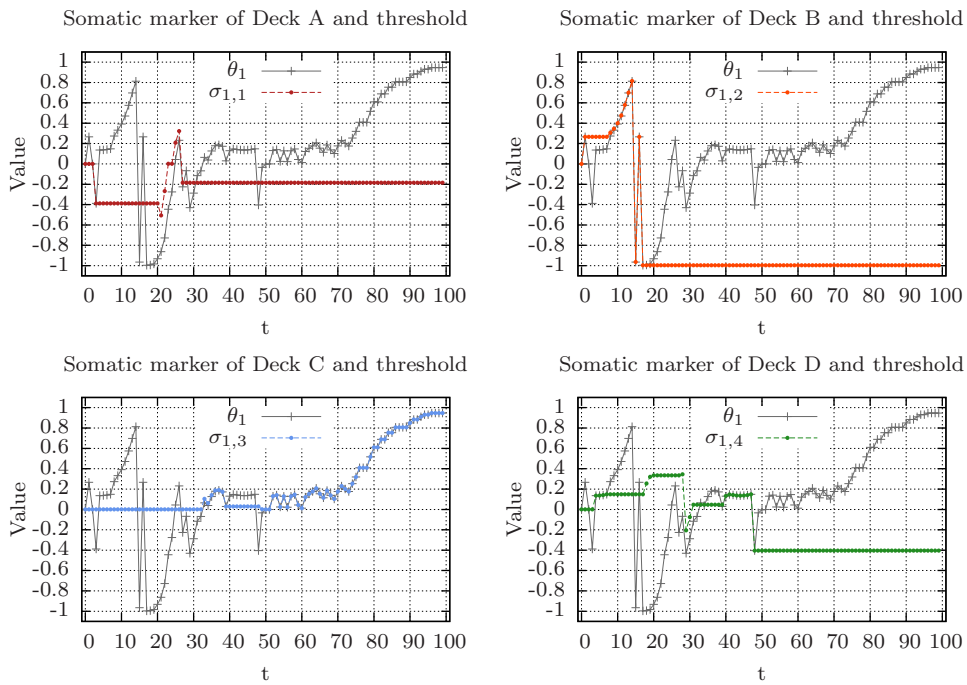

Figure 2.15: Exemplary course of the somatic markers and the threshold (c=10).

results from the used deck configurations, further experiments were performed. For these experiments the configurations of the advantageous decks as well as of the disadvantageous decks were equal. Firstly, a configuration with more frequent penalties but lower magnitudes was used which means that both advantageous decks were configured like deck C and that the disadvantageous decks were configured like deck A. The following listing shows the mentioned deck configuration:

- **Deck A**: Every card gives a benefit of **$100** and **five** out of ten cards additionally have a penalty of **-$250**.
- **Deck B**: Every card gives a benefit of **$100** and **five** out of ten cards additionally have a penalty of **-$250**.
- **Deck C**: Every card gives a benefit of **$50** and **five** out of ten cards additionally have a penalty of **-$50**.
- **Deck D**: Every card gives a benefit of **$50** and **five** out of ten cards additionally have a penalty of **-$50**.

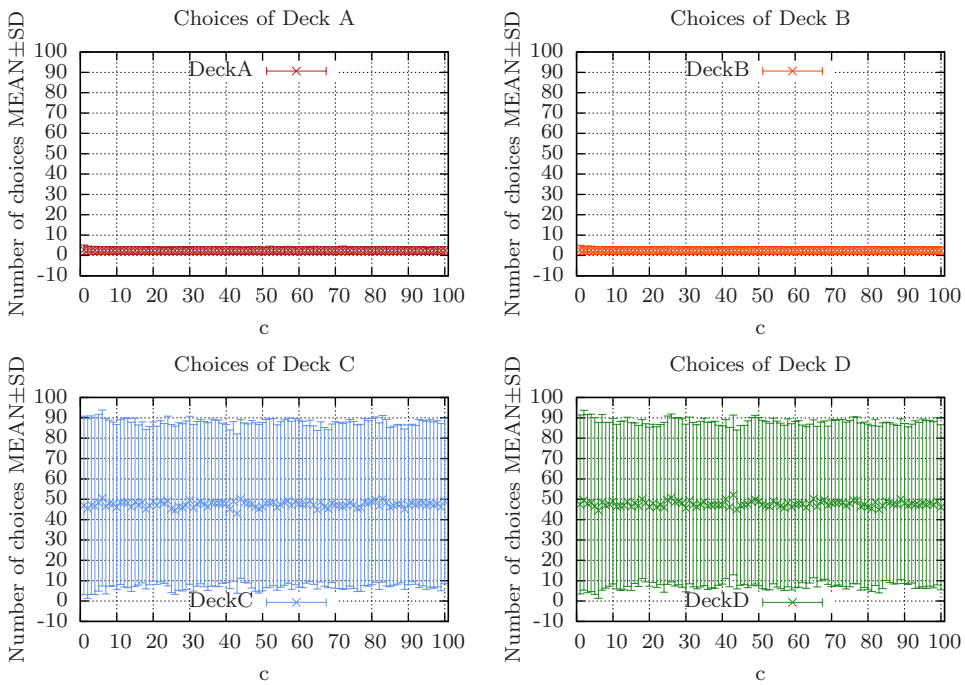

Figure 2.16: Results of the IGT when more frequent negative rewards are used.

The results for this configuration are shown in figure 2.16. As expected, the number of choices is very balanced within the advantageous decks as well as within the disadvantageous decks. The standard deviations are still low for disadvantageous decks, which shows that the agent still is able to reliably identify the advantageous decks. For the second experiment a configuration with less frequent penalties but with higher magnitudes was used. The following listing shows the deck configuration with less frequent penalties:

- **Deck A**: Every card gives a benefit of **$100** and **one** out of ten cards additionally has a penalty of **-$1250**.
- **Deck B**: Every card gives a benefit of **$100** and **one** out of ten cards additionally has a penalty of **-$1250**.
- **Deck C**: Every card gives a benefit of **$50** and **one** out of ten cards additionally has a penalty of **-$250**.
- **Deck D**: Every card gives a benefit of **$50** and **one** out of ten cards additionally has a penalty of **-$250**.

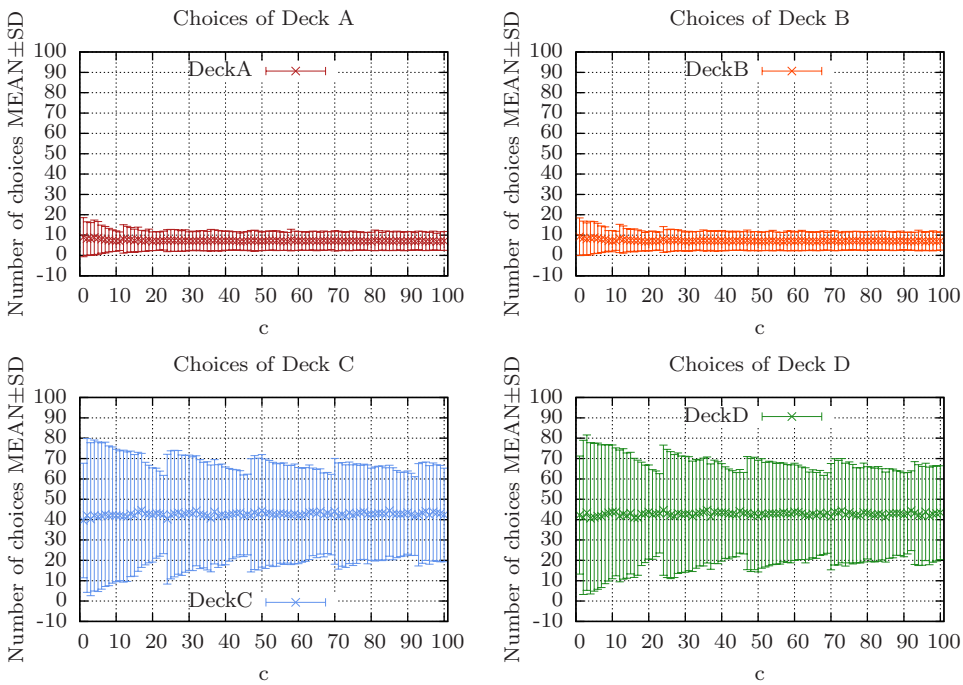

Figure 2.17: Results of the IGT when less frequent negative rewards are used.

Figure 2.17 illustrates the results for the second modification. Just as for the first modification, the number of choices is very balanced within the advantageous and the disadvantageous decks. In contrast to the results shown in 2.16, higher standard deviations for the disadvantageous decks can be noticed. This results from the fact that it is less likely to obtain a punishment when using the deck configuration with less frequent punishments. Consequently, the agent may prefer a disadvantageous deck for a longer period of time, especially at the beginning of the game. It is observable that in both scenarios no single deck is preferred, which was the case in the original configuration, but that the number of advantageous choices still dominates.

In figure 2.17 a conspicuous pattern can be observed for the choices of the decks C and D. From $c = 1$ to $c = 23$ a downward trend of the standard deviations is observable until the standard deviation increases considerably after $c = 24$ has been used. From then on, a similar pattern is repeated until $c = 47$ and so on. A similar pattern is also visible in figure 2.10.

It is assumed that these effects result from the used rewards. With a specific value for c, a used reward is categorized into the next equivalent class. A closer look at how a reward of 50 changes the reliability gives more information. Exemplary computations for the reliability $\widehat{\kappa_i^t}$ are shown in

the equations (2.31) and (2.32). It can be observed that the reliability is increased by one when $c = 23$ is used and by two when $c = 24$ is used. This has an influence on the computations for the weightings of new and collected knowledge. However, as deck C and D are advantageous decks this effect does not contest the overall performance of the algorithm.

$$\widehat{\kappa_i^{t+1}} = \kappa_i^t + \left\lceil \frac{50}{1150} \cdot 23 \right\rceil = \kappa_i^t + 1 \tag{2.31}$$

$$\widehat{\kappa_i^{t+1}} = \kappa_i^t + \left\lceil \frac{50}{1150} \cdot 24 \right\rceil = \kappa_i^t + 2 \tag{2.32}$$

For the sake of completeness, the following IGT configuration was tested in order to support the assumption that the pattern is dependent on the used rewards.

- **Deck A**: Every card gives a benefit of **$75** and **five** out of ten cards additionally have a penalty of **-$200**.
- **Deck B**: Every card gives a benefit of **$150** and **one** out of ten cards additionally has a penalty of **-$1850**.
- **Deck C**: Every card gives a benefit of **$100** and **five** out of ten cards additionally have a penalty of **-$130**.
- **Deck D**: Every card gives a benefit of **$50** and **one** out of ten cards additionally has a penalty of **-$150**.

Using this configuration, deck A and B are disadvantageous decks, as ten consecutive drawn cards of one deck lead to a net loss of -350. Deck C and D are advantageous decks and lead to a net gain of 350 when ten cards are drawn consecutively from one deck. The configuration is comparable to the one presented in section 2.2 but uses other magnitudes.

The results are shown in figure 2.18. It can be observed that the agent is still able to identify the advantageous decks. Again, a pattern is observable but it is shifted compared to the patterns visible in figures 2.10 and 2.17. The first abrupt increase of the standard deviation shows after the use of $c = 35$ leading to a reward of 50 changing the reliability by 2 (see eq. (2.33) and (2.34)).

$$\widehat{\kappa_i^t} = \kappa_i^t + \left\lceil \frac{50}{1850 - 150} \cdot 34 \right\rceil = \kappa_i^t + 1 \tag{2.33}$$

$$\widehat{\kappa_i^t} = \kappa_i^t + \left\lceil \frac{50}{1850 - 150} \cdot 35 \right\rceil = \kappa_i^t + 2 \tag{2.34}$$

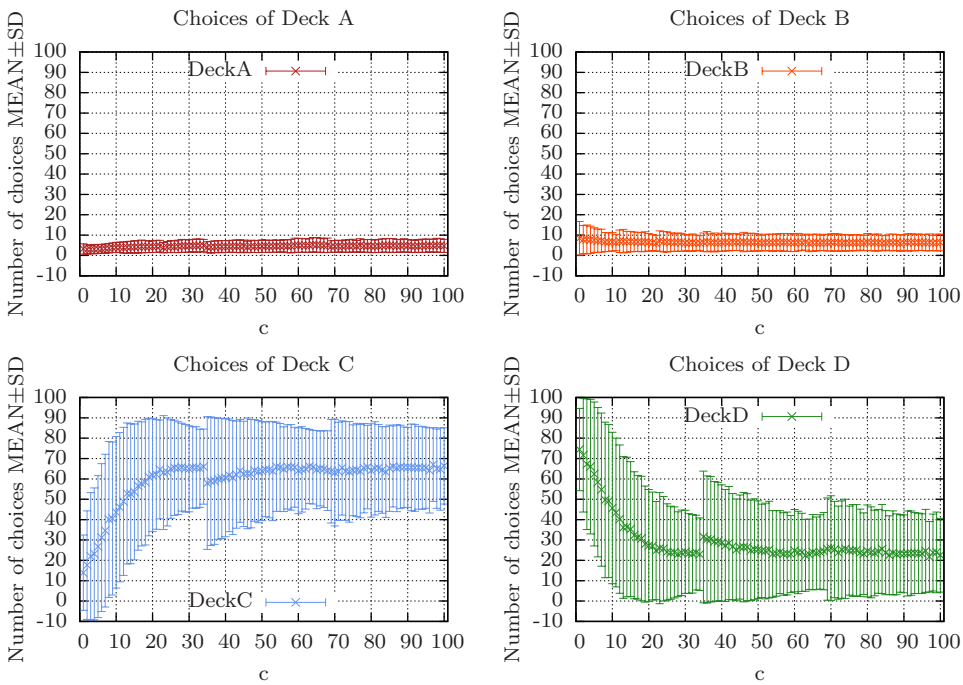

Figure 2.18: Results of the IGT with an alternative configuration that uses different rewards' magnitudes.

Finally, the decision making algorithm has been evaluated by using the IGT with different deck configurations. Most importantly, it can be noticed that the agent is able to identify the advantageous decks in all cases and that the parameter c hardly affects the overall results.

Similar to human subjects, the agent shows a preference for disadvantageous decks in early phases and an overall preference for advantageous decks when considering the whole game. A difference is observable in later phases, in which human subjects tend to still make a few decisions for disadvantageous decks, while the agent remains exclusively at the advantageous decks. As already mentioned, it is questionable if a robot companion would profit from a mechanism leading to few disadvantageous decisions in later phases.

Interim Conclusion: In this chapter a learning and decision making algorithm, influenced by Damasio's SMH, has been presented. For creating the algorithm a variety of design decisions were made. Therefore, some key aspects are summarized in the following. Based on the IGT which was used by Damasio to support his hypothesis, four criteria seeming to be important for the human emotional decision making part were extracted (see section

2.4). For solving the IGT human subjects use new knowledge (current reward) but also collected knowledge (previous rewards). This aspect is considered in the algorithm as the computations of the somatic markers as well as of the frustration levels are based on the current reward but also on rewards gathered previously. Furthermore, the magnitudes of rewards in conjunction with the frequency of their occurrence play an important role within the IGT. This is taken into account especially through the modification of the exponential smoothing function. Additionally, the weightings for the inclusion of new knowledge and collected knowledge are adapted automatically based on the reliability of collected knowledge. It can be concluded that more reliable information leads to collected knowledge becoming more important for the decision making process. Another and more obvious case is that there is no need to consider any collected knowledge when it is unreliable or still not existing. Studies made with human subjects show evidence that somatic markers are also responsible for reversal learning. Due to the chosen weightings within the algorithm, the agent is able to relearn at any time because new knowledge is always considered even when reliable information is available. Furthermore, the computation of the reliability is dependent on the reward's magnitude. This leads to rewards with high magnitudes being able to e.g. decrease the reliability to 0 even when the maximum was reached. Consequently, collected knowledge can be discarded very quickly, which also assists relearning.

Of course the developed algorithm represents only one of many possibilities of implementing the SMH for artificial intelligent systems. The focus in this thesis lies on creating an algorithm providing all benefits of the emotional human decision making process. As discussed in section 2.3, there are some drawbacks in using high dimensional emotional models for customizable robot companions. Therefore, it was not the goal to give the agent artificial feelings such as fear but to create an algorithm generating decisions which are comparable to those made by the human emotional decision making process. The quantitative evaluation presented in this section showed similarities between the decision making behaviour of human subjects and the agent when playing the IGT, especially at the beginning of the game in which the subjects had to learn before they were able to rely on their emotions in order to identify the advantageous decks. Regarding later stages of the game the evaluation revealed differences between the human decision making behaviour and the algorithm's decision making behaviour. Here, human subjects made few decisions from disadvantageous decks while the agent remained choosing advantageous decks. As discussed previously in this section, it is possible to modify the algorithm letting the agent explore other options with a certain probability to reach a more accurate human-like behaviour. However, studies

revealed that such an exploration might not be advantageous for robot companion systems.

For now, the evaluation allowing a comparison between the results of humans and the agent was performed exclusively based on the overall results of the IGT. The question as to whether the agent draws its conclusions in a similar way as human beings requires a further evaluation including a more detailed consideration of single runs. In order to investigate whether the algorithm can produce human-like decisions, especially in early phases of the game, a study was conducted which is presented in the following chapter. This study has focused on the question as to how human subjects perceive the agent's decisions.

Chapter 3

Human Perception of the Agent's Decisions

The overall results of the previous evaluation presented in section 2.7 show that the agent is able to make decisions comparable to those made by human subjects when solving the IGT. One major difference is observable regarding later stages of the IGT in which human subjects made single choices from disadvantageous decks while the modelled agent remained choosing from advantageous decks.

Based on this evaluation a subsequent study was conducted in order to reveal if the agent's decisions are also perceived as human-like by human subjects. As the previously presented evaluation has shown acceptable differences in the decision making behaviour of human subjects on the one hand and the modelled agent on the other hand, the following study focuses exclusively on decisions made at the early stages of the IGT. The study presented in this chapter, which is comparable to the tests presented by Turing (1950), aimed at testing if human subjects are able to distinguish the decisions made by human players from those made by the agent. The results presented in this chapter have also been published in Hoefinghoff et al. (2013a). More information about used terms and statistical methods is presented in the appendix (A).

3.1 Method and Measurements

In the following the experimental setup is described. An overview is shown in figure 3.1.

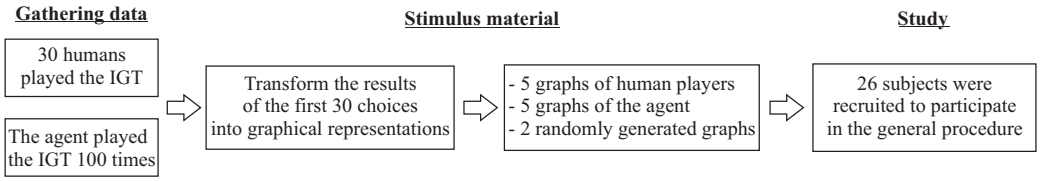

Figure 3.1: Overview of the experimental setup. Gathering data for the stimulus material took place previous to the study. Consequently, none of the 26 subjects was involved in this process.

Participants: In total 26 participants (15 male; 11 female) between 20 and 49 years (MEAN = 25.5; SD = 5.16) took part in this study. 64,4% of them were students and 34,6% were employees. Five of the 26 participants were additionally recruited to support the results with think aloud[1] protocols. The IGT was unknown to most of the participants (18). In the evaluation the influence of this variable was controlled with the result that it had not affected the variables of interest.

Stimulus Material: As the focus of this study was gathering information about the decision making behaviour, the decisions made by the modelled agent as well as those made by human players were transformed into graphical outputs to ensure that the kind of robot does not have any influence on the results. Overall, data from 30 human players[2] and 100 from the artificial agent were taken, of which the first 30 choices were transferred into graphical representations as shown in figure 3.2.

Based on these materials, 10 graphical outputs were randomly selected, 5 of human players and 5 of the artificial agent. In addition, two randomly generated outputs were added. This allowed for determining if subjects are able to distinguish between differences in output files at all. In total, the 12 graphical outputs shown in figure 3.3 were presented to all of the 26 subjects in random order.

Questionnaire: Since the main goal of the study was to test if the artificial somatic marker framework leads to comprehensible decisions from the perspective of human users, different output graphics of random, human and artificial players were presented to the participants. In order to test whether they perceived differences between the outputs of the named sources, participants were asked to rate 8 self-generated bipolar item-pairs on a 7-point

[1]Subjects had to speak out their thoughts during the whole procedure.
[2]None of the 30 players participated in this study.

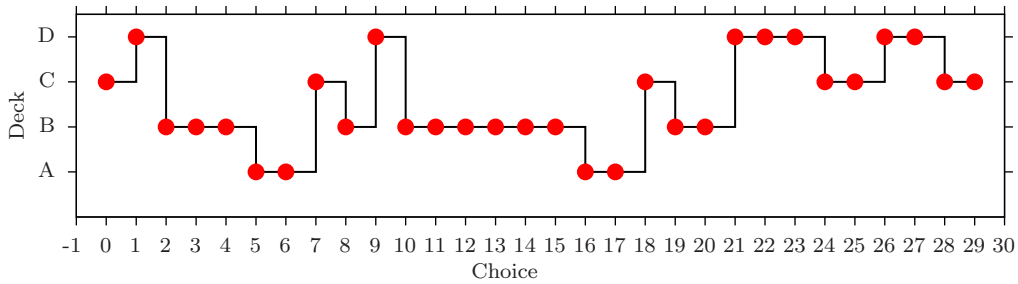

Figure 3.2: Exemplary output which was presented to the participants.

	Items
1	comprehensible - incomprehensible
2	predictable - unpredictable
3	random - deliberate
4	familiar - unfamiliar
5	complex - simple
6	human - machine-like
7	artificial - natural
8	programmed - spontaneous

Table 3.1: Item-pairs which were used to measure predictability and naturalness (7-point semantic differential).

semantic differential (e.g. 1=complex; 7=simple) for each output graphic. The item-pairs are shown in table 3.1. The aim of these items was to measure to which extent the decisions of a player were perceived as comprehensible and predictable on the one hand, and to which extent the course of the game was rated as natural on the other hand.

Furthermore, the participants were asked about the criteria which they used for their evaluation of the output files. Here, participants should indicate whether they used one or more of 6 given criteria shown in table 3.2. Furthermore, free space was left where participants could type in other (missing) criteria. Additionally, age, gender, education, and prior experience with the IGT were collected as moderating aspects that might have affected the evaluation.

3.2 General Procedure

Initially, each participant was instructed to solve the IGT at a computer. In contrast to the study presented in Bechara et al. (1994), a version of the IGT that included randomly generated decks was used since it has been criticised in Fellows and Farah (2005) and Kovalchik and Allman (2006) that

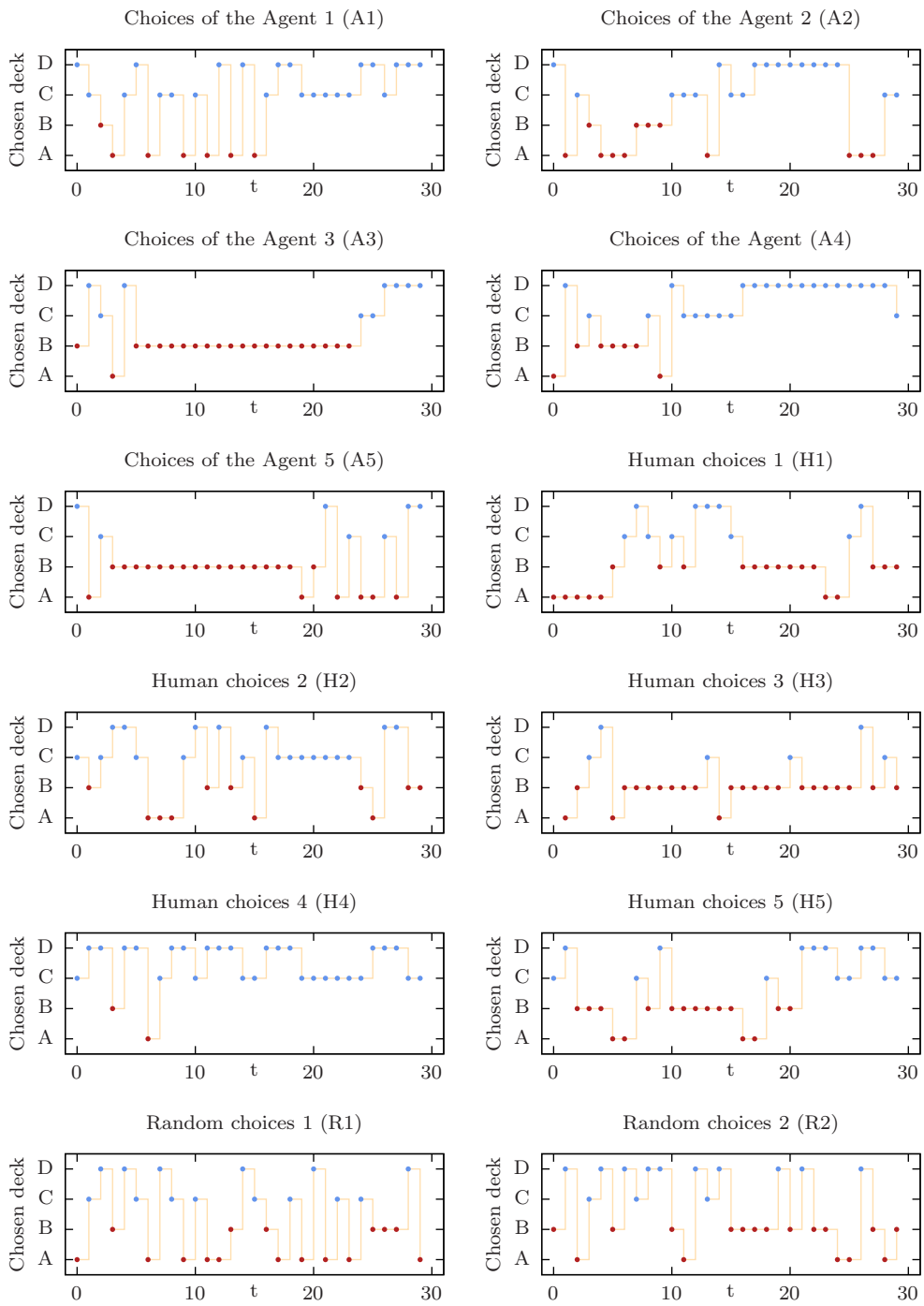

Figure 3.3: Stimulus material which was used: 5 human results, 5 agent's results and 2 random results.

	Evaluation criteria
1	the frequency of changing decks
2	similarity to own procedure
3	procedure at the beginning
4	procedure at the end
5	comprehensibility
6	tendency to choose good decks

Table 3.2: Evaluation criteria which were used for the categorization.

the placement of the punishments has an influence on the results. Needless to say, that each deck was generated according to the rules described in section 2.2.

When the participant had finished the task the experimenter explained the configuration of the decks (e.g. A and B are disadvantageous decks as they lead to a loss of money when drawing 10 cards from one deck). Although the IGT was explained to the participants, it could be expected that they still would have difficulties with evaluating the graphical outputs of other players. Therefore, an output file which visualized the participant's own 30 decisions during the IGT was generated instantly by the experimenter. Then the participants had to explain their choices based on the output file.

For the main part of the study an online survey was displayed on the screen which included the graphical outputs (in random order) as well as items to evaluate the stimulus material. Participants were told that the graphics stemmed from either a human player or a computer. Each page of the survey showed one graphical output file followed by the evaluation items (see table 3.1). For each graphical output participants were instructed to try to reconstruct the course of the IGT while observing it. Afterwards, they were told to rate the course of the game according to the bipolar items listed in table 3.1 and indicate whether they believed that the player was human or artificial. After having passed this procedure twelve times, the participants were asked on which criteria they had based their decisions (see table 3.2). Ultimately, moderating aspects were collected before participants were fully debriefed and thanked for participation.

3.3 Results and Discussion

Since it is assumed that the think aloud procedure might have an influence on the results, the analysis initially was performed based on the sample without the additional 5 participants (N=21). However, in order to determine whether the results are different when the complete sample is used, all analyses were repeated using N=26.

	Factor	
	predictability	naturalness
comprehensible - incomprehensible	.773	
predictable - unpredictable	.732	
random - deliberate	-.674	
familiar - unfamiliar	.658	
complex - simple	-.658	
human - machine-like		.882
artificial - natural		-.836
programmed - spontaneous		-.728
Cronbach's alpha	.749	.773
explained variance (%)	33,51	28,89

Table 3.3: Factor analysis for the evaluation of the output graphics.

Evaluation of the Course of the Game: For the purpose of comparing the different types of outputs (agent, human, random), the dependent variables were summarized into one variable for each type of output (i.e. the evaluations of agent1 - agent5 were summarized in the variable "evaluation of agent output").

The bipolar item-pairs for the evaluation of the output graphics were reduced via factor analysis. Two factors could be extracted which were labelled predictability (5 items, Cronbach's α= .749) and naturalness (3 items, Cronbach's α= .773) according to their constituting items (see table 3.3). These factors were used for further analysis.

To test whether the output files of the artificial agent could be distinguished from those produced by humans or random assignment, repeated-measures ANOVAs were conducted for predictability and naturalness as dependent variables and type of the output (agent, human or random) as within-subject variable, contrasting each type with the artificial agent.

The analysis yielded a significant main effect for predictability ($F(2; 40)=55.05$; $p < .001$; $\eta^2=.734$). According to inner subject contrasts, the evaluation of the artificial agent differed significantly ($p < .001$) from randomly assigned outputs, but not ($p > .05$) from human ones. The results are shown in figure 3.4. Outputs from the artificial agent (MEAN=-0.20, SD=0.47) and human outputs (MEAN=-0.13, SD=0.25) were (surprisingly) perceived as less comprehensible and familiar than randomly assigned ones (MEAN=0.96, SD=0.63). It can be concluded that the framework is able to produce decisions that are comparable to human decisions, at least with respect to predictability.

Regarding naturalness no main effect was obtained. Neither did the moderating variables have any significant impact on the results, when they were included as covariates. A repeated analysis with N=26 revealed comparable results without any noticeable difference.

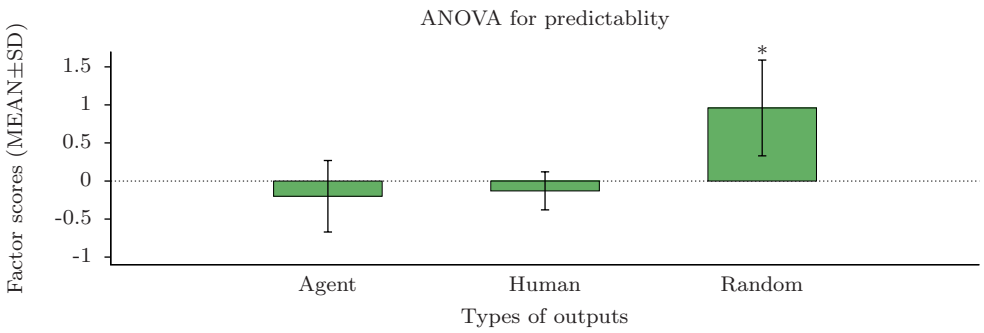

Figure 3.4: Results of the ANOVA for the factor predictability. Random outputs differed significantly from outputs stemming from humans or the agent.

Categorization as Human or Computer: Apart from the evaluation of the factors predictability and naturalness, the participants' choices as to whether a player was a human being or a computer were analysed with regard to the different output types. As depicted in table 3.4, all types of output files were more frequently categorized as stemming from a human player than stemming from a computer. χ^2-tests for each output type revealed that these differences were significant for the outputs of human players (χ^2(1, N=21)=5.95, $p < .05$) and for outputs of the artificial agent (χ^2(1, N=21)=4.20, $p < .05$). For random outputs, the frequency of their categorization as either 'human' or 'computer' did not differ in any significant way, when the data of N=21 were used.

However, a repeated analysis that included the results of all participant (N=26) revealed a different result concerning the categorization of random outputs. For N=26 the randomly generated outputs were categorized as stemming from a human player significantly more often (χ^2(1, N=26)=6.23, $p < .05$). The outputs of the artificial agent (χ^2(1, N=26)=7.87, $p < .01$) and of human players (χ^2(1, N=26)=12.31, $p < .01$) were still categorized as stemming from a human player significantly more often, however with increased levels of significance.

Regardless of the number of the participants considered in the analysis, it can be noted that the outputs of the artificial agent were significantly more often categorized as 'human'. The same applies for the outputs of human players.

The output H4 (see figure 3.3) is the only human output that was more frequently rated as stemming from a computer than as stemming from a human player. As the output H4 shows the result of a very successful human

Results N=26 (N=21)	Categorized as computer	Categorized as human
A1	14 (11)	12 (10)
A2	6 (6)	20 (15)
A3	8 (7)	18 (14)
A4	10 (7)	16 (14)
A5	11 (11)	15 (10)
Total categorizations of agent's outputs	**49 (42)**	**81 (63)**
H1	7 (7)	19 (14)
H2	5 (4)	21 (17)
H3	11 (11)	15 (10)
H4	16 (13)	10 (8)
H5	6 (5)	20 (16)
Total categorizations of human outputs	**45 (40)**	**85 (65)**
R1	10 (9)	16 (12)
R2	7 (6)	19 (15)
Total categorizations of random outputs	**17 (15)**	**35 (27)**
Total	**111 (97)**	**201 (155)**

Table 3.4: Results of the categorization.

player, it can be assumed that the participants were skeptic whether this output could stem from a human player especially when their own result was not as good as the shown output (H4).

Randomly generated outputs were significantly more often categorized as 'human' when N=26 was used for the analysis. In conclusion, this might be an indication of the participants' general tendency to opt for the categorization 'human' regarding all kinds of outputs. Although there is also an observable trend of rating random outputs as 'human' for N=21, the result is not significant ($p > .05$). Therefore, it can also be suggested that the think aloud procedure may have influenced the categorization. Additionally, the participants were forced to make a decision for one of the both categories. The results of the ANOVA show that the perception of the random outputs differed significantly from the human outputs and the agent's outputs at least with respect to predictability. Consequently, this might have led to confusion on the part of the subjects during the categorization which may have resulted in arbitrary categorizations.

However, the participants were not able to distinguish between human outputs and the agent's outputs. This result supports the assumption that the decision making algorithm is able to make human-like decisions at least when solving the IGT (see section 2.7).

Criteria	Frequency of choice N=26 (N=21)
frequency of changing decks	18 (16)
similarity to own procedure	16 (11)
procedure at the beginning	14 (11)
comprehensibility	14 (14)
tendency to choose good decks	14 (10)
procedure at the end	9 (5)

Table 3.5: Frequencies of choosing the given criteria.

Evaluation Criteria: In order to answer the question why the output files were perceived as human-like, or which criteria served as basis for the evaluation, different criteria were also checked within the analyses. The criteria were mainly collected to give further insights into the set of criteria according to which outputs from human and artificial players were distinguished. The results of the main study revealed that no difference between the outputs from the artificial agent and human players were observable. Thus, it was no longer necessary to take a closer look at the criteria in order to analyse which one is the decisive criterion that distinguishes the agent's outputs from human output files. Consequently, the results gained for the criteria are only briefly summarized in the following.

This finding is also reflected in the participants' choices of the given criteria and further mentioned criteria (from the survey and the protocols). Almost all given criteria are chosen equally often, demonstrating that no single one seems to be the one criterion that determines whether the output is perceived as human-like or artificial (see table 3.5). Instead, all criteria seem to be equally important for the evaluation of the output files, and are moreover equally fulfilled by the agent's and the human outputs.

However, the analysis of the think aloud protocols revealed that the criteria given in the questionnaire were relevant to the evaluation of the IGT since the same criteria could be extracted from the protocols. Furthermore, participants reported (in the survey as well as in the think aloud sessions) that they also considered many other criteria for their evaluation like testing of each deck in the beginning, repetition of procedures or how often one deck was chosen consecutively. As the results show, the outputs of the artificial agent as well as the human output files fulfil what the participants perceived as human(-like) decision making.

Interim Conclusion: The results presented in this chapter show that the participants were not able to distinguish the agent's output files from the human ones. In conclusion, the algorithm is able to make decisions which are perceived as human-like.

Certainly it could not be ensured that the complexity of the evaluation task itself might not have influenced the results. Furthermore, the amount of output graphics (12) presented to one participant might have been too high resulting in a fatigue effect. However, the stimulus material was presented in random order to control this bias.

Based on the results presented in this and in the previous chapter it can be noticed that the decision making algorithm fulfils two important criteria to be applied for customizable robot companions. Firstly, the algorithm does not have any sensitive user-given parameters. Secondly, the algorithm is able to solve tasks successfully (e.g. the IGT). Consequently, the algorithm is implemented on a real robot in order to investigate how humans perceive the decision making behaviour in human-robot interaction (HRI) scenarios.

In section 1.1 three different user types namely *Computer Scientist*, *Researcher* and *Normal User* were mentioned. With the purpose of giving all types of users the possibility of enhancing the robot's capabilities, the implementation includes different interfaces which can be used in dependency of the users' expertise. More details on the implementation are presented in the following chapter.

Chapter 4

Implementation on the Humanoid NAO Robot

As mentioned in the introduction, many decision making approaches are not available on a real robot, which leads to their evaluations being exclusively performed in simulations like those described in section 2.7. Examples for such evaluated algorithms can be found in Hoogendoorn et al. (2009); Pimentel and Cravo (2009). If the simulations use a simulated environment and robot, they are often subjects to restrictions regarding the complexity of real world situations. For now, the evaluation of the presented decision making algorithm also uses restrictions, like sequential processes or the usage of placeholders for stimuli or actions. In order to use this algorithm on a real robot companion, the implementation has to deal with different circumstances of the real world like a high degree of parallelism. Due to these circumstances, it is possible that a new stimulus occurs while an action is already performed or that a performed action has to be stopped in order to start a new action. Furthermore, the robot might have to be capable of already receiving a reward while performing an action and not afterwards.

With respect to the goal that the robot companion has to be able to deal with a variant number of tasks, it is of great importance that the implementation offers the possibility of easily extending the robot's recognizable stimuli, performable actions and so on. In the following, details on the implementation of the presented decision making algorithm on the humanoid NAO robot are presented. These include the specification of the robot, the software's implementation architecture and the realization of specific modules. The shown concepts are not limited to an application on the NAO robot but can be transferred to other robotic systems. Some of the parts presented in this chapter have also been published in Hoefinghoff et al. (2012, 2013b).

Figure 4.1: NAO robot of the Aldebaran Robotics Company.

4.1 NAO Robot

The NAO robot is a humanoid robot of the Aldebaran Robotics Company with 25 degrees of freedom (see figure 4.1). It has different sensors to perceive its environment, such as cameras, ultrasonic, microphones and tactile sensors. Additionally, the NAO robot is able to interact with its environment by speech or with its actuators. In addition to a software development kit (SDK), a software called *Choregraphe* is provided which allows users to create behaviour networks with a graphical user interface (GUI).

The main software architecture is based on a broker pattern (see figure 4.2). A broker software called *NAOqi* to which new modules can be registered is running on the robot. Modules which are to be executed directly on the robot have to be written in C++ or Python, while it is also possible to use further programming languages like Java on desktop computers. By working with the *Choregraphe*, users are able to use/create boxes which can access functions of other modules via Python. It is possible to cascade several boxes. A box may consist of the following different parts:

1. **Parameters:** A box can have several parameters the values of which can be accessed in the corresponding Python code.

2. **Inputs:** A box can receive signals through several inputs. Each input is dedicated to a piece of code, which is executed when a signal is received through this input. Furthermore, an input can come directly from a variable out of the globally accessible memory which is called *ALMemory*[1].

[1]The names of modules that are provided by the Aldebaran Robotics Company begin with *AL*.

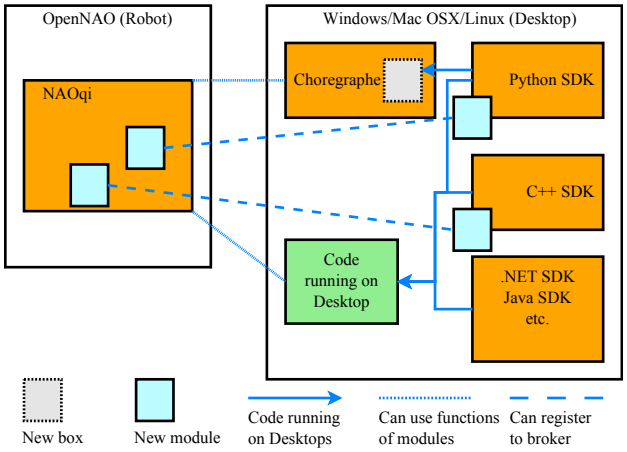

Figure 4.2: The NAO software architecture (version 1.14.5). The main broker *NaoQi* to which new modules can be registered runs on the robot. Furthermore, it is possible to access the modules' functionalities from desktop computers via self-programmed software or the *Choregraphe*.

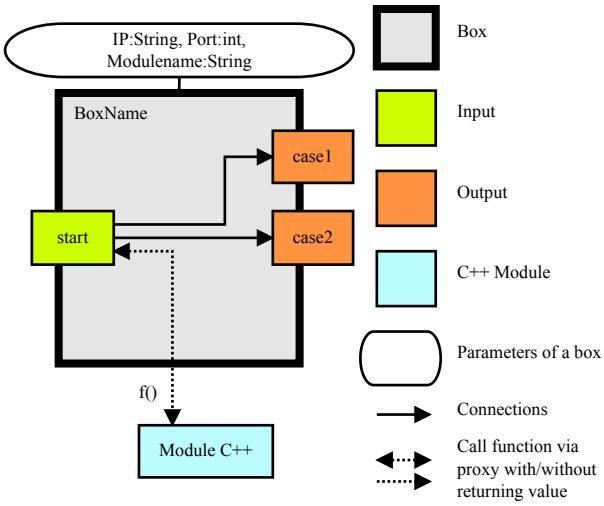

Figure 4.3: Exemplary box which can be created by using the *Choregraphe*. A box can have several parameters, inputs, outputs and is able to access other modules' functions via Python code.

3. **Outputs:** A box can have several outputs which are used to send signals to other boxes.

Figure 4.3 shows an exemplary box with one input (named *start*) and two outputs (named *case1* and *case2*). The box gets the class name of the module

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        self.ProxyOfModule = ALProxy(
        self.getParameter("Modulename"), # Modulename is MyModule
        self.getParameter("IP"),
        self.getParameter("Port"))

    def onLoad(self): # code that is executed when the box is loaded
        pass

    def onUnload(self): # code that is executed when the box is unloaded
        pass

    def onInput_start(self):
        result = self.ProxyOfModule.f()
        if result == 0:
            self.case1()
        else:
            self.case2()
```

Listing 4.1: Exemplary Python code of a box which calls a function f() via a proxy of the module *MyModule* (see listing 4.2).

which should be addressed (here *MyModule*), the IP address of the robot and the port of the *NAOqi* software as parameters. In the corresponding Python code the first step consists of obtaining a proxy object of the desired module (see listing 4.1). Therefore, the parameters *Modulename*, *IP*, and *Port* are used. The code that is executed when a signal is received via the input *start*, is defined through *onInput_start(self)*. In this example the function f(), the return of which is an integer value, of the module *MyModule* is called (see listing 4.2). If the returned value of the function is 0, a signal is sent through the output *case1*, otherwise the output *case2* sends a signal.

Before the implementation architecture of the decision making algorithm is described in the following, the most important terms are summarized.

Choregraphe: High level GUI, provided by the Aldebaran Robotics Company. The software is used to create behaviours for the robot.

Module: Software component which offers a specific functionality (e.g. *MyModule* shown in listing 4.2). All modules developed for this thesis are written in C++ and are executed directly on the robot.

NAOqi: Main broker at which modules can be registered to offer their functionalities.

```
// each module is derived from the class ALModule
MyModule::MyModule(
        boost::shared_ptr<AL::ALBroker> broker,
        const std::string& name): AL::ALModule(broker, name)
{
  // description of the module
  setModuleDescription("This is an exemplary module.");

  // description of the function
  functionName("f", getName(), "An exemplary function");
  // description of the returning value
  setReturn("return", "Returns a random integer value between 0-9");
  // a bound function can be accessed via a proxy
  BIND_METHOD(MyModule::f);
}

// method that can be called via proxy
int MyModule::f()
{
  srand(time(NULL));
  return rand() % 10;
}
```

Listing 4.2: Exemplary C++ code of the module *MyModule*. Its function f() can be accessed via a proxy.

Box: Part of the *Choregraphe* from which functionalities provided by modules can be accessed via Python code (see listing 4.1).

4.2 Implementation Architecture of the Decision Making Algorithm

As the research on the development of robot companion systems continuously reveals new findings, a modular architecture makes the implementation of those as convenient as possible. Based on the work of Vitay and Hamker (2011) an architecture in which different brain regions involved in the human decision making process are modelled is chosen. The main focus lies on a clear separation of the modules' competences, not on the creation of an accurate simulation of brain processes. Furthermore, modules which are not directly inspired by brain regions are added in order to provide additional functions or to support other modules. A short overview on the different modules and their competences is given by the following listing.

- *SensoryCortex:* Is responsible for the stimulus recognition.
- *Amygdala:* Is implemented as a control centre which spreads all incoming information to the dedicated parts.

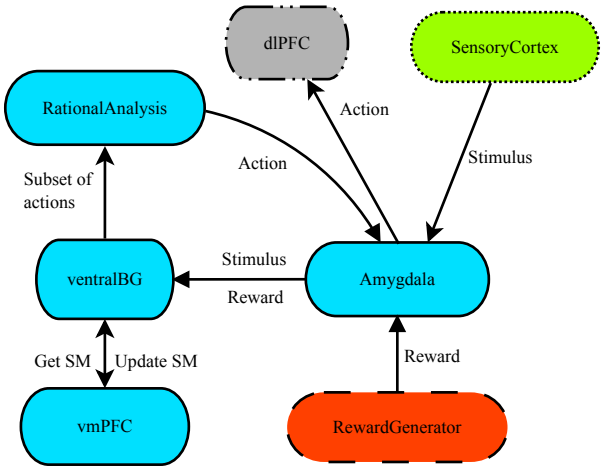

Figure 4.4: Software architecture for the implementation on the NAO robot, which is adapted from Vitay and Hamker (2011). The different parts can be divided into four areas of responsibility: stimulus recognition (green), decision making (blue), execution of actions (grey) and creation of rewards (red).

- *ventral basal ganglia (ventralBG):* Are responsible for the emotional selection based on the somatic markers.
- *ventromedial prefrontal cortex (vmPFC):* Is responsible for the creation and update of the somatic markers and all the other values of the algorithm.
- *RationalAnalysis:* Is responsible for the rational selection.
- *dorsolateral prefrontal cortex (dlPFC):* Is responsible for the execution of actions.
- *RewardGenerator:* Is responsible for the creation of rewards.

Figure 4.4 shows all modules as well as the most important connections and sent values. A complete description of all connections and values is presented in the following section. Every cycle starts with the recognition of a stimulus. When the *SensoryCortex* recognizes a stimulus, it sends it to the *Amygdala* which relays it to the *ventralBG*. Within the *ventralBG* the emotional selection is performed using the information from the *vmPFC*. The resultant subset is sent to the *RationalAnalysis* which chooses one final action from the subset. This action is sent to the *Amygdala* which relays it to the *dlPFC* that initiates the execution of the action. Then the robot is able to obtain a reward via the *RewardGenerator* which sends it to *Amygdala*. From

Name of the variable	Value
LeftBumperPressed	0
LastWordRecognized	"Hello"

Table 4.1: Examples for entries inside the ALMemory.

the *Amygdala* the reward is sent to the *ventralBG* which relay it to the *vmPFC* in order to update the emotional memory. Details on the implementation of each single module as well as on the interactions among the different modules are described in the following sections.

4.3 Stimulus Recognition

In humans the perception of stimuli starts with the sensory cortex (Watson et al., 2010, p. 78-80) which contains different sensory areas like the visual cortex or the auditory cortex. In the same way the NAO robot needs to be able to recognize defined stimuli. As already mentioned in section 2.4, a stimulus can be a single signal or sensory value or even a combination of different inputs which describes a whole situation.

Every sensor of the robot and even most high level modules, like the speech recognition, provide their values through variables into the *ALMemory*. An example can be seen in table 4.1 which shows the variables for the left foot's bumper and the result of the speech recognition. While the variable *LeftBumperPressed* can only take the values 0 (bumper is not pressed) or 1 (bumper is pressed), the variable *LastWordRecognized* contains the last word/sentence that was recognized. Modules are able to subscribe to variables which lead to a callback function being executed when the value of a variable is changed.

In the following, a stimulus is defined within an XML file. Every definition contains different elements. Listing 4.3 shows the definition of an exemplary stimulus named *BumperPressed*. This stimulus occurs when either the right foot's bumper or the left foot's bumper is pressed. The parameter *neededConditions* defines the number of conditions which have to be fulfilled for this stimulus. A change of this parameter from 1 to 2 results in both bumpers having to be pressed at the same time to trigger this stimulus. Via the parameter *timeDifference* it can be defined in which time interval (in seconds) this specific stimulus can occur. This is supposed to give the robot time to react before the same stimulus is recognized again and the decision making process is started once more. Furthermore, the parameters *maxReward* (r_{max_i}) and *rewardResolution* (c) used in the algorithm are defined (see

```
<stimuli>
  <stimulus name="BumperPressed" neededConditions="1" timeDifference="2" maxReward="50"
       rewardResolution="10">
    <condition>
      <currentValue>LeftBumperPressed</currentValue>
      <operator>==</operator>
      <value>1</value>
    </condition>
    <condition>
      <currentValue>RightBumperPressed</currentValue>
      <operator>==</operator>
      <value>1</value>
    </condition>
  </stimulus>
  .
  .
  .
</stimuli>
```

Listing 4.3: Exemplary stimulus named *BumperPressed* which is triggered when either the robot's left foot's bumper or the right foot's bumper is pressed.

section 2.4). Multiple conditions can be defined for one stimulus. The first element *currentValue* defines the variable in the *ALMemory* which is used for comparison (e.g. *LeftBumperPressed*). Furthermore, the relational operator (e.g. ==) is defined as well as the comparative value (e.g. 1).

In order to create a stimulus recognition system which offers an interface that allows an easy definition of new stimuli without programming, the two modules *SensoryCortex* and *CheckSensoryData* are created in C++. The module *SensoryCortex* is responsible for the recognition of the defined stimuli and the forwarding of those to the decision making process, while the module *CheckSensoryData* is a supportive part which starts a separate thread continuously checking the defined conditions for each stimulus. An exemplary case is shown in figure 4.5. Both sensors (left bumper and right bumper) write their values into the *ALMemory*. In the case that one bumper is pressed, the corresponding value (*LeftBumperPressed* or *RightBumperPressed*) becomes 1 which is recognized by a thread of the *CheckSensoryData*. If all conditions for the defined stimulus are fulfilled (see listing 4.3), the thread changes the value of the variable with the name corresponding to the name of the stimulus (here *BumperPressed*). The value type of this variable is a time stamp. This is due to the reason that the conditions of a stimulus may have different types of values (see table 4.1). Subsequently, the *SensoryCortex*, which is subscribed to the variable *BumperPressed*, is notified about the change and writes/updates two variables named *CurrentStimulus* and *CurrentTimeStamp*

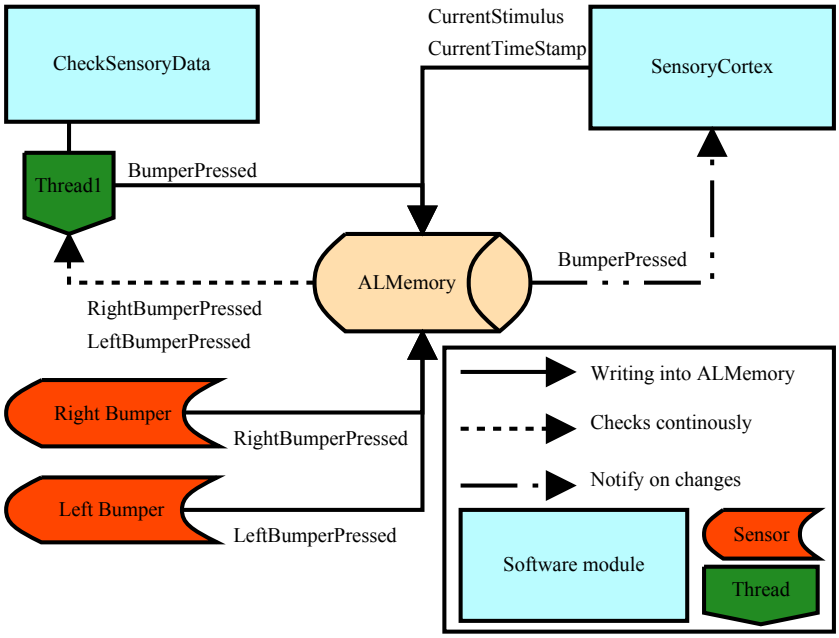

Figure 4.5: Exemplary case that shows the stimulus recognition.

into the *ALMemory*. The value of *CurrentStimulus* is the name of the stimulus (*BumperPressed*), while the value of *CurrentTimeStamp* is the time stamp.

In addition to the C++ modules, boxes corresponding to the modules are created with the *Choregraphe*. Figure 4.6 and 4.7 show schematic representations of those. Both receive their information on the stimuli being recognized from an XML file. When the input *start* is triggered at the *CheckSensoryData_Box*, the module starts a separate thread for each stimulus. All started threads can be ordered to terminate via the input *stop* in order to disable the recognition of stimuli.

When the input *start* is triggered at the *SensoryCortex_Box*, the module subscribes to all stimuli which are defined within the XML file. From this point on the module gets notified about any change of the corresponding variable inside the *ALMemory*. As already explained before, the *SensoryCortex* writes the name of the stimulus into a variable *CurrentStimulus*. In the case that the variable's value is updated, the input *CurrentStimulus*[1] is triggered and the box relays the name of the stimulus to the output *stimulusDetected*. This output is connected to the decision making process. In this implementation the robot always reacts to the stimulus that is recognized at last.

[1]Inputs with names starting with a capital letter are connected directly to variables of the *ALMemory*.

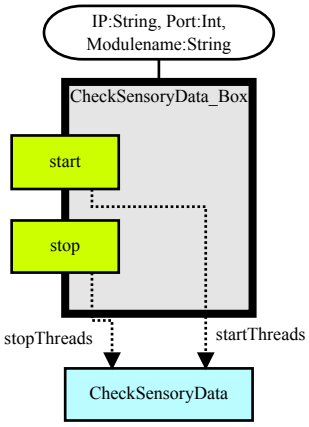

Figure 4.6: Corresponding box of the module *CheckSensoryData* in the *Choregraphe*.

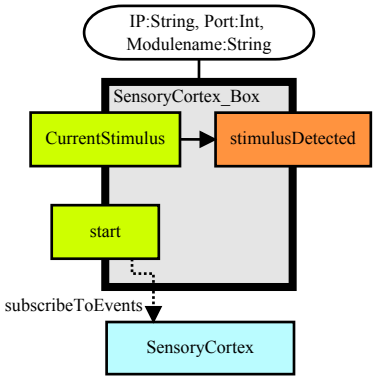

Figure 4.7: Corresponding box of the module *SensoryCortex* in the *Choregraphe*.

Referring to the algorithm, the modules *CheckSensoryData* and *SensoryCortex* perform the first step of the algorithm described in section 2.6. Due to the created interface, it is possible to add new stimuli without writing any source code by just modifying the XML configuration file. Furthermore, software developers are able to easily integrate their modules as long as they provide their results through the *ALMemory*.

4.4 Decision Making

According to Damasio different human brain parts are involved in the emotional human decision making process. The amygdala is mainly responsible for the primary emotions and triggers innate behaviours (Damasio, 1994, p. 131-134), while the ventromedial prefrontal cortex (vmPFC) is responsible for secondary emotions (Damasio, 1994, p. 134-139). Although the algorithm does not consider any innate behaviour, the amygdala is implemented as a control centre which spreads all information to the dedicated modules. If needed, the *Amygdala* can be enhanced in order to add mechanisms to include innate behaviours.

Furthermore, Damasio has described that the vmPFC acts via the amygdala which makes it important even for the processing of secondary emotions. In accordance with Damasio, secondary emotions are used for the creation of somatic markers: "[...]*somatic markers are a special instance of feeling generated from secondary emotions.*"(Damasio, 1994, p. 174). Therefore, the implemented module, which represents the *vmPFC*, is responsible for the

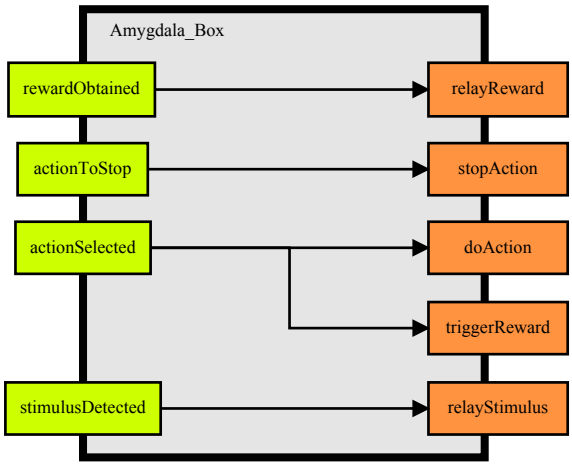

Figure 4.8: Box of the *Amygdala* in the *Choregraphe*.

creation and the update of the somatic markers. In addition, any further values which are part of the algorithm, like the frustration levels, are created and updated via the *vmPFC*. In several works like Doya (2000) or Vitay and Hamker (2011), the ventral basal ganglia (*ventalBG*) are described as playing a decisive role for reinforcement learning and decision making. Hence, the *ventralBG* are implemented to perform the emotional selection by using the information from the *vmPFC*.

Finally, a further module *RationalAnalysis* is created in order to make the final decision out of the subset which is selected by the *ventralBG*. As the rational decision making part of the algorithm, presented in chapter 2.5, just randomly chooses an action from the subset, there is no necessity for implementing this functionality into a separate module but with regard to enhance the rational analysis it is advisable.

To go into more detail, figure 4.8 shows a schematic of the *Amygdala_Box*. As mentioned before, the main function which is assigned to the *Amygdala_Box* is to control the information flow. Due to that reason, no C++ module is created and the box does not need any parameters. Every time the *Amygdala_Box* receives a stimulus via the input *stimulusDetected*, the stimulus is sent to the emotional decision making part via the output *relayStimulus*. Furthermore, the finally chosen action, which is received from the *RationalAnalysis* via the input *actionSelected*, is sent to the *dlPFC* via the output *doAction* in order to start its execution. Parallel to that, the output *triggerReward* becomes active to signalize that a reward can be obtained. In case of a negative reward, the *Amygdala_Box* is able to receive the name of an action that is supposed to be cancelled via the input *actionToStop*.

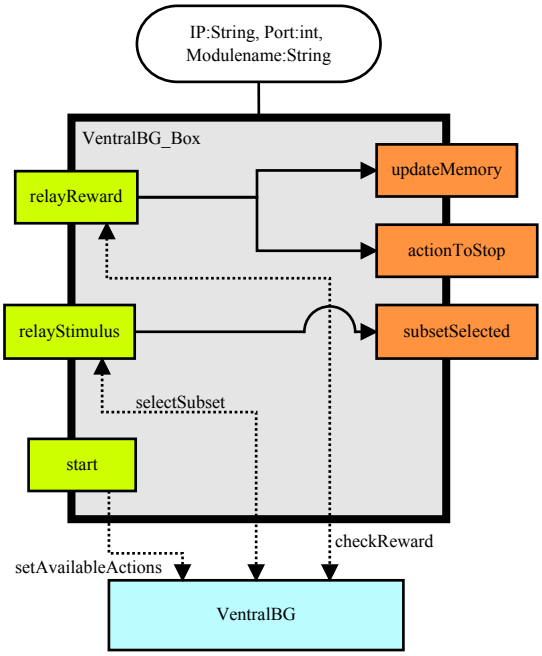

Figure 4.9: Corresponding box of the module *ventralBG* in the *Choregraphe*.

This action is relayed via the output *stopAction* to initiate its cancellation. Furthermore, the rewards are received via the input *rewardObtained* and are relayed via the output *relayReward* to the *ventralBG*.

Figure 4.9 shows a schematic of the *ventralBG* which are responsible for the emotional selection (see step 2 in section 2.6). The selection process is started when a stimulus is received via the input *relayStimulus*. All available actions are set via the input *start* at the initialization. An action is represented through a separate *Choregraphe* network in which a specific behaviour is defined. It is possible that one action consists of parallel behaviours and action sequences. Furthermore, the obtained reward is processed via the *ventralBG*. A reward that is received via the input *relayReward*, is relayed via the output *updateMemory* in order to update the emotional memory. Additionally, a signal to stop the current action is sent via the output *actionToStop* in case that the obtained reward is negative.

Like the stimuli, also the available actions are defined within the XML file. An example is shown in listing 4.4. Each action is represented through a *Choregraphe* network which was transferred to the robot. Figure 4.10 shows an exemplary network for the action *Greeting*. This behaviour is performed when the action *Greeting* is chosen. The exemplary network leads to the robot standing up at first and then saying “hello” while waving simultaneously.

```
<actions>
  <action>Greeting</action>
  <action>SitDown</action>
  <action>Dance</action>
  .
  .
  .
</actions>
```

Listing 4.4: Exemplary actions which the robot is able to execute.

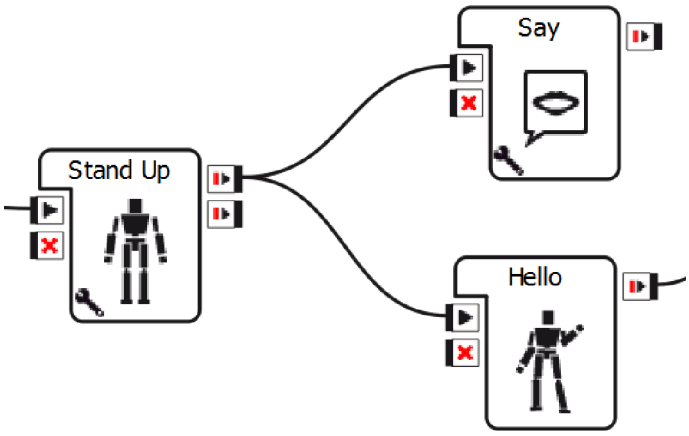

Figure 4.10: Exemplary action (*Greeting*).

Subsequent to the emotional decision making part, the *ventralBG* send the selected subset to the *RationalAnalysis* via the output *subsetSelected* (see figure 4.11). Here, a final action is chosen randomly out of the subset (see step 3 in section 2.6).

When the robot obtains a reward, the *vmPFC* is responsible for updating all values of the algorithm like the somatic markers or the frustration levels (see figure 4.12). Based on the incoming information the steps 5-7 of the algorithm presented in section 2.6 are performed.

4.5 Action Execution

The execution of actions is initiated by the dorsolateral prefrontal cortex (dlPFC). Figure 4.13 shows the corresponding box of the module *dlPFC*. Via the input *doAction* the execution of an incoming action is started. As explained before, it is also possible to stop the execution of an action in the case that a negative reward has been obtained. Therefore, the input *stopAction* is used. Natural environments are subjects to change. Consequently, it is

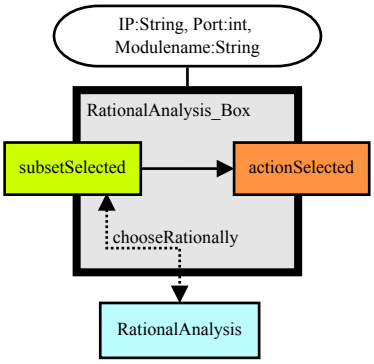

Figure 4.11: Corresponding box of the module *RationalAnalysis* in the *Choregraphe*.

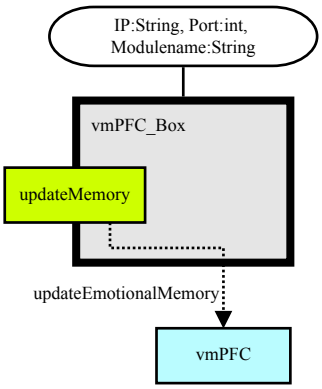

Figure 4.12: Corresponding box of the module *vmPFC* in the *Choregraphe*.

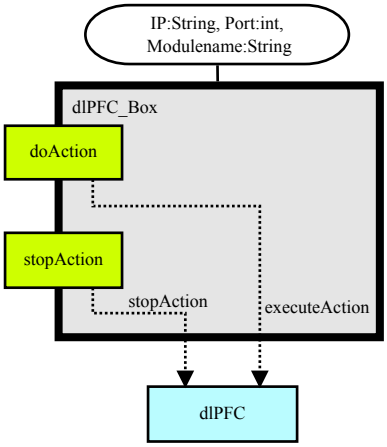

Figure 4.13: Corresponding box of the module *dlPFC* in the *Choregraphe*.

possible that a new situation occurs, while an action is already being executed. Due to this reason, the action execution module has to deal with parallel processes.

To ensure that the robot is able to react to a present stimulus at any time, there are different cases that have to be considered. The *dlPFC* is able to manage all the cases described in the following. Due to the implementation it is ensured that the robot never stops recognizing the environment or making decisions even when the execution of an action is already planned or in progress. An initial position (*initposition*) from which the robot is able to start its actions safely is added to the robot. Figure 4.14 shows the seven

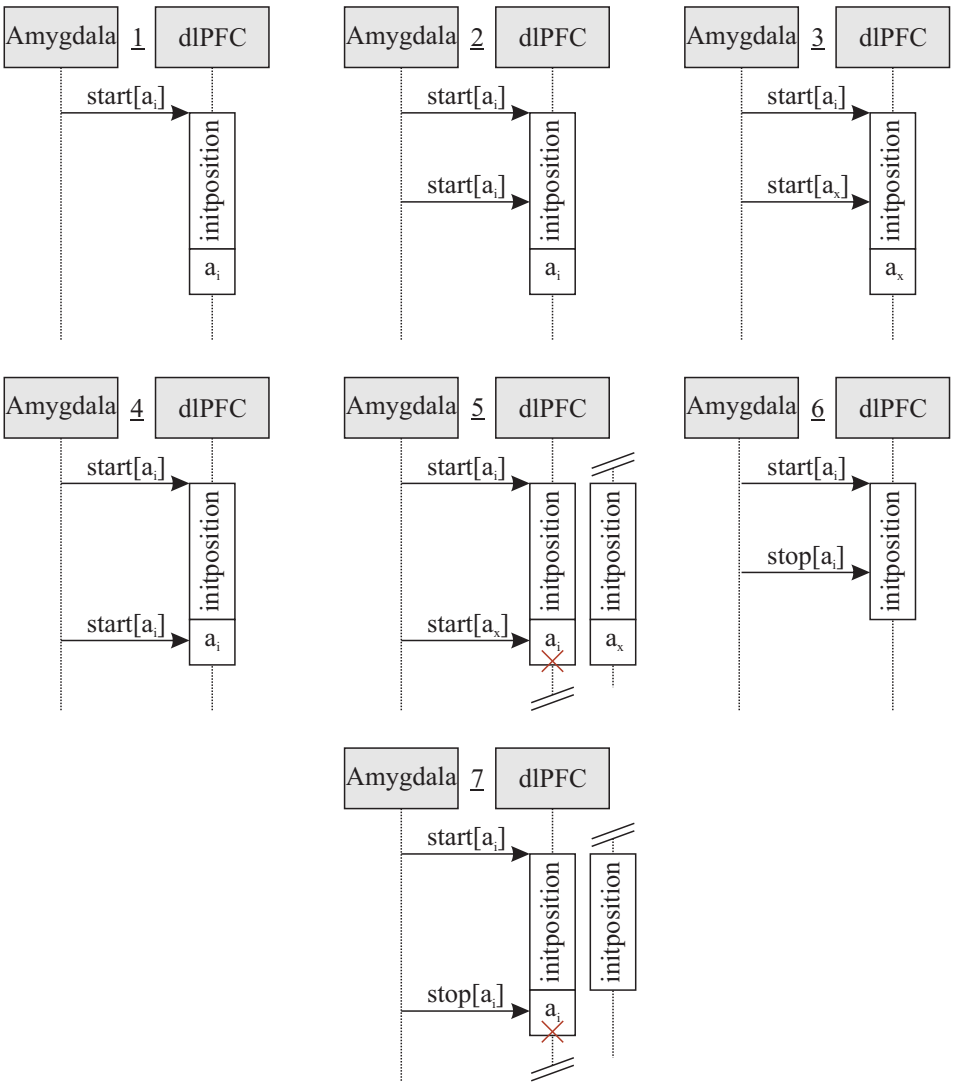

Figure 4.14: The different cases which are distinguished in order to handle the execution and cancellation of actions. The operations *start* and *stop* correspond to the inputs *doAction* and *stopAction* of the *dlPFC_Box*.

different cases which can occur. How these cases are handled is described by the following enumeration:

1. If the robot is not yet busy, it goes into the initial position and starts the planned action a_i afterwards.

2. If the robot decides to start the same planned action a_i while going into the initial position, the robot executes action a_i after having finished going into the initial position.

3. If the robot decides to start a different action a_x while going into the initial position, the robot discards the originally planned action a_i and executes action a_x after having finished going into the initial position.

4. If the robot decides to start the same action a_i while already executing a_i, the robot finishes the execution of a_i and does not start the action again.

5. If the robot decides to start a different action a_x while executing a_i, the robot stops the execution of a_i, goes into the initial position and starts the action a_x afterwards.

6. If the robot decides to stop an action a_i while going into the initial position, the robot only finishes going into the initial position.

7. If the robot decides to stop an action a_i while already executing a_i, the robot stops the execution of a_i and goes into the initial position.

All these cases are handled automatically by the *dlPFC*. Therefore, the user does not have to take care about parallelism when creating an application for the robot. This is indispensable in order to ensure that even non-experts are able to use the framework.

4.6 Reward Generation

As the robot needs be able to adapt its behaviour via users' feedback, a module *RewardGenerator* is created (see figure 4.4). Beside the possibility of affecting the robot's behaviour the way how the feedback can be given to the robot needs to be configurable. Due to that reason, the ways of giving rewards are also configured within the XML file. Listing 4.5 shows an exemplary configuration with which the user is able to give the robot a positive reward (+50) by touching the frontal tactile sensor on the robot's head or to give the robot a negative reward (-50) by saying "Bad".

The corresponding box can be seen in figure 4.15. When the input *start* is triggered, the module starts the initialization using the information within the XML file. Every time a pair of a stimulus and an action is sent to the input *triggerReward* from the *Amygdala*, the module subscribes to the variables defined in the XML file. From this point, the robot is able to

```
<rewards>
  <reward rewardValue="50">
    <triggerName>FrontTactileTouched</triggerName>
    <operator>==</operator>
    <value>1</value>
  </reward>
  <reward rewardValue="-50">
    <triggerName>LastWordRecognized</triggerName>
    <operator>==</operator>
    <value>Bad</value>
  </reward>
  .
  .
  .
</rewards>
```

Listing 4.5: Exemplary configuration which includes two possibilities of obtaining a reward.

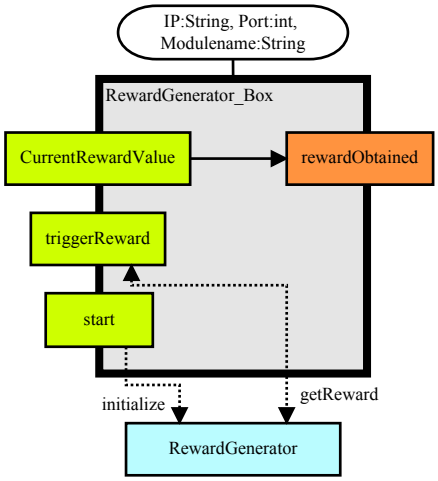

Figure 4.15: Corresponding box of the module *RewardGenerator* in the *Choregraphe*.

receive a reward when one condition, specified within the XML file, is fulfilled. If one condition is fulfilled (e.g. the front tactile sensor is pressed), the module unsubscribes from all variables and writes the reward's value into the variable *CurrentRewardValue* (see step 4 in section 2.6). Subsequent to that, a triple consisting of stimulus, action and reward is sent through the output *rewardObtained*.

4.7 Case Example of one Decision Making Cycle

A case example is presented in this section in order to clarify how the previously described modules and boxes interact with each other. Figure 4.4 shows all connections between the modules/boxes. The following example shows in detail which data is sent and received. For that purpose the notation shown in eq. (4.1) is used. The notation reflects that, if the box *BoxName* receives a signal with some values through the input *InputName*, it sends a signal with some values to the *Receivers* through the output *OutputName*.

$$\begin{aligned} BoxName : I(InputName[Values]) \\ \overset{Receivers}{\rightarrow} O(OutputName[Values]) \end{aligned} \tag{4.1}$$

The information of listings 4.3, 4.4 and 4.5 are used for the example. Given that, the robot is able to recognize one stimulus, is able to execute three actions and can be rewarded in two different ways. This leads to the following formal description:

- $S = \{\underbrace{BumperPressed}_{s_1}\}$
- $A = \{\underbrace{Greeting}_{a_1}, \underbrace{SitDown}_{a_2}, \underbrace{Dance}_{a_3}\}$
- $R_{s_1} = \{-50, +50\}$

The case that the user is pressing one of the robot's bumpers is the starting point. This is recognized by the *SensoryCortex* which sends the stimulus to the *Amygdala* (see eq. (4.2)).

$$\begin{aligned} SensoryCortex : I(CurrentStimulus[s_1]) \\ \overset{Amygdala}{\rightarrow} O(stimulusDetected[s_1]) \end{aligned} \tag{4.2}$$

The *Amygdala* receives the stimulus and relays it to the *ventralBG* in order to initiate the emotional decision making process (see eq. (4.3)).

$$\begin{aligned} Amygdala : I(stimulusDetected[s_1]) \\ \overset{ventralBG}{\rightarrow} O(relayStimulus[s_1]) \end{aligned} \tag{4.3}$$

Based on the incoming stimulus s_1, the *ventralBG* select a subset A' using the decision making algorithm shown in chapter 2.5. Subsequent to the selection, the stimulus and the resultant subset are sent to the *RationalAnalysis* (see eq. (4.4)).

$$\begin{aligned} & ventralBG : I(relayStimulus[s_1]) \\ \overset{RationalAnalysis}{\rightarrow} & O(subsetSelected[s_1, A' = \{a_2, a_3\}]) \end{aligned} \tag{4.4}$$

During the rational analysis an action is randomly chosen from the subset. The stimulus and the chosen action are sent to the *Amygdala* (see eq. (4.5)).

$$\begin{aligned} RationalAnalysis : & I(subsetSelected[s_1, A' = \{a_2, a_3\}]) \\ & \overset{Amygdala}{\rightarrow} O(actionSelected[s_1, a_2]) \end{aligned} \tag{4.5}$$

When the final decision is received at the *Amygdala*, it relays it to the *dlPFC* in order to initiate the execution. Simultaneously, the action and the stimulus are relayed to the *RewardGenerator* to signalize, that from now on, the robot is able to obtain a reward (see equations (4.6) and (4.7)).

$$\begin{aligned} Amygdala : & I(actionSelected[s_1, a_2]) \\ & \overset{dlPFC}{\rightarrow} O(doAction[a_2]) \end{aligned} \tag{4.6}$$

$$\begin{aligned} Amygdala : & I(actionSelected[s_1, a_2]) \\ \overset{RewardGenerator}{\rightarrow} & O(triggerReward[s_1, a_2]) \end{aligned} \tag{4.7}$$

The *dlPFC* receives the chosen action and starts its execution without sending any further signals to other modules (see eq. (4.8)).

$$dlPFC : I(doAction[a_2]) \tag{4.8}$$

As long as the *RewardGenerator* does not receive a new pair of a stimulus and an action, it is possible to give a reward for the currently received pair (see eq. (4.9)). If a reward is given, a triple consisting of stimulus, action and reward is sent to the *Amygdala* (see eq. (4.10)).

$$RewardGenerator : I(triggerReward[s_1, a_2]) \tag{4.9}$$

$$\begin{aligned} RewardGenerator : I(CurrentRewardValue[-50]) \\ \overset{Amygdala}{\rightarrow} O(rewardObtained[s_1, a_2, r_{1,2} = -50]) \end{aligned} \tag{4.10}$$

The *Amygdala* relays the received triple to the *ventralBG* (see eq. (4.11)).

$$\begin{aligned} Amygdala : I(rewardObtained[s_1, a_2, r_{1,2} = -50]) \\ \overset{ventralBG}{\rightarrow} O(relayReward[s_1, a_2, r_{1,2} = -50]) \end{aligned} \tag{4.11}$$

When the *ventralBG* receive the triple, it is checked, if the obtained reward is negative. If a negative reward is obtained, which is the case in this example, a signal with the current action is sent to the *Amygdala* (see eq. (4.12)). Irrespective of whether the obtained reward is positive or negative, the triple is sent to the *vmPFC* in order to update the emotional memory (see eq. 4.13).

$$\begin{aligned} ventralBG : I(relayReward[s_1, a_2, r_{1,2} = -50]) \\ \overset{Amygdala}{\rightarrow} O(actionToStop[a_2]) \end{aligned} \tag{4.12}$$

$$\begin{aligned} ventralBG : I(relayReward[s_1, a_2, r_{1,2} = -50]) \\ \overset{vmPFC}{\rightarrow} O(updateMemory[s_1, a_2, r_{1,2} = -50]) \end{aligned} \tag{4.13}$$

Within the *vmPFC* the emotional memory is updated using the information of the received triple (see eq. 4.14). As the obtained reward is negative, the *Amygdala* relays the current action to the *dlPFC* in order to stop its execution (see eq. 4.15).

$$vmPFC : I(updateMemory[s_1, a_2, r_{1,2} = -50]) \tag{4.14}$$

$$\begin{aligned} Amygdala : I(actionToStop[a_2]) \\ \overset{dlPFC}{\rightarrow} O(stopAction[a_2]) \end{aligned} \tag{4.15}$$

Finally, the *dlPFC* receives the action which has to be stopped (see eq. (4.16)).

$$dlPFC : I(stopAction[a_2]) \tag{4.16}$$

Subsequent to these steps, the robot is able to react to the next stimulus. As already explained, it is also possible that a stimulus is recognized while the robot is executing an action. Due to the mechanisms presented previously, the robot is able to react to the current stimulus in such a case.

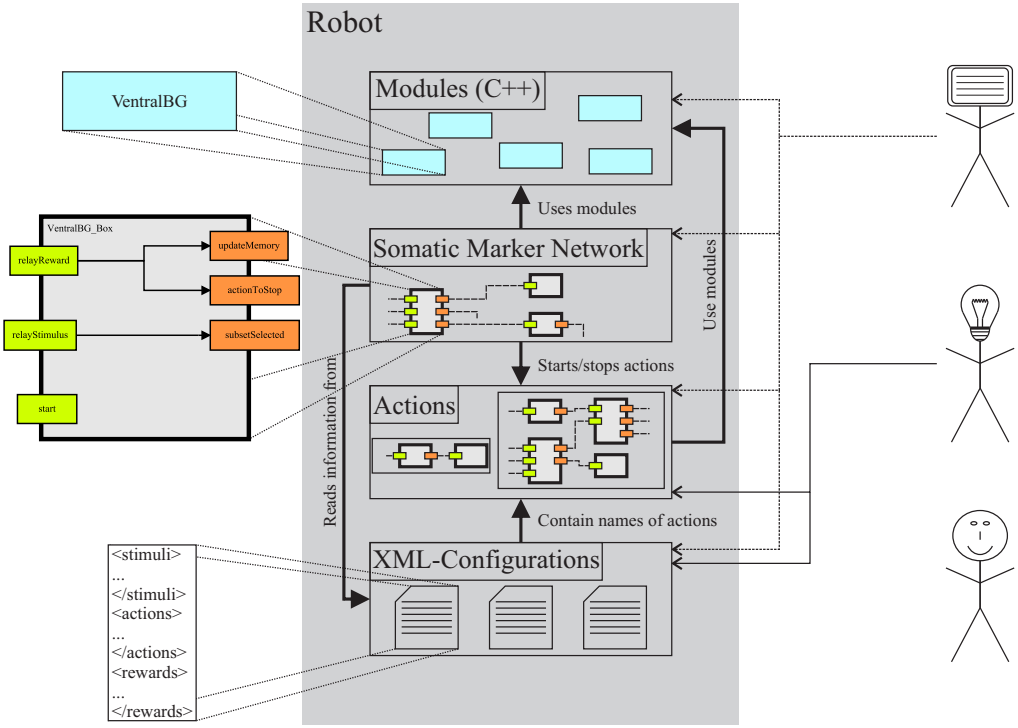

Figure 4.16: Overview that shows how users of the type *Computer Scientist* and *Researcher* are able to enhance the robot's capabilities.

4.8 Summary

In this chapter the implementation of the decision making algorithm on the humanoid NAO robot has been presented. Figure 4.16 gives an overview on all parts of the decision making framework presented previously. Furthermore, the figure shows which types of users are already able to work with the decision making framework. The basis for the whole decision making framework are the different modules which have been presented in the sections 4.3, 4.4, 4.5 and 4.6. Each module is written in C++ and is executed directly on the robot. The modification of the modules is reserved for users of the type *Computer Scientist*.

An exemplary modification could be the implementation of a different action selection mechanism based on the somatic markers. Due the modular software architecture, a computer scientist just needs to know that the action selection is performed in the *ventralBG*. With this information a computer scientist is able to modify the current algorithm or to replace the whole module. The replacement of a module is possible without any problems as long as the interface, defined by the corresponding box within the *Somatic*

Marker Network, is not changed. For instance, the *ventralBG* get the current stimulus s_i as input and send a subset A' to the *Rational Analysis*. In the same way, modifications of the *Somatic Marker Network* are reserved to users of the type *Computer Scientist*. In the case that a module is replaced, the corresponding box may have to be modified when functions or parameters are changed. At least the parameter *Modulename* of the corresponding box needs to be changed to ensure that a proxy object of the new module is gathered.

In contrast to *Modules* and the *Somatic Marker Network*, the creation and modification of *Actions* can be performed by users of the type *Computer Scientist* as well as of the type *Researcher*. To create a new action, the GUI of the *Choregraphe* software is used. The creation of actions does not require any knowledge about the functionality of the decision making algorithm. In order to create applications, such as the IGT, an XML configuration file containing information about the recognizable stimuli, available actions and possibilities of giving rewards to the robot has to be created. Even this step can be performed by users of the types *Computer Scientist* and *Researcher*. Due to this possibility, new applications can be created quickly without any programming being necessary. Furthermore, the user does not need to care about the synchronization of parallel processes as this is handled completely by the *Modules* and the *Somatic Marker Network*. Of course knowledge about e.g. sensors is needed. More concretely this means that the user has to know which range of values exists for a specific sensor in order to create appropriate conditions for stimuli or rewards. When the output of a module is supposed to be used within a condition (e.g. SpeechRecognition), the user has to know the output's data type.

Although, the XML file is already a powerful interface to create new applications quickly and easily, users of the type *Normal User* still have no connection to the system. This is due to the reasons that the creation of the configuration files still requires too much knowledge about sensors and so on. Furthermore, the writing of XML code is unacceptable for users of the type *Normal User*. In order to give users of the type *Normal User* the possibility of creating new applications and enhancing the robot's capabilities, a configuration tool which is presented in the next chapter has been developed.

Chapter 5

Configuration Tool

In the previous chapters the decision making algorithm as well as the implementation of the same on the humanoid NAO robot has been presented. For now, the presented evaluations are focused on the performance of the decision making algorithm and not on the usability of the framework. The results presented previously provide a good basis to start with the inclusion of users of the type *Normal User*.

In sum, the presented results show that the algorithm's decisions are comparable to those made by humans, at least to a certain extent. This can be concluded from the evaluation in section 2.7 which is based on the comparison of human results and the agent's results when solving the IGT. This finding is supported by the results of the study presented in chapter 3. Beside these findings, which are admittedly more important for studies concerning HRI aspects than for studies focusing on usability, the evaluation reveals information about the sensitivity of the user-given parameter c. It is shown that the only user-given parameter c is not very sensitive, as it does not have a huge impact on the results. Consequently, a default value can be used for this parameter and its existence can be hidden from the user. All remaining parameters are adapted automatically by the algorithm. This is an essential aspect in order to use this algorithm for robot companions and simultaneously allow users without programming skills or technical expertise to enhance the applications of the robot to personal needs.

5.1 Inclusion of Normal Users

In order to create applications it is shown how to define stimuli, actions and possibilities of giving rewards. All these definitions are written into an XML file which contains all necessary information for the robot. The user does not

have to deal with any further details like synchronization of parallel processes. Due to the XML interface, the user is able to combine any available sensory information or functionalities of modules as desired.

Still, the use of the framework requires too much expertise, as the user e.g. needs to know which sensors exist and how they are named. Even the creation of an XML file itself and the transfer of the same onto the robot is unacceptable. For these reasons, a tool which assists the user in the creation of its own applications is developed. The tool creates an XML file according to the user's inputs, transfers it to the robot and starts the application. Therefore, the user does not need knowledge about the existence of the XML file at all.

Figure 5.1 shows an overview of the whole system with the addition of the configuration tool (*Conf-Tool*). The *Conf-Tool* is not a part of the robot but can be used to create XML configuration files. With this tool even users of the type *Normal User* are able to create new applications. Needless to say, that also users of the types *Computer Scientist* and *Researcher* are able to use the tool in order to create applications.

Basically, the tool gives users the possibility of defining stimuli, actions and rewards. In general, each user should benefit from the works made by other users, especially from those made by users with greater competences.

For instance, a module that classifies fruits is developed and provided by a computer scientist. The module writes its classification result into the *ALMemory* which can be used by a researcher or a normal user in order to create stimuli (the classification result is used within a condition).

Another example is an action that makes the robot able to put a fruit into a fruit bowl. This action is developed by a researcher and can be used by normal users within an application (e.g. sort fruits).

Preliminary study: As a starting point, a tool providing all necessary functionalities was developed without focusing on usability aspects. The usability of this tool was evaluated in a preliminary study in order to get information for constructing a user-friendly tool. For this purpose 5 participants (all students) have been asked to solve a specific task with the tool.

Each interaction with the tool lasted approximately 30 minutes. The participants were told to think aloud while solving the task. All participants were interviewed after the task in order to get feedback about the usability. Finally, the participants were able to ask questions. For a subsequent analysis, the audio and the screen were recorded during the whole procedure.

The analysis of the results reveals different problems which led to difficulties during the processing of the task. One of the occurring problems was

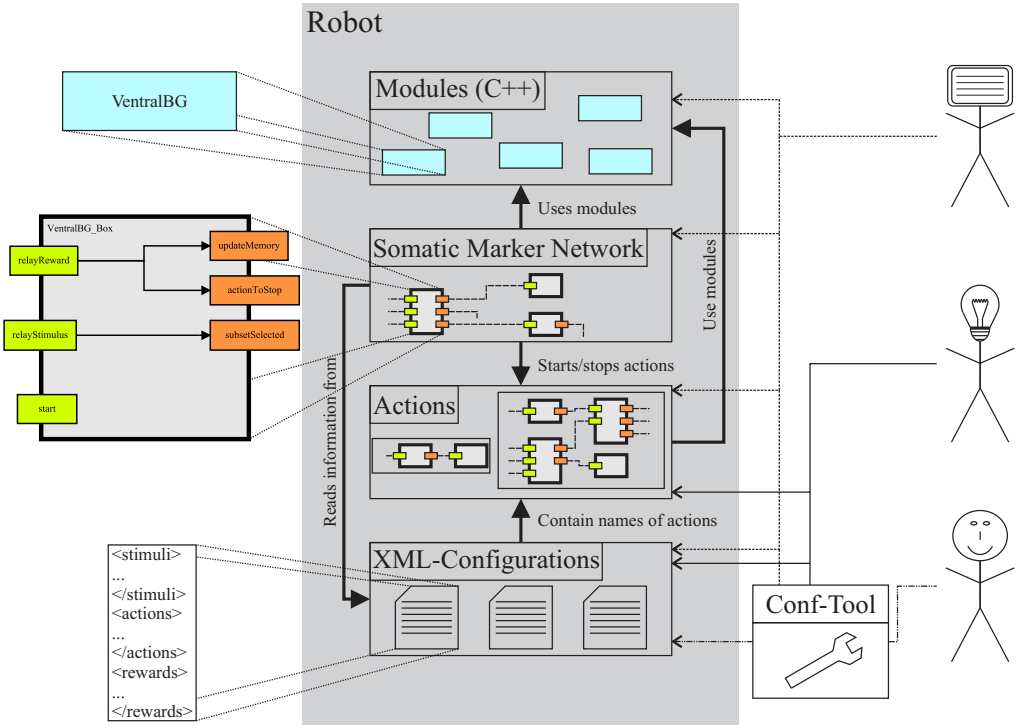

Figure 5.1: Overview that shows how all types of users are able to enhance the robot's capabilities.

the participants having difficulties with the navigation within the tool. These problems can be traced back to a missing overview on the single steps that have to be processed.

A more severe problem was the participants' confusion about some terms like 'stimulus'. Due to the fact, that there were no or insufficient explanations of the whole learning concept and the used terms, the participants were not able to understand the semantic meaning of the single processing steps.

Based on the preliminary study's results, a redesigned tool was developed to overcome the previously discussed problems. In the following the different components of the redesigned tool are described. Subsequent to that, an evaluation concerning the usability of the tool is presented.

5.2 Implementation Details

An overview on the tool is given in figure 5.2. The tool is written in Java and uses JavaFX for the graphical user interface. The main menu consists of seven items. In the following, the functionality of each menu item is briefly

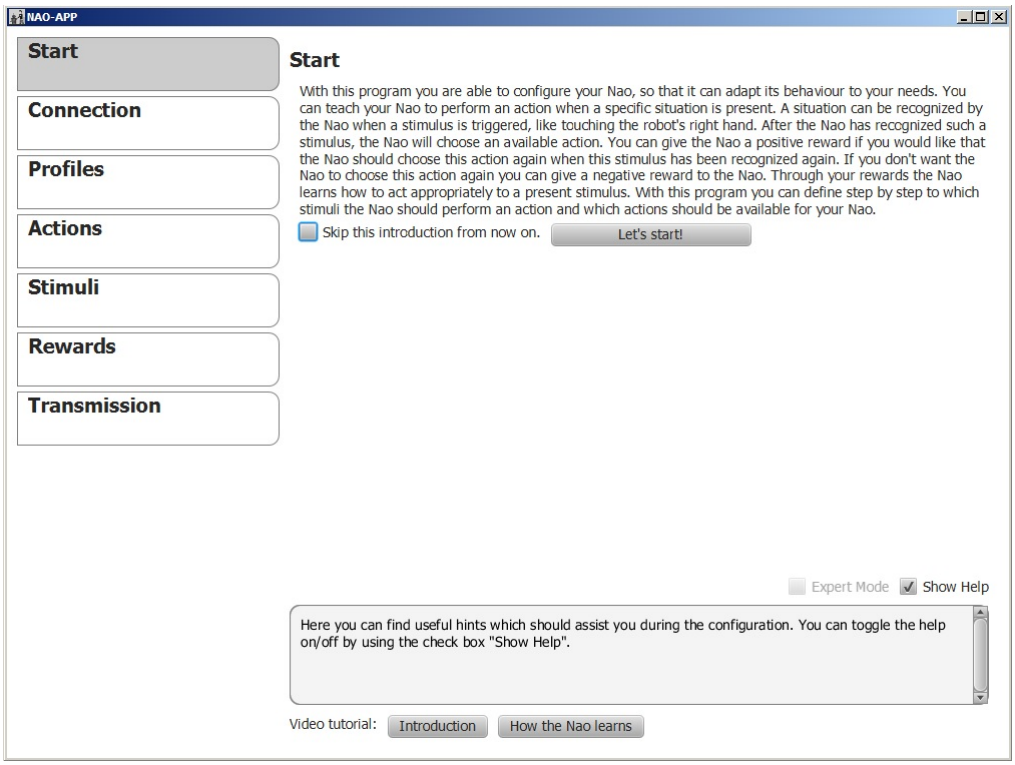

Figure 5.2: Start screen of the *Conf-Tool*. The main area offers information and elements for assistance, while navigation is shown on the left hand side.

described before the evaluation is presented. The tool offers an *Expert Mode* which can be used by experienced user to configure an application. As this mode did not play any role for the study it is not explained any further.

1. Start: When the program is started, the user gets information about the functional range of the tool and general information about the framework. This information includes a short description of stimuli, actions and the possibilities of giving the robot rewards in order to influence its behaviour. Furthermore, the user can start two short videos one of which also gives an overview on the tool, while the other provides a more detailed explanation on how the robot learns.

2. Connection: If one or multiple robots are powered on, the user is able to connect to one specific via the menu item *Connection*. The tool automatically checks which robots are available and shows them to the user. For more

```
  .
  .
  .
<condition>
  <currentValue>LeftBumperPressed</currentValue>
  <operator>==</operator>
  <value>1</value>
</condition>
  .
  .
  .
```

Listing 5.1: Exemplary condition for which a self-documenting alias is existing.

information, the user is able to start a tutorial video which shows how to connect to a robot.

3. Profiles: A profile represents a completely configured application which includes the definitions of recognizable stimuli, executable actions and possibilities of obtaining rewards. Therefore, each profile is saved into a separate XML configuration. The user is able to save completed configurations and to load them in order to do some enhancements or modifications. For getting more information about profiles the users can open a tutorial video.

4. Actions: The tool automatically checks which actions are installed on the robot and displays them to the user. The user then is able to choose the relevant actions for the application. Thus, the user defines the set of actions A (see section 2.4). An exemplary entry within the XML file is shown in section 4.4 (see listing 4.4). As for the other menu items, which are described before, a tutorial video is available which can be used by the user to get more information.

5. Stimuli: In addition to the set of actions A, the user has to define the set of stimuli S (see section 4.3). In order to assist the user, self-documenting aliases for different sensors as well as for complete conditions are offered to the user.

For example, the user can choose *left hand pressed* as a condition which is transformed internally into the XML code shown in listing 5.1. Therefore, the user does not need to know, the name of the sensor's variable (LeftBumperPressed), the relational operator (==) and the sensor's range of values (0 or 1). Even for this menu item a tutorial video is available to assist the user.

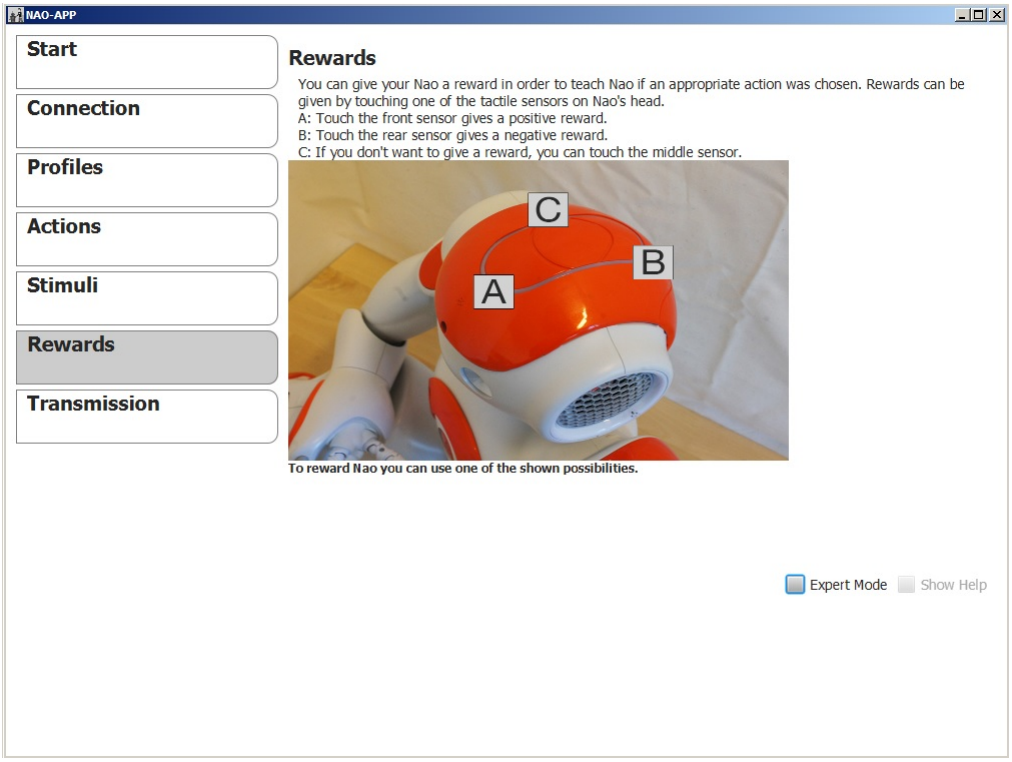

Figure 5.3: Shows the default configuration of the reward possibilities.

6. Rewards: Before the configuration of an application is finished, the user has to define how the robot can be rewarded. If the expert mode is deactivated, a default configuration is used which allows users to give rewards via the tactile sensors on the robot's head (see figure 5.3). For the study, only the default mode was used, therefore this configuration step was of a purely informative character and did not require any actions by the user.

7. Transmission: In order to start the application, the user is able to transfer the created configuration to the robot. After the transmission is finished, the robot restarts its *NAOqi* (see section 4.1) and loads the information from the XML file. Furthermore, the user is advised to get the robot into a safe position for the restart. If a specific action called *safePosition* is available on the robot, this action is executed automatically before the restart.

5.3 Usability Study

The tool offers all functionalities for configuring applications without programming. A usability study was conducted in order to evaluate if users without programming skills or technical expertise are able to use this tool.

It is expected that especially elder people might have more difficulties in using new technologies. However, there are a variety of applications for robot companions which deal with issues especially concerning the elderly such as home care. Due to that reason, it is important to include the elderly into the development process of robot companions.

Furthermore it is expected, that it is easier for younger people to use technologies that the elderly are able to use than the other way round. Due to this reason, the presented usability study was conducted with participants at the age of 40 or older. In total 5 participants (2 male; 3 female) were recruited.

5.4 General Procedure

Each participant was told to solve an openly designed scenario that consisted of creating two different profiles, one for the summer and one for the winter. There were no further restrictions regarding the stimuli or actions the participants had to use.

The study focused exclusively on the usability of the configuration tool and did not include any interaction with the robot. During the whole procedure the participants were told to think aloud and the interaction was recorded (screen and audio) for subsequent analyses. An examiner was present while the participants tried to solve the task.

A limited range of actions and conditions from which the participants were able to choose was given. When the menu item *Actions* was active, the participants could choose from 9 different actions which were: *clear the snow, scrape the ice off, brush the terrace, water flowers, brush the leaves, plant flowers, hide Easter eggs, wash up* and *vacuum*.

For the definition of stimuli, the participants were able to choose from four different conditions which were: *right foot is pressed, left foot is pressed, right hand is pressed* and *left hand is pressed*. Each of the selectable conditions represented a user friendly description of an entry for the XML file. For instance, the choice of the condition *left foot is pressed* led to the entry within the XML file shown in listing 5.1.

The user did not have to define how the robot can obtain rewards as the tool offers a default configuration when the menu item *Rewards* is active.

Here, the participant was able to see that the robot can be rewarded by pressing one of the tactile sensors on its head (see figure 5.3). As mentioned before, it was not specified which stimuli the participants had to create or which actions they had to use.

Subsequent to the task, each subject was interviewed in order to gather further information. Finally, the participant was fully debriefed and thanked for the participation. The results are presented in the following section

5.5 Results and Discussion

The analysis of the recorded material and the interviews reveals that the participants had problems to solve the task on their own. Although explanations and video tutorials about the semantic meaning of every step were available, the subjects still had difficulties to understand the learning concept. Especially, the meaning and definition of stimuli led to major problems.

It was observed that the assistance in the form of help text and videos was only used partially. Furthermore, some participants did not understand that the videos only show examples of how to configure the robot. Beside the problems concerning the understanding of learning concept, difficulties when using the tool were observed. These difficulties partially result from the missing understanding of the learning concept. In general, the participants often were confused about their progress in solving the task. Based on these findings it can be concluded, that the offered assistance of the current tool is not sufficient and needs to be improved. Especially, the placement and the kind of assistance have to be revised in order to ensure that the users understand the learning concept.

Maybe it is advisable that the user has to solve a tutorial when the tool is started the first time. At each step a specific task has to be solved correctly before the next step becomes available. This tutorial may include non-optional videos in order to ensure that the assistance is used. Furthermore, better feedback mechanisms are necessary to inform the users about their progress.

Although the use of the tool still seems to be too difficult for people of this age group, it offers a suitable basis for further research. It is part of further research if the use of better feedback mechanisms and assistance helps to overcome the observed difficulties. Furthermore, studies which include other age groups are of great interest in order to evaluate which functional range the tool should offer for different age groups.

In order to focus on the usability of the tool within this study, there was no interaction with robot included. It is assumed that a previous interaction with a robot performing a sample application could improve the subjects'

capabilities of using the configuration tool. This is a point of contact for subsequent research. Due to the fact that new applications can be created very fast and easily with this tool, it can assist psychologists to create studies concerning human-robot interaction.

In the following chapters two studies which were conducted to gain information about the decision making framework concerning human-robot interaction aspects are presented. For both studies, it is possible to create the interaction scenario using the presented configuration tool. As the tool had not been finished at the time when the studies presented in the following were conducted, the XML configuration files were created manually which is not necessary in the future anymore.

Chapter 6

Human-Robot Interaction (Playful Scenario)

As already mentioned in previous chapters, it is essential to include the user's view when creating robot companions. For that purpose two studies with respect to human-robot interaction are presented in this and in the following chapter. In both studies, participants had to interact with the humanoid NAO robot (see section 4.1) and were able to reward the robot in order to influence its decisions. The main aim of the studies was to reveal how participants perceive the robot's learning capabilities. The findings can give important evidence of necessary modifications or enhancements of the decision making algorithm helping to increase the users' acceptance of the robot. It is expected that the context of the interaction itself has an influence on the results, especially on the perception of the robot's learning capabilities. Due to that reason, each of the two studies focused on a different context.

In the first study, a playful interaction was used. The participants had to play the card game 17+4 with the robot. This game is comparable to blackjack but with some simplifications and modifications which were made for the experiment. One important modification was that the participants did not play against the robot, but should teach the robot how to make advantageous decisions. This created a cooperative playful interaction.

The second study is presented in chapter 7 and was focused on everyday social interactions. In the second study, three different interactions were used. At first, the participants had to teach the robot to choose an appropriate greeting. Subsequent to that, the participants had to teach the robot to choose appropriate depositories for different kinds of objects. During the last interaction, the robot should learn which hobby the participants preferred.

As for the evaluation of the configuration tool, participants of the age group 40+ were recruited in order to interact with the robot. This choice

was made as it is expected that especially the elderly have more difficulties with accepting and understanding new technologies. Furthermore, there are a lot of interesting applications for robot companions in conjunction with elderly people. An example for such an application and an experiment on human-robot interaction can be found in Wada et al. (2005).

In order to create the interacting scenarios, several XML configuration files[1] were created in advance, according to the structure presented in sections 4.3, 4.4 and 4.6. As the focus of the studies presented in this and in the following chapter was on HRI aspects, the creation of the XML configuration files was not the duty of the participants. Therefore, the participants were able focus exclusively on the interaction.

6.1 Setting and General Procedure

Initially, the participants were given an instruction on the study and informed that the whole interaction would be videotaped for evaluation purposes. Furthermore, they were notified that all gathered data were rendered anonymous and treated as confidential. Finally, the participants had to give a written consent for the participation. In order to gather demographic information as well as data on the participants' attitudes and anxieties towards robots, the participants were asked to answer a questionnaire before interacting with the robot.

As part of the interaction with the robot, they were asked to imagine that they own this robot and use it at home for different purposes. The aim of the interaction was to teach the robot 17+4, so that the participants' grandchildren might play the game with the robot during their next visit. Goal of the game 17+4 is to reach an accumulated card value that comes close to or is exactly 21. The game is lost when the value 21 is exceeded. After each chosen card, the player (here the robot) can decide whether to take a further card or to hold the reached accumulated card value. If the player decides against an additional card the current round ends. Figure 6.1 shows what the participants saw during the interaction. For each accumulated card value a separate card that could be shown to the robot was available. Furthermore, a card showing a sad smiley was available. This card was shown to the robot, when the accumulated card value exceeded 21 in order to signalize that the game was lost.

[1] In general, it is possible to create the different XML configuration files using the tool presented in chapter 5. As the tool was developed collaterally to the human-robot interaction studies, the XML configuration files were created without the tool.

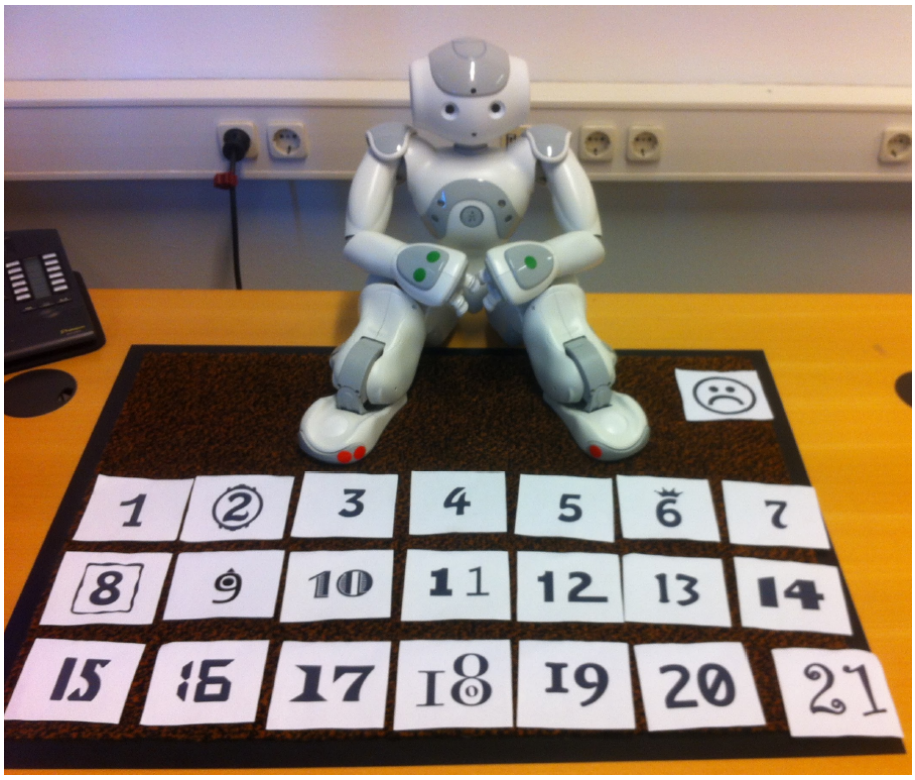

Figure 6.1: Shows what the participants saw during the interaction with the robot.

As the robot had no knowledge about the game, it was the participant's duty to teach the robot in which cases it is advisable to take a further card or to end the round. For this purpose the participant was able to reward the robot after each decision via the robot's tactile sensors. For a better identification of the connection between a particular sensor and the corresponding reward, the tactile sensors were marked with coloured dots (see figure 6.1). Four different reward options were available which were either strongly negative (two red dots), slightly negative (one red dot), slightly positive (one green dot) or strongly positive (two green dots).

The card decks, from which the cards were chosen, contained the values 1 to 9, which was known by the participant. Due to that reason, it was always uncritical to choose a further card as long as the accumulated card value was equal to or smaller than 12. If the accumulated card value exceeded 12, the risk of losing the game increased the closer the accumulated card value came to 21. Even when an accumulated card value of 21 was reached, the robot was able to choose a further card which was obviously a disadvantageous decision.

	Deck 1	**Deck 2**	$\cdots$	**Deck 15**
first card	6	9	$\cdots$	$\cdots$
second card	7	4	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
sum first+second	13	13	$\cdots$	$\cdots$

Table 6.1: Exemplary decks for the game 17+4.

In total 15 rounds were played. For each round an own card deck was used. In order to create equal conditions for each participant, the decks were prepared in such a way, that the same order of cards was used for every participant. Each round started with the investigator drawing a card and showing it to the participant. Based on the drawn card, the participant had to add the shown value to the current accumulated card value (0 at the beginning). Subsequently, the participant had to take the card which was correspondent to the accumulated value and had to show the card to the NAO robot. After the robot had made a decision, the participant had to reward the robot. When the robot asked for a further card, the next card was drawn, its value was added to the current accumulated card value and the participant had to show the card with the accumulated card value to the robot. This procedure was repeated until the robot decided against a further card or the game was lost because the accumulated card value exceeded 21. In the case that the game was lost, the participant had to show the card with the sad smiley to the robot.

In order to make the robot's learning process observable more quickly, it was assured that specific accumulated card values had the chance to occur more frequently. Of course, the occurrence of the values was also dependent on the robot's decisions which were influenced by the obtained rewards. An example is shown in table 6.1. In this example, it is possible to reach an accumulated card value of 13 in the first and in the second round. However, it is not guaranteed that the value 13 is reached as the robot e.g. can decide to end the round after the first card of deck 1 and also after the first card of deck 2.

After 15 rounds had been played, the participant had to answer a questionnaire which included items concerning the robot's learning ability. Finally, the participant was fully debriefed and thanked for the participation.

6.2 Implementation of the Card Game

In this section some details about the configuration of the scenario are given. According to the presented decision making framework, the following description of the game 17+4 was used:

- $S = \{\underbrace{Value1}_{s_1}, \underbrace{Value2}_{s_2}, \ldots, \underbrace{Value21}_{s_{21}}\}$
- $A = \{\underbrace{TakeCard}_{a_1}, \underbrace{HoldCards}_{a_2}\}$
- $R_{s_1} = R_{s_2} = \ldots = R_{s_{21}} = \{-100, -50, 50, 100\}$

In order to reduce the technical effort, the robot did not need to sum up the shown card values on its own but received the accumulated card value through a stimulus given by the participant. For each accumulated card value a stimulus existed ($s_1, ..., s_{21}$). The NAO software offers an easy possibility of letting the robot recognize objects based on visual features. Once an object has been learned, it can be recognized by the robot. Therefore, all cards shown in figure 6.1 were added to the vision recognition database of the robot. In addition, the robot was able to recognize a card which shows a sad smiley. When the sad smiley was recognized by the robot, a pre-wired behaviour was started which led to the robot asking the participant to start a new round. As the robot's reaction to this card should not be part of the learning process, the sad smiley was not defined as a stimulus. The robot could perform two different actions. It could either choose a further card or it could hold its cards to end the current round.

Listing 6.1 shows the XML configuration file that was used for this scenario. First of all the definition of the stimuli can be seen. The variable *PictureDetected* is an array which is updated when the robot has recognized an object that was included in the database. The operator *image* is a self-defined operator that compares the name of the currently detected picture to the defined comparative value (e.g. one). Furthermore, the XML configuration file contains the actions *TakeCard* and *HoldCards*. Each action is a separate behaviour network which was created using the *Choregraphe*. When the action *TakeCard* is performed the robot asks for a further card. The action *HoldCards* lets the robot say that it would like to end this round. Finally, the four possibilities of rewarding the robot can be found in the configuration file. Each time the robot had made a decision, the participants were able to give a reward by pressing one of the defined sensors.

```
<configuration>
  <stimuli>
    <stimulus name="Value1" neededConditions="1" timeDifference="5"
      maxReward="100" rewardResolution="10">
      <condition>
        <currentValue>PictureDetected</currentValue>
        <operator>image</operator>
        <value>one</value>
      </condition>
    </stimulus>
    .
    .
    .

    <stimulus name="Value21" neededConditions="1" timeDifference="5"
      maxReward="100" rewardResolution="10">
      <condition>
        <currentValue>PictureDetected</currentValue>
        <operator>image</operator>
        <value>twenty-one</value>
      </condition>
    </stimulus>
  </stimuli>
  <actions>
    <action>TakeCard</action>
    <action>HoldCards</action>
  </actions>
  <rewards>
    <reward rewardValue="100">
      <triggerName>HandRightBackTouched</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
    <reward rewardValue="50">
      <triggerName>HandLeftBackTouched</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
    <reward rewardValue="-50">
      <triggerName>LeftBumperPressed</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
    <reward rewardValue="-100">
      <triggerName>RightBumperPressed</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
  </rewards>
</configuration>
```

Listing 6.1: XML configuration file that was used for the card game 17+4.

6.3 Measurements and Participants

Moderating variables: In total 21 participants (10 male; 11 female) between 46-75 years (MEAN = 58.33, SD = 7.638) were recruited. In order to gather more information on the sample, different moderating variables concerning the participants' negative attitudes toward robots and anxieties toward robots were collected.

The NARS (Negative Attitudes Towards Robots Scale) questionnaire, consisting of 14 items, was used to gather information about the participants' attitudes towards robots (Nomura et al. (2006b)). All items had to be rated on a 5-point Likert scale (1=strongly disagree; 5=strongly agree). The NARS questionnaire consists of three sub dimensions:

- *S1(NARS): Negative Attitudes toward Situations and Interactions with Robots* (6 items, e.g. "I would feel very nervous just standing in front of a robot.", Cronbach's $\alpha = .853$). The item "The word robot means nothing to me" was excluded from the analysis in order to increase the scale's reliability.

- *S2(NARS): Negative Attitudes toward Social Influence of Robots* (5 items, e.g. "I feel that if I depend on robots too much, something bad might happen.", Cronbach's $\alpha = .821$)

- *S3(NARS): Negative Attitudes toward Emotions in Interaction with Robots* (3 items, e.g. "If robots had emotions, I would be able to make friends with them.", Cronbach's $\alpha = .709$)

Figure 6.2 shows the results of the NARS questionnaire. Moderate negative attitudes towards situations and interactions with robots (S1), social influence of robots (S2) and emotions in interaction with robots (S3) are observable.

In addition to the NARS, the RA (Measurement of Anxiety toward Robots) questionnaire was used which reveals information about specific anxieties towards robots (Nomura et al. (2006a)). The RA consists of 11 items which are rated on a 6-point scale (1=not uncomfortable at all; 6=very uncomfortable) and groups items into three sub dimensions:

- *S1(RA): Anxiety toward Communication Capability of Robots* (3 items, e.g. "Conversation with robots may be inflexible.", Cronbach's $\alpha = .875$)

- *S2(RA): Anxiety toward Behavioural Characteristics of Robots* (4 items, e.g. "How robots will act.", Cronbach's $\alpha = .911$)

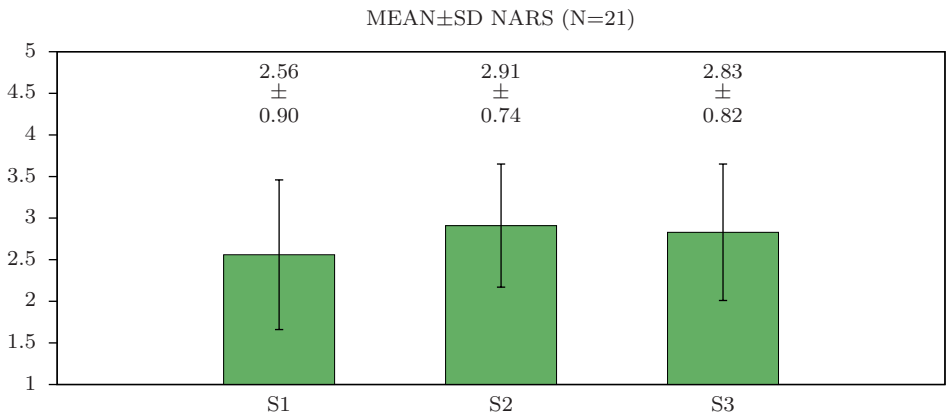

Figure 6.2: Results of the NARS for the playful interaction.

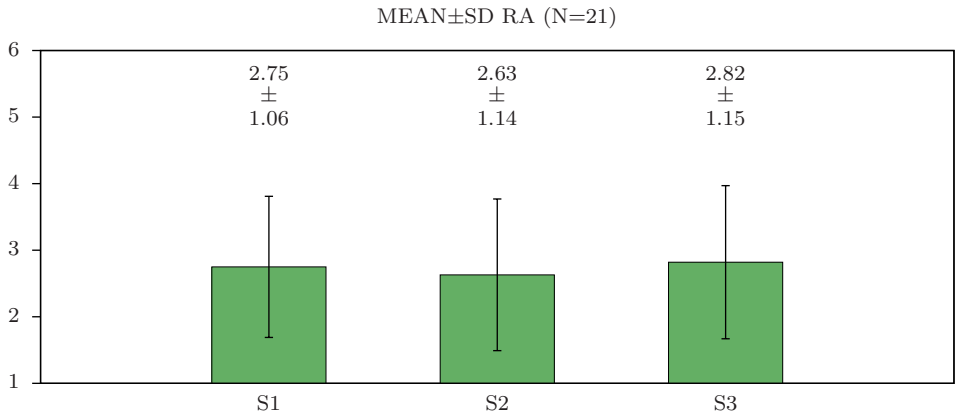

Figure 6.3: Results of the RA for the playful interaction.

- *S3(RA): Anxiety toward Discourse with Robots* (4 items, e.g. "How I should talk with robots.", Cronbach's $\alpha = .949$)

The results of the RA (see figure 6.3) show moderate anxieties toward the communication capability of robots (S1), behavioural characteristics of robots (S2) and the discourse with robots (S3).

Based on this information it can be noted that the sample consists of participants which stated to have moderate negative attitudes as well as moderate anxieties towards robots.

Item	Scale
Did you recognize the robot's learning progress?	Yes / No
I have perceived the learning progress as being...	1 = very fast 5 = very slow
How would you rate the robot's learning ability?	1 = very good 5 = very bad
Do you think that the robot's way to learn is expedient?	1 = very expedient 5 = not expedient
Did you use the different rewards' gradations?	Yes / No
Do you think that the rewards' gradations are expedient?	Yes / No
I could imagine interacting with such a robot in my everyday life.	1 = absolutely not 5 = definitely

Table 6.2: Items which were asked for gathering information about the perception and acceptance of the robot's decision making behaviour.

Dependent variables: As the study was conducted in order to gather information on the perception and acceptance of the robot's learning behaviour, some self-generated items were included as well. These items, shown in table 6.2, served to give insight into the participants' capability of recognizing the robot's learning process at all and into their general perception of the robot's learning abilities. In addition to that, some of the items aimed at finding out whether the participants had used a variety of the available gradations of the different rewards and whether the gradations were regarded as expedient. One of the items was also meant to reveal if the participants could imagine to use such a robot in their daily life.

6.4 Results and Discussion

Table 6.3 shows results of the questions concerning the robot's learning behaviour. Only 52.4% of the subjects stated that they had recognized the robot's learning progress. There are plausible explanations for this small number. Although the decks were prepared in such a way that specific stimuli were very likely to occur, it was not possible to guarantee that a stimulus occurred more than once. In unfavourable cases this led to the robot not being able to apply the acquired knowledge at all or only in rare cases. In conclusion, the number of 15 played rounds was too low. Another possible reason for participants not perceiving the robot's learning progress could be the fact that the robot tried every action once before decisions were made based on the somatic markers. This could result in a disadvantageous action being chosen by the robot, although another action had been rewarded positively

Item	Result
Did you recognize the robot's learning progress? (N=21)	Yes (52.4%)
I have perceived the learning progress as being... (N=12)	Moderate MEAN= 2.83 ± .84
How would you rate the robot's learning ability? (N=14)	Moderate MEAN= 2.86 ± .86
Do you think that the robot's way to learn is expedient? (N=18)	very expedient MEAN= 1.11 ± .32
Did you use the different rewards' gradations? (N=20)	Yes (90.0%)
Do you think that the rewards' gradations are expedient? (N=21)	Yes (95.0%)
I could imagine interacting with such a robot in my everyday life. (N=20)	Maybe MEAN= 3.10 ± 1.21

Table 6.3: Results of the participants' perception and acceptance of the robot's decision making behaviour (playful scenario).

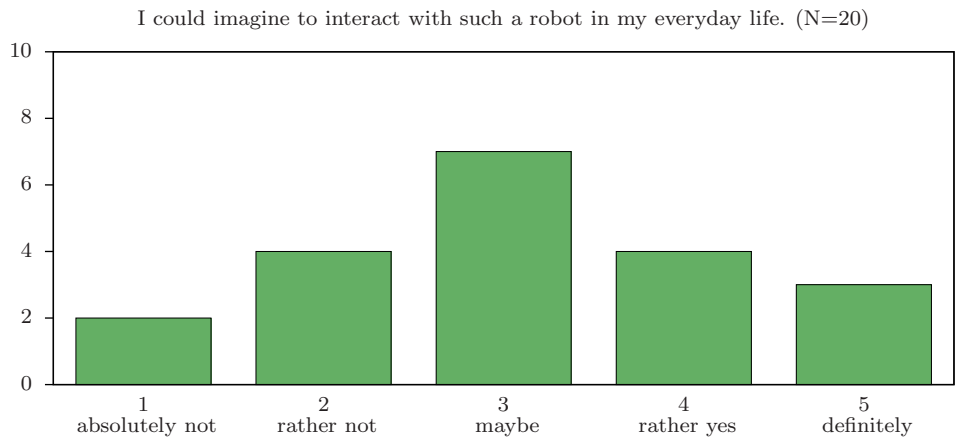

Figure 6.4: Results of the acceptance of the playful interaction.

in the past. Both listed causes, the small number of rounds and trying out of every action once, can also be an explanation for the robot's learning progress as well as the robot's learning ability having been rated as only moderate.

In contrast to that, the robot's way to learn was rated as being very expedient. Most of the participants stated that they had used the different rewards' gradations and deemed them expedient. Regarding the question if the participants could imagine to interact with such a robot in their everyday life, neither a strong refusal nor a strong approval is observable. Figure 6.4 shows the detailed results.

In conclusion, the results show that the participants reacted positively to the robot's way to learn and the used reward mechanisms. Unfortunately,

the robot's learning progress was only perceived by 52.4% of the participants which may result from the small number of rounds having been played and the properties of the algorithm.

In order to reveal how much influence the experimental setup had on the results, a further HRI study was conducted. For the study, which is presented in the following chapter, the decision making algorithm remained unchanged but different interaction scenarios were used. The discussion at the end of chapter 7 includes a comparison of the results from both studies in order to deduce improvements for the algorithm as well as for the experimental designs.

Chapter 7

Human-Robot Interaction (Social Scenario)

While a playful interaction scenario was used in the previously presented study, the study which is described in the following focused on social interaction scenarios. In contrast to the playful interaction, every scenario consisted of an initial learning task and included a subsequent relearning as well. In total three different interaction scenarios, which are described in the following section, were used. Due to the relearning part, the benefit of an adaptable behaviour based on an algorithm, in contrast to pre-wired connections between stimuli and actions, becomes more visible. If pre-wired connections are used, the user is forced to change the configuration every time the robot should act differently. Due to the used decision making framework, the user is able to change the behaviour through interaction and does not need to change any configurations.

7.1 Setting and General Procedure

As in the previously presented study all participants were informed that the entire interaction would be videotaped and that the gathered data would be rendered anonymous and treated as confidential. Initially, each participant had to answer questions concerning demographic information, followed by the NARS and the RA questionnaires. After that, each participant got a short demonstration of the interaction with the robot. Additional information on interacting with the robot was provided via a handout.

Subsequently, the subjects had to interact with robot. Three different interaction scenarios were used. As mentioned before, every interaction scenario consisted of a learning phase and a relearning phase. In all scenarios

the robot was placed on a table and the participant sat in front of the robot. The scenarios partly differed in the number of stimuli and actions. Furthermore, different kinds of stimuli were used (visual or auditive). The participant was able to reward the robot's decisions by touching specific sensors (same sensors as described in section 6.1). Four different reward possibilities were available which were either strong negative, small negative, small positive or strong positive.

Greeting Scenario: During the first scenario, the participants were told to teach the robot to choose an appropriate greeting for themselves. Therefore, the robot was able to react to verbal stimuli when either the words "hello" or "good day" were recognized by the robot. Each time the robot was greeted by the participant, it chose one answer out of seven replies which were "good day", "hello", "hey", "I am hungry", "yooooo", "bye" and "what's up". After the robot had given an answer, the participant rewarded the robot in order to signalize if the answer was appropriate. This procedure was repeated until the participant believed the robot to have learned the greeting preferred by the subject.

After that, the participant was told to imagine that his/her grandchild was going to come for a visit. Now it was the participant's task to teach the robot the preferred greeting for the grandchild. During this process the robot had to relearn through the rewards given by the participant. Again, this procedure was repeated until the participant believed the robot to have learned the greeting preferred for the grandchild. This was the end of the first interaction scenario and the participant had to fill in a short questionnaire which included items to rate the robot's learning ability. More details about the questionnaire are presented in section 7.3.

Sorting Scenario: Right after the participant had finished the questionnaire, the second interaction scenario began with a short introduction given by the investigator. The context of the second scenario was to teach the robot appropriate depositories for different objects. In this scenario the robot was able to recognize images which showed three different objects (pen, apple and book). Every time the robot recognized an object, it answered with one out of five different depositories which were "fruit bowl", "shelf", "rubbish bin", "desk" and "pencil case". The robot performed a pointing gesture, while giving the answer. Figure 7.1 shows the objects which the participant was able to show to the robot. After each suggestion made by the robot, the participant gave a reward to signalize if the made suggestion was good or

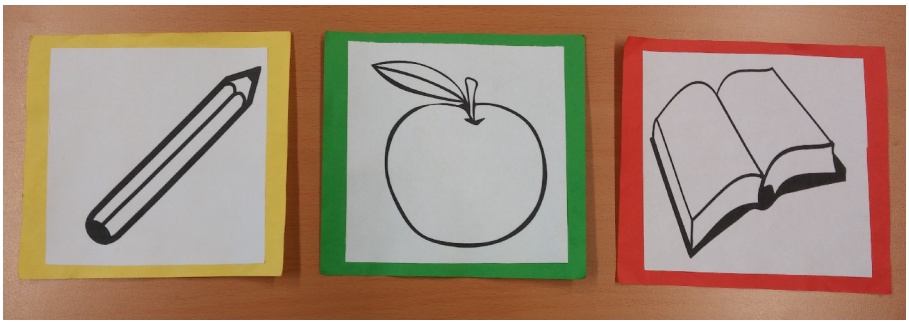

Figure 7.1: These three objects can be recognized by the robot.

not. The whole procedure was repeated until the participant believed that the robot had learned the favoured depositories for the different objects.

Subsequently, the participant was told to rearrange the objects and to teach the robot the new depositories. This started the relearning phase which again was repeated until the participant believed that the robot had learned the new depositories. Afterwards, the participant had to rate the robot's learning ability with a short questionnaire.

Hobby Scenario: In the last scenario, the robot should learn the favourite hobby of the participant. Every time the participant said the words "idea" or "further", the robot reacted with a suggestion for a leisure activity. The robot had eight different suggestions available which were "theatre", "zoo", "riddle", "pub", "smoking", "tv", "pairs" and "computer". After each suggestion the participant rewarded the robot to signalize if the made suggestion was good or not. The whole procedure was repeated until the participant believed that the robot had learned the favoured hobby.

Subsequently, the participant was asked to teach the robot to suggest an appropriate leisure activity for his/her grandchild. Therefore, the robot had to relearn based on the rewards given by the participant. This relearning phase lasted until the participant believed the robot to have learned the grandchild's favoured hobby.

Just as after the previous scenarios, the participant had to fill in a short questionnaire concerning the robot's learning ability followed by a final questionnaire. In contrast to the previously presented study (see chapter 6), the final questionnaire again contained the items of the NARS and RA in order to compare the results before and after the interactions. In addition, items to rate the robot's learning ability were part of the questionnaire (more details in section 7.3). Having answered the questionnaire, the participant was fully debriefed and thanked for the participation.

	Greeting Scenario	**Sorting Scenario**	**Hobby Scenario**
Stimuli $S =$	s_1 = Hello	s_1 = Apple s_2 = Pen s_3 = Book	s_1 = Idea
Actions $A =$	a_1 = greeting_goodday a_2 = greeting_hello a_3 = greeting_hey a_4 = greeting_iamhungry a_5 = greeting_yooooo a_6 = greeting_bye a_7 = greeting_whatsup	a_1 = sort_fruit_bowl a_2 = sort_shelf a_3 = sort_rubbish_bin a_4 = sort_desk a_5 = sort_pencil_case	a_1 = hobby_theater a_2 = hobby_zoo a_3 = hobby_riddle a_4 = hobby_pub a_5 = hobby_smoking a_6 = hobby_tv a_7 = hobby_pairs a_8 = hobby_computer
Rewards $R_{s_i} =$	$R_{s_1} = \{-100, -50, 50, 100\}$	$R_{s_1} = \{-100, -50, 50, 100\}$ $R_{s_2} = \{-100, -50, 50, 100\}$ $R_{s_3} = \{-100, -50, 50, 100\}$	$R_{s_1} = \{-100, -50, 50, 100\}$

Table 7.1: Overview of the social interaction scenarios.

7.2 Implementation of the Social Interactions

In the following, the configurations of the three different interaction scenarios are shown. For each scenario a separate XML configuration file was created. An overview of the three scenarios is shown in table 7.1.

Greeting Scenario: For the first scenario, the robot was able to recognize one stimulus *Hello* which was triggered when the robot was greeted either with the words "hello" or "good day". The robot had seven different answers available $(a_1, \cdots, a_7)$. Each action consisted of a speech output. As in the previously presented study, the robot was able obtain rewards by being touched either on the tactile sensors on its hands or on the bumpers on its feet. These reward possibilities were also used for the other interaction scenarios. The whole configuration can be seen in listing 7.1. As the XML configuration files of the other scenarios are comparable to the configuration shown in listing 7.1, the illustration of those is omitted in the following.

Sorting Scenario: During the second scenario the robot was able to recognize three different stimuli named *Apple, Pen, Book*. Each of these stimuli was triggered when the robot was shown an image representing either an apple, a pen or a book. These images had previously been added to the robot's vision recognition database (see figure 7.1). The robot was able to suggest five different depositories for the recognized object $(a_1, \cdots, a_5)$. All actions consisted of a speech output and a pointing gesture which were performed simultaneously.

```xml
<configuration>
  <stimuli>
    <stimulus maxReward="100" rewardResolution="10" name="Hello" neededConditions="1"
        timeDifference="7">
      <condition>
        <currentValue>LastWordRecognized</currentValue>
        <operator>==</operator>
        <value>Hello</value>
      </condition>
      <condition>
        <currentValue>LastWordRecognized</currentValue>
        <operator>==</operator>
        <value>Good Day</value>
      </condition>
    </stimulus>
  </stimuli>
  <actions>
    <action>greeting_goodday</action>
    <action>greeting_hello</action>
    <action>greeting_hey</action>
    <action>greeting_iamhungry</action>
    <action>greeting_yooooo</action>
    <action>greeting_bye</action>
    <action>greeting_whatsup</action>
  </actions>
  <rewards>
    <reward rewardValue="-100">
      <triggerName>LeftBumperPressed</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
    <reward rewardValue="-50">
      <triggerName>RightBumperPressed</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
    <reward rewardValue="50">
      <triggerName>HandRightBackTouched</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
    <reward rewardValue="100">
      <triggerName>HandLeftBackTouched</triggerName>
      <operator>==</operator>
      <value>1</value>
    </reward>
  </rewards>
</configuration>
```

Listing 7.1: XML configuration file that was used for the greeting scenario.

Hobby Scenario: In the last scenario the robot was able to recognize one stimulus named *Idea*. The stimulus was triggered when the speech recognition module identified the words "idea" or "further". The robot was able to suggest eight different leisure activities ($a_1, \cdots, a_8$). Each action consisted of a speech output.

7.3 Measurements and Participants

Moderating variables: In total 20 participants[1] (6 male; 14 female) between 42-72 years (MEAN = 55.53, SD = 8.28) had been recruited. The data of one participant was excluded from the analysis due to the reason that major technical issues occurred during the interaction with the robot.

The NARS and RA questionnaires were used in order to get further information about the sample. In contrast to the previously presented study, the participants had to fill in both questionnaires before and after the interaction with the robot in order to see, if the interaction led to significant changes. In this section only the results of the questionnaires before the interaction are presented.

- *S1(NARS): Negative Attitudes toward Situations and Interactions with Robots* (6 items, Cronbach's $\alpha = .752$). The item "The word robot means nothing to me" was excluded from the analysis in order to increase the scale's reliability.
- *S2(NARS): Negative Attitudes toward Social Influence of Robots* (5 items, Cronbach's $\alpha = .405$)
- *S3(NARS): Negative Attitudes toward Emotions in Interaction with Robots* (3 items, Cronbach's $\alpha = .646$)

Figure 7.2 shows the results of the NARS questionnaire. Moderate negatives attitudes toward situations and interactions with robots (S1) and social influence of robots (S2) are observable. However, it has to be noticed that the reliability of S2 is low ($\alpha = .405$). While a higher score for S1 and S2 indicates a more negative attitude towards robots, S3 is an inverse scale which means that a higher score indicates a more positive attitude. Therefore, a slightly positive attitude toward emotions in interaction with robots (S3) can be observed. Apart from the NARS questionnaire, the RA questionnaire had to be filled in by the participants before and after the interaction.

[1] None of these participants took part in the study presented in chapter 6.

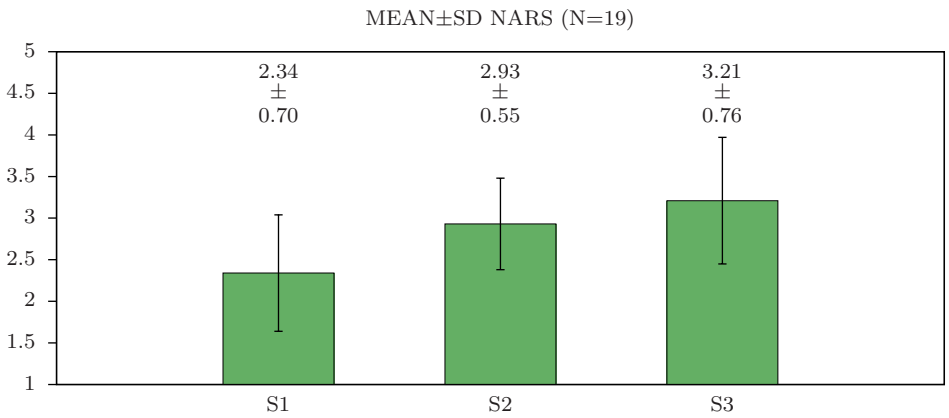

Figure 7.2: Results of the NARS for the social interaction scenarios before the participants interacted with the robot.

- *S1(RA): Anxiety toward Communication Capability of Robots* (3 items, Cronbach's $\alpha = .800$)
- *S2(RA): Anxiety toward Behavioural Characteristics of Robots* (4 items, Cronbach's $\alpha = .845$)
- *S3(RA): Anxiety toward Discourse with Robots* (4 items, Cronbach's $\alpha = .749$)

The results of the RA show moderate anxieties toward the communication capability of robots (S1) and the discourse with robots (S3) (see figure 7.3). Participants stated that they rather did not feel anxious toward behavioural characteristics of robots (S2).

Dependent variables: The same self-generated items which have been presented in the previous chapter were used for the analysis of the perception of the robot's learning ability (see table 6.2). In addition to these items, the participants had to fill in a short questionnaire directly after each of the three different interaction scenarios which included the self-generated items presented in table 7.2. The first three items aimed at gathering information about the participants' perception of the robot's learning ability. Based on this information it is possible to analyse if significant differences can be found between the three interaction scenarios. As technical issues may affect the participants' perception of the robot's learning ability, an item was added to request if noticeable problems had occurred during the interaction.

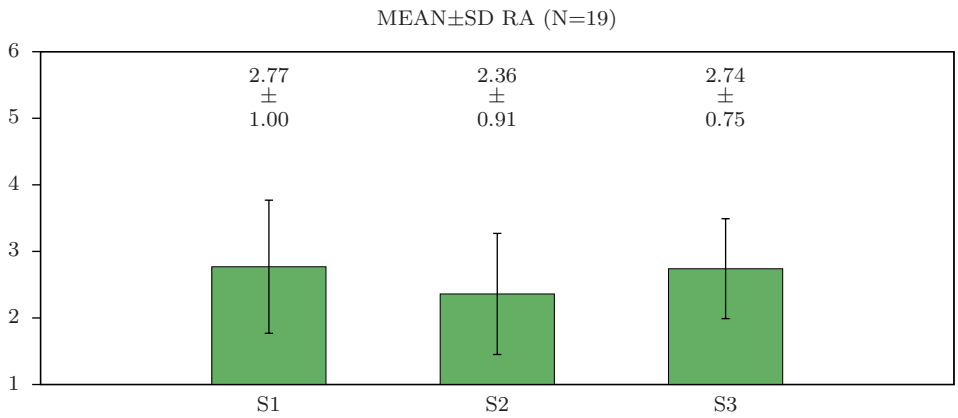

Figure 7.3: Results of the RA for the social interaction scenarios before the participants interacted with the robot.

Item	Scale
How satisfied are you with the robot's learning?	1 = very unsatisfied 5 = very satisfied
Did the robot learn what you had wanted to teach it?	Yes / No
Did you think that the time the robot needed to learn was appropriate?	Yes / No
Did noticeable problems occur during the interaction?	Yes / No

Table 7.2: Items concerning the robot's learning ability which were asked after each interaction scenario.

7.4 Results and Discussion

NARS and RA analysis: In order to reveal information about the participants' attitudes toward robots the NARS was used. The mean values and standard deviations can be seen in figure 7.4. The reliabilities of the NARS-scales after the participants had interacted with the robot are S1: Cronbach's $\alpha = .644$, S2: Cronbach's $\alpha = .704$ and S3: Cronbach's $\alpha = .553$. The repeated-measures t-test for the NARS before and after the interaction reveals no statistically significant effect for each of the three sub scales. Therefore, the interaction did not change the negative attitudes toward the robot.

- *S1(NARS): Negative Attitudes toward Situations and Interactions with Robots:* $t(18) = 1.73, p > .05$

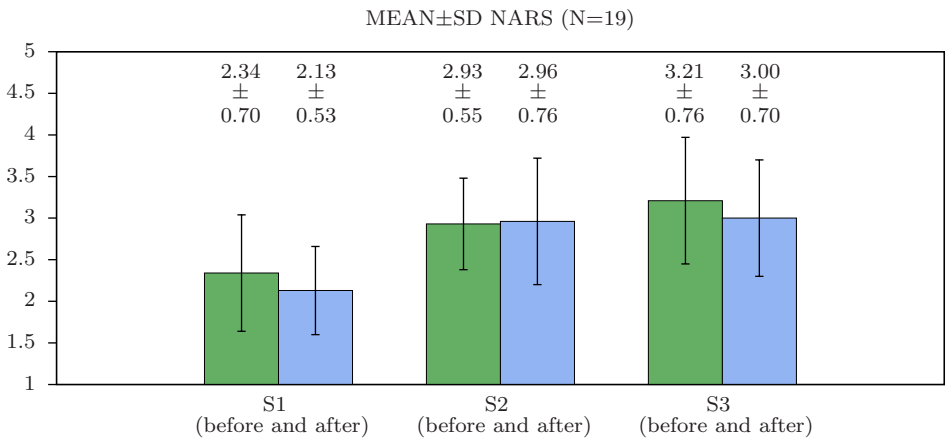

Figure 7.4: Results of the NARS for the social scenarios before and after the interaction with the robot.

- *S2(NARS): Negative Attitudes toward Social Influence of Robots:* $t(18) = -.22, p > .05$
- *S3(NARS): Negative Attitudes toward Emotions in Interaction with Robots:* $t(18) = 1.68, p > .05$

The results of the RA questionnaire are shown in figure 7.5. The reliabilities of the RA-scales after the participants had interacted with the robot are S1: Cronbach's $\alpha = .884$, S2: Cronbach's $\alpha = .869$ and S3: Cronbach's $\alpha = .892$. As for the NARS, a paired t-test was performed based on the data gathered from the RA. No statistically significant changes concerning the participants' anxieties toward the communication capability of robots and toward behavioural characteristics of robots are observable. However, a significant result can be found regarding the participants' anxiety toward the discourse with robots. The participants were less anxious toward the discourse with robots after the interaction.

- *S1(RA): Anxiety toward Communication Capability of Robots:* $t(18) = 1.21, p > .05$
- *S2(RA): Anxiety toward Behavioural Characteristics of Robots:* $t(18) = -.34, p > .05$
- *S3(RA): Anxiety toward Discourse with Robots:* $t(18) = 2.13, p < .05$

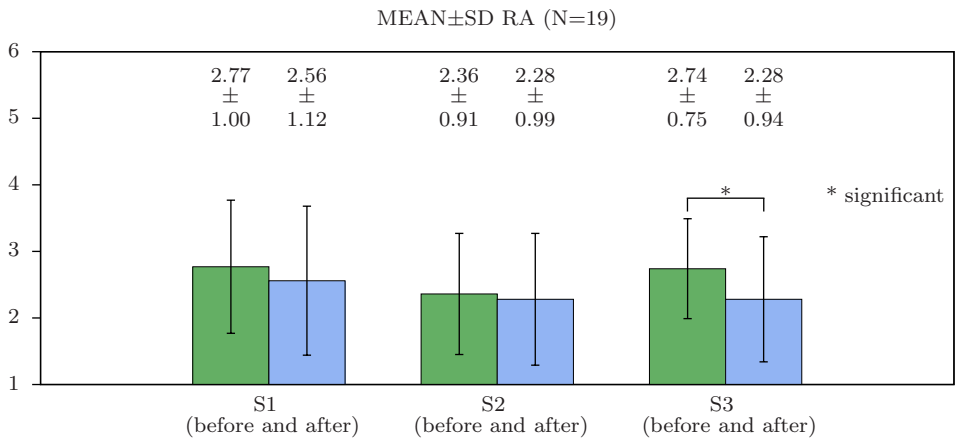

Figure 7.5: Results of the RA for the social scenarios before and after the interaction with the robot.

	Greeting Scenario	**Sorting Scenario**	**Hobby Scenario**
How satisfied are you with the robot's learning behaviour?	Satisfied MEAN= 4.26 ± .56	Satisfied MEAN= 4.00 ± .75	Satisfied MEAN= 4.16 ± .96
Did the robot learn what you had wanted to teach it?	Yes 100%	Yes 100%	Yes 84.2%
Do you think that the time the robot needs to learn was appropriate?	Yes 89.5%	Yes 63.2%	Yes 73.7%
Did technical conspicuousness occur during the interaction?	No 94.7%	No 61.1%	No 88.9%

Table 7.3: Results concerning the learning behaviour of the social scenarios.

Results of the robot's learning behaviour: For each of the three scenarios, the subjects had answer items within the questionnaire concerning the robot's learning behaviour. The results are shown in table 7.3. First of all, it can be noticed that most of the subjects were satisfied with the robot's learning behaviour in each scenario.

Furthermore, all participants stated that the robot had learned what they had tried to teach it during the *Greeting Scenario* and during the *Sorting Scenario*. For the *Hobby Scenario* still 84.2% stated that the robot had learned what they had tried to teach it. Because the *Greeting Scenario* and the *Hobby Scenario* were similar in so far as only one stimulus could be recognized by the robot and because the scenarios only differed insignificantly in the number of

available actions (7 for the *Greeting Scenario* and 8 for the *Hobby Scenario*), the 84.2% may result from the random component of the algorithm within the rational selection part.

The learning time that the robot needed was rather rated as being appropriate than as being inappropriate. However, the results reveal noticeable differences between the three scenarios concerning the learning time. Especially for the *Sorting Scenario* with its three recognizable stimuli, the number of participants who rated the learning time as being appropriate is comparatively low (63.2%) in contrast to the results of the *Greeting Scenario* (89.5%) and the *Hobby Scenario* (73.7%). One reason for this could be that the robot tried every action once for each stimulus before it started to decide based on the information stored in the emotional memory. Due to the fact that, within the *Sorting Scenario*, the robot was able to recognize three different stimuli and had five different actions available, in total $3 \cdot 5 = 15$ decisions were made without consideration of the obtained rewards. In contrast, there were only 7 actions to try in the *Greeting Scenario* and 8 actions to try in the *Hobby Scenario*.

For each scenario, the participants were able to state if technical issues had occurred during the interaction (e.g. the robot did not recognize a stimulus). While there were hardly any technical issues observed by the participants during the *Greeting Scenario* and *Hobby Scenario*, a noticeable number of technical issues was observed during the *Sorting Scenario*. This may result from problems with the visual recognition process, like disadvantageous illuminations. Additionally, the actions within the *Sorting Scenario* consisted of movements (the robot performed a pointing gesture) in contrast to the actions that were used in the *Greeting Scenario* and *Hobby Scenario*. Due to the movements, there was a higher chance for technical issues to occur, especially when the participants did not follow the interaction rules. In spite of the explanation on how to interact with the robot, some participants did not wait until the robot had finished its action before they gave a reward. In case of a negative reward this led to the robot stopping its movement and sometimes remaining in a disadvantageous position. This could have been avoided by the specification and usage of an appropriate *initposition*, which had not been considered in this study.

In addition to the items which had to be answered directly after each interaction scenario, the participants had to answer questions concerning the robot's learning process after all interactions had been finished. The results are shown in table 7.4. In contrast to the previously presented study every participant with only one exception has answered each question. Therefore, a comparison between both results is only partially meaningful as in the first study many items were left unanswered.

However, concerning the perception of the learning process every participant gave an answer. Only 52.4% stated that they had recognized the robot's learning progress during the playful interaction study presented in chapter 6. In contrast to that, nearly all participants (94.7%) were able to recognize the robot's learning progress in this study. Therefore, it can be assumed that the used interaction scenarios within this study are more suitable to show the robot's learning progress. In contrast to the previously presented playful interaction in which the robot was able to recognize 21 different stimuli, all social scenarios used less stimuli, namely one or three. This supports the conclusion presented in section 6.4 that the number of 15 rounds during the playful scenario was too low for this high number of stimuli. During the social scenarios, the participants were able to decide on their own how long the interaction lasted.

Although the learning progress was recognized by nearly all participants, the robot's learning progress as well as the robot's learning ability was rated as only being moderate. These results are comparable to the results of the playful scenario. Furthermore, the robot's way to learn was rated as being expedient in the playful scenario as well as in the social scenarios.

In view of the latter, most of the participants stated that they had used the different rewards' gradations but only 63.2% rated the rewards' gradations as being expedient. In contrast to that nearly all participants (95%) rated the rewards' gradations as being expedient in the previously presented study. One explanation for this effect might be the tasks itself. It can be assumed that the rewards' gradations are more expedient in the playful interaction. Within the playful interaction, there were decisions which were clearly advantageous, such as taking a further card when the accumulated card value is 12 or below. There were also decisions which were clearly disadvantageous or risky, such as taking a further card when the accumulated card value was very close to 21. In those situations it might be more expedient to give a reward with a maximum magnitude being either positive or negative. In contrast to that, there were situations in which the advantageous or disadvantageous character of a decision was less distinct. This might have led to rewards with smaller magnitudes being used.

A negative tendency can be observed concerning the question if the participants could imagine interacting with such a robot in their everyday life. Figure 7.6 shows the detailed results. Compared to the previously presented study (see figure 6.4), less participants could imagine to interact with the robot in their everyday life, although the robot's learning progress was evaluated as rather positive.

In summary, it can be noticed that most participants were satisfied with robot's learning abilities. The results of both studies show similarities as well

Item	Result
Did you recognize the robot's learning progress? (N=19)	Yes (94.7%)
I have perceived the learning progress as being... (N=19)	Moderate MEAN= 3.11 ± .69
How would you rate the robot's learning ability? (N=19)	Moderate MEAN= 2.63 ± .60
Do you think that the robot's way to learn is expedient? (N=19)	Expedient MEAN= 2.05 ± .52
Did you use the different rewards' gradations? (N=19)	Yes (84.2%)
Do you think that the rewards' gradations are expedient? (N=19)	Yes (63.2%)
I could imagine interacting with such a robot in my everyday life. (N=18)	Rather not MEAN= 2.61 ± .61

Table 7.4: Results of the participants' perception and acceptance of the robot's decision making behaviour (social scenario).

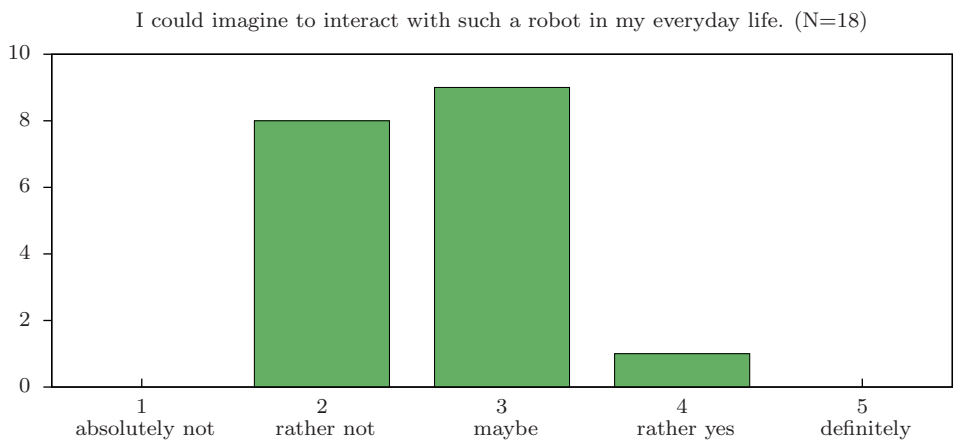

Figure 7.6: Results of the acceptance of the social interactions.

as differences, which can be explained by the different characteristics of the interaction scenarios. In the following, the most important findings, revealed by both human-robot interaction studies, are summarized.

Summarized results of both HRI-Studies: The results of both studies reveal important information in order to improve the decision making framework. While, during the playful scenario, only ≈ 50% of the subjects recognized that the robot is able to learn, nearly all subjects recognized the learning progress during the social interaction scenarios. One reason for that is the high number of stimuli within the playful interaction in contrast to the social interactions. It can be assumed that the 15 rounds during the game

study were not sufficient to make the robot's learning progress observable, as specific stimuli were rarely present. Furthermore, it led to confusion that the robot tried every action once before the choices were made based on the algorithm. Due to this implementation detail, it is possible that the robot's behaviour was perceived as less intelligent because the participant's feedback in some cases was ignored by the robot.

As an exemplary case, it could be considered that the robot recognized the stimulus *eighteen* for the first time and chose to hold its cards. In spite of receiving a positive reward from the participant, the second occurrence of the stimulus *eighteen* then led to the robot taking an additional card due to the characteristics of the algorithm. For subsequent studies the algorithm could be changed in so far, that the gathered feedback is used directly for decision making instead of trying every action once at the beginning.

Another reason for some decisions being perceived as less intelligent results from the random choice within the rational decision making part. This can lead the robot to perform undesired actions, especially when it has to relearn all somatic markers and the threshold can reach the minimum possible value. Consequently, it is possible that the robot chooses an action repeatedly although this action is rewarded negatively every time. Therefore, mechanisms for identifying such cases are needed in order to prevent the same negatively marked action from being chosen many times.

Apart from findings on possible enhancements of the algorithm, the perception of the reward possibilities was focused during the studies. The results show that most of the subjects consider the different levels of the rewards to be expedient. However, it is also observable that the task itself has an influence on the rating of the gradations. In view of this, it is of great interest to reveal how the subjects prefer to give rewards (e.g. by speech) which is a point of contact for subsequent works.

Furthermore, the results show different attitudes toward the acceptance of such a robot companion. One reason for the negative stances of some participants might be the fact that the used tasks were still too far away from the reality. Consequently, it is part of further research to investigate whether the acceptance increases, when the robot is able to handle more complex applications.

Chapter 8

Conclusion

In this chapter a summary of the whole thesis is given. Subsequently, benefits as well as drawbacks of the presented system are discussed. Based on this an outlook is given in order provide a point of contact for future research.

8.1 Summary

In this thesis a decision making framework for robot companions based on Damsio's somatic marker hypothesis has been presented. In the development of the framework different aspects playing a decisive role for robot companions have been considered, the focus thereby not only lying on the system (technical aspects) itself but on the user's perception of the system as well.

The framework fulfils different criteria which are essential in the creation of robot companion systems that find the user's acceptance. Furthermore, the framework is easy to use and thus may contribute to further research in the field of robot companions, which is of interest for many different disciplines like computer science or psychology. The presented decision making algorithm is applicable for variant tasks and does not require the adjustment of sensitive parameters.

Although the elaborations and findings of this thesis had a focus on computer science, the evaluation of the framework was extended with psychological studies. For evaluation purposes the IGT was used. The results have shown that the algorithm is able to solve this task successfully and that the decisions made by the robot companion system are comparable to those made by humans solving the IGT. These results have been confirmed by an additional study conducted in the scope of this thesis: having been asked to assign made decisions to their sources, the participants were not able

to distinguish between decisions made by humans from those made by the algorithm.

Based on these results the algorithm was implemented on the humanoid NAO robot in order make investigations concerning HRI aspects. The algorithm was implemented with the aim of giving even non-expert users the possibility of enhancing the robot's capabilities. This was realized with different interfaces that can be used in dependency of the user's expertise. A tool enabling the end user to create new applications without any programming skills being necessary was developed. In order to evaluate the tool, a usability study with participants of the age group 40 or older was conducted. Although the tool offers many mechanisms to assist the user, the results have shown that the users still had difficulties with solving the given task.

Finally, the decision making framework was evaluated concerning HRI aspects. For this purpose, two different studies in which participants interacted with the NAO robot were conducted. For both studies participants of the age group 40 or older had been recruited. In the first study it was the participants' duty to teach the robot how to play the game 17+4. The second study focused on daily interactions, for instance the participants had to teach the robot to assign certain kinds of objects to appropriate depositories. Overall, the results have revealed that the participants were rather satisfied with the robot's learning capabilities. However, it could be observed that the given scenario had an influence on the results. Based on these findings it is possible to derive certain ways of improving the framework.

Figure 8.1 gives an overview of the whole thesis. The arrows with continuous lines show all elements which have been realized in the scope of this thesis while the remaining arrows represent points of contact for subsequent works. In section 1.1, it has been discussed that there is often a gap between the implementation of a decision making algorithm on a real robot and respective studies concerning HRI aspects. Due to that reason, the feedback gathered from HRI studies is hardly usable in order to improve decision making algorithms for robot companions. The presented framework closes this gap and allows for using the gathered information from the studies in order to derive improvements for the used decision making algorithm. Furthermore, the framework also considers that non-expert users are able to enhance the robot's capabilities which has often been unattended in spite of being an important factor in the creation of robot companions.

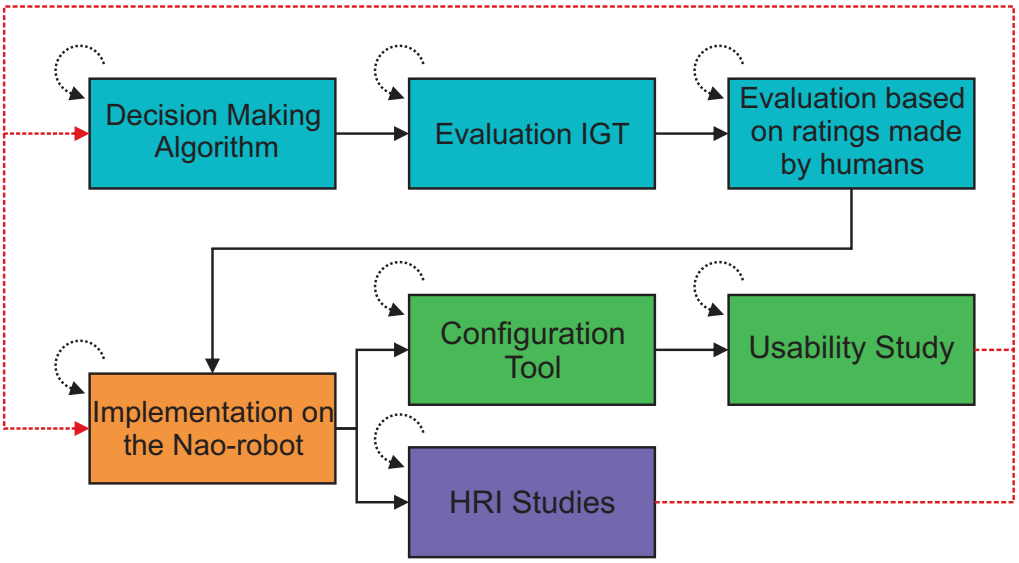

Figure 8.1: Overview of the whole thesis. There is now a direct connection between the implementation of the decision making algorithm and subsequent studies concerning human-robot interaction aspects. Each step itself leaves open research questions and would profit from subsequent research (black dotted arrows).What is most important is that the feedback gathered from user studies could be used directly to implement improvements (red dashed arrows).

8.2 Discussion and Outlook

Since the focus of this thesis lies on the completeness of the decision making framework, some aspects have temporarily been ignored. As already indicated in figure 8.1, each step has the potential for being enhanced. The immense number of further research questions makes it impossible to discuss every approach the developed framework still holds in store for new scientific findings. In order to give impulses for subsequent research, however, some selected aspects are presented in the following.

One important aspect of the decision making algorithm is, that most of the parameters are adapted automatically. There is only a single user-given parameter which does not have a huge impact on the results. In chapter 2 examples and justifications for each adaptation method are presented. A point of contact for subsequent research could be the implementation of different adaptation mechanisms in order compare the results to the current approach. Furthermore, it would be of great interest to evaluate the algorithm with other tasks than the IGT. Based on the results of the additional tasks, studies

like those presented in chapter 3 can be conducted, in order to compare the agent's decisions to those made by human subjects.

In addition, mechanisms allowing the generalization of gathered knowledge are expedient with respect to the efficiency of the decision making process even in unknown situations. Therefore, it is advisable that stimuli do not have to be defined by the user but are created by the system itself from information gathered from the environment. In combination, this would allow the agent to apply acquired knowledge to unknown stimuli, when a comparable stimulus has already been recognized. The card game 17+4, which is presented in chapter 6, is a prime example for an application potentially benefiting from the generalization of knowledge. In the scenario presented this thesis 21 different stimuli were used, one for each card value. If the agent was able to generalize its knowledge, it would be possible to reduce the number of stimuli from 21 to 2. One stimulus, then, would merge all card values at which it is advisable to take a further card, while the other stimulus was to merge all card values at which it is advisable to hold the cards. For now, the decision making algorithm only considers that appropriate actions for one specific stimulus can be learned. There are no possibilities of learning action sequences in order to solve a more complex task, which would be an important extension. All the previously listed examples demonstrate how many problems need to be solved in subsequent works just with regard to the decision making algorithm.

Concerning the implementation on the NAO robot presented in chapter 4, it can be suggested that subsequent works focus on data fusion mechanisms. For now, only one action can be executed at the same time. An expedient extension should make it possible to execute different actions in parallel as long as they do not use the same resources of the robot. An example is the execution of an action that consists of a movement and an action that consists of speech at the same time.

In order to give even non-expert users the possibility of customizing their robot's capabilities, a tool presented in chapter 5 was developed. The results of the usability study have revealed that the users still had some serious difficulties during the configuration of an application. The missing understanding of the learning concept has been identified as a main reason for the difficulties. A revision of the tool is necessary and should focus on assistance for the user. However, the tool is also helpful for experienced users in order to create new applications.

Finally, the decision making framework was evaluated concerning HRI aspects. The results have given important evidence on possible improvements of the decision making algorithm. This leads to the necessity of conducting further studies in which a control group interacts with a robot that uses the

current implementation of the algorithm, while the other group interacts with a robot on which an improved version of the decision making framework has been implemented. This would reveal if the derived enhancements lead to better results.

Needless to say that each step of the framework presented in this thesis needs to be examined further. Especially the conducted studies concerning the perception of decisions made by the algorithm, the usability of the configuration tool and human-robot interaction are a point of contact for subsequent works. However, all studies presented in this thesis have indicated, at least, that the next step in the developing process can be started.

In contrast to many other works which often focus on only one of the discussed topics, many issues have been covered in this thesis. Due to this, some of the previously listed aspects could not be fully elaborated here and thus are part of further investigations. However, the presented approach can be used as a basis for subsequent research, be it focused on decision making algorithms, usability or human-robot interaction.

Appendix A

Statistical Methods

For the experiments shown in the chapters 3, 6 and 7 several statistical methods were used to analyze the gathered data. This chapter provides further explanations of the used methods for the purpose of helping the reader to interpret the results. Detailed information on the mathematical approaches and requirements for specific kinds of statistical tests can be found in the referenced literature. All information presented in this chapter is based on the books Coolidge (2006); Howell (2007); Field (2009); Dugard et al. (2010).

A.1 Terms

In the following sections the statistical tests used in this thesis are briefly described. The most important terms needed for the explanations of procedures and results are listed below. All definitions are quoted from Coolidge (2006):

- **Alternative hypothesis** (H_a) - Most frequently, what the experimenter thinks may be true or wishes to be true before he or she begins an experiment: also called the research hypothesis. It can also be considered the experimenter's hunch (Coolidge, 2006, p. 147).
- **ANOVA repeated-measures design** - Similar to the dependent t test design in that the same participants are used for each level of the independent variable. The research interest is whether the means for each level of independent variable are significantly different from each other (Coolidge, 2006, p. 279).
- **Chi-square (χ^2) test** - One of the most popular nonparametric tests that involves the assessment of one or more

independent variables, each with two or more levels of nominal or categorical data (Coolidge, 2006, p. 350).

- **Degrees of freedom** - A parameter that is equal to the number of observations or groups in a study minus some value(s) that limit the observations' or groups' freedom to vary (Coolidge, 2006, p. 190).
- **Dependent t test** - A test designed to determine the statistical difference between two means where the participants in each group are either the same or matched pairs (Coolidge, 2006, p. 235).
- **Effect size** - How strongly the independent variable affects the dependent variable. Effect size ranges from small to large (Coolidge, 2006, p. 214).
- **Factor analysis** - A statistical procedure that explores the underlying conceptual structure of a set of dependent variables by examining the correlation between each variable in the set with every other variable in the set. Factor analysis can also be used to reduce the set of variables by identifying redundant or unnecessary items (Coolidge, 2006, p. 370).
- **Null hypothesis** (H_0) - The starting point in scientific research where the experimenter assumes there is no effect of the treatment or no relationship between two variables (Coolidge, 2006, p. 148).
- **p level** - The probability of committing the Type I error; that is rejecting H_0 when H_0 is true (Coolidge, 2006, p. 148).
- **Repeated-measures t test** - A dependent t test design where the participants are the same in both groups and usually are measured pretreatment and posttreatment (Coolidge, 2006, p. 235).
- **Significance** - Findings are considered statistically significant if the probability that we are wrong (where we reject H_0 and H_0 is true) is less than .05. Significant findings indicate that the results of the experiment are substantial and not due to chance (Coolidge, 2006, p. 148).
- **Type I error** - When an experimenter incorrectly rejects the null hypothesis when it is true (Coolidge, 2006, p. 148).

	Items
1	comprehensible - incomprehensible
2	complex - simple
3	human - machine-like
4	predictable - unpredictable
5	artificial - natural
6	random - deliberate
7	programmed - spontaneous
8	familiar - unfamiliar

Table A.1: Item-pairs which were used to measure predictability and naturalness (7-point semantic differential).

Item	1	2	3	4	5	6	7	8
1	1.000	-.326	.141	.495	-.271	-.362	.196	.524
2	-.326	1.000	-.133	-.438	.165	.348	-.153	-.291
3	.141	-.133	1.000	.048	-.707	.096	-.499	.417
4	.495	-.438	.048	1.000	-.135	-.318	.200	.348
5	-.271	.165	-.707	-.135	1.000	-.118	.385	-.363
6	-.362	.348	.096	-.318	-.118	1.000	-.441	-.290
7	.196	-.153	-.499	.200	.385	-.441	1.000	-.005
8	.524	-.291	.417	.348	-.363	-.290	-.005	1.000

Table A.2: Correlation matrix which results from the analysis in chapter 3.

- **Within-subjects variable** - A factor where the same group of participants is measured on the dependent variable at every level of the independent variable (Coolidge, 2006, p. 330).

A.2 Factor Analysis

The factor analysis is performed to identify significant correlations of variables. Based on this information, it is possible to reduce the dimensions. All variables which are highly correlated load on the same factor (latent variable). In this thesis an exploratory factor analysis has been used which is based on the Principal Component Analysis (PCA) with varimax rotation (Field, 2009, p. 628-672). The first step is the computation of the correlation matrix in order to get information about the correlation between items. The questionnaire in chapter 3 included the items shown in table A.1. Table A.2 shows the resultant correlation matrix based on the gathered information.

Based on the correlation matrix the eigenvalues of the matrix are calculated in order to determine the eigenvectors (factors). The result of this computation are 8 eigenvectors (one for each item). Only eigenvectors with an eigenvalue being greater than 1 are considered (Kaiser's criterion). The eigenvalues are shown in table A.3. Two factors with an eigenvalue greater than 1 are

	Eigenvalues without rotation			Eigenvalues with rotation		
Component	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
C1	**2.756**	34,456	34,456	**2.680**	33,506	33,506
C2	**2.236**	27,945	62,400	**2.312**	28,894	62,400
C3	.759	9,494	71,894			
C4	.675	8,440	80,334			
C5	.480	5,997	86,331			
C6	.453	5,660	91,992			
C7	.403	5,040	97,032			
C8	.237	2,968	100,000			

Table A.3: Eigenvalues computed with the PCA without varimax rotation and with varimax rotation.

extracted. The first factor C1 explains 34,456% of the variance while the second factor C2 explains 27,945% of the variance.

Subsequent to the extraction of the factors, it is possible to determine the degree to which the variables load on these factors. The varimax rotation is used in order to assist the interpretation of the factor loadings by rotating the factor axes which results in each variable having a high loading on only one of the factors. The rotation also influences the eigenvectors and eigenvalues (see table A.3). In this specific case, the rotation does not have a huge impact, as the eigenvalues of the components already have a similar magnitude and therefore explain nearly a similar percentage of the variance. Finally, the rotated component matrix including the factor loadings (high magnitudes denote high loadings), which is shown in chapter 3, is computed (see table 3.3). In table 3.3 values with magnitudes smaller than 0.45 are suppressed, as they are not significant. Table A.4 shows the factor loadings in all detail while figure A.1 shows a graphical representation of them. It can be observed that the items 1, 2, 4, 6 and 8 load highly onto the first factor labeled as predictability while the items 3, 5 and 7 load highly onto the second factor labeled as naturalness.

Subsequent to the factor analysis, the reliabilities of the resultant subscales (predictability and naturalness) were tested. For this purposes Cronbach's α, which gives information about the internal consistency, was computed for each subscale (Field, 2009, p. 673-681). Only values greater than or equal to zero give information whereupon higher values denote a higher reliability. The maximum value for α is 1. For the first subscale Cronbach's α is .749 while, for the second subscale, Cronbach's α is .773. Based on these values, both scales are reliable.

		Factors with labels	
		predictability (Component 1)	naturalness (Component 2)
1	comprehensible - incomprehensible	**.773**	.131
2	complex - simple	**-.658**	-.050
3	human - machine-like	.138	**.882**
4	predictable - unpredictable	**.732**	-.004
5	artificial - natural	-.218	**-.836**
6	random - deliberate	**-.674**	.340
7	programmed - spontaneous	.401	**-.728**
8	familiar - unfamiliar	**.658**	.410
	Cronbach's α	.749	.773

Table A.4: Rotated component matrix of the factor analysis. Bold numbers denote high magnitudes.

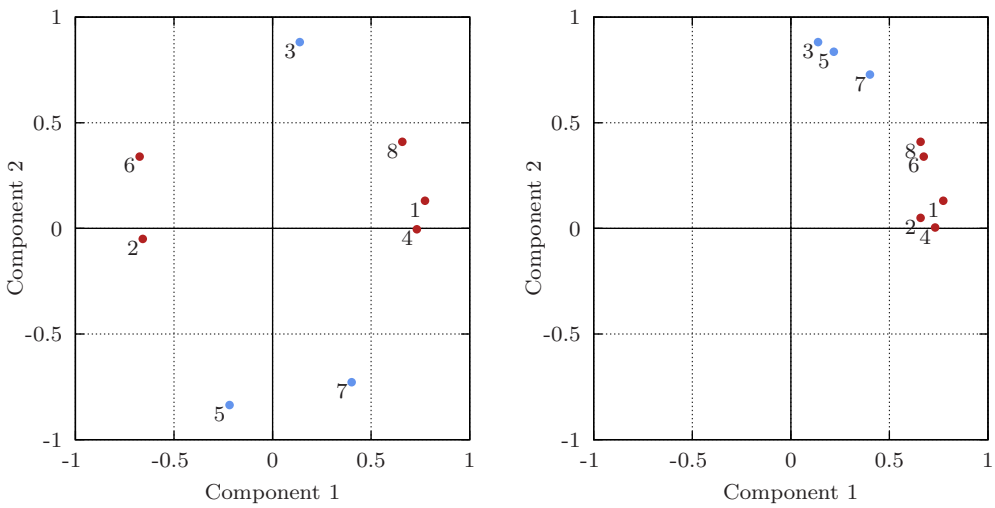

Figure A.1: On the left hand side this figure shows the factor loadings from table A.4. Another representation which considers only the magnitudes of the values is shown on the right hand side and makes the grouping of the items more clear.

A.3 Analysis of variance (ANOVA)

The analysis of variance (ANOVA) is used in order to test whether three or more means are significantly different from each other (Field, 2009, p. 474-482). In chapter 3 an ANOVA (repeated-measures design) was conducted for each of the extracted factors (predictability, naturalness). The null hypothesis of this test is that there are no differences between the means. In general a significance level defining a point from which the null hypothesis is rejected

(usually 5% or 1%) is set for each test. In the case that the p-level of a test is smaller than the significance level, the null hypothesis is rejected. Here, this means that there is at least one mean value that differs significantly.

In the experiment (see chapter 3) three different types of outputs were considered (three means were compared). The results presented in chapter 3 are equivalent to those shown in eq. A.1. At first the degrees of freedom of the F-distribution are reported. The first degree of freedom df_1 is computed as shown in eq. (A.2) which is based on the different number of output types (treatments) τ. The computation of the second degree of freedom df_2 includes the number of participants (see eq. (A.3)).

$$F(\underbrace{2}_{df_1};\underbrace{40}_{df_2}) = \underbrace{55.05}_{\text{F-value}};\underbrace{p < .001}_{\text{p-level}};\underbrace{\eta^2 = .734}_{\text{effect size}} \tag{A.1}$$

$$df_1 = \tau - 1 = 2 \tag{A.2}$$

$$df_2 = (N - 1) \cdot (\tau - 1) = 40 \tag{A.3}$$

The F-value is the quotient of the model mean squares and the residual mean square. A value greater than 1 indicates that there is an effect. In order to test whether this effect is significant, the computed F-value is compared to a critical value which has been computed for a specific significance level based on the same degrees of freedom (critical values can be found in F-tables). In this case the significant level is set to 1% which means that the critical F-value is 5.18 (Field, 2009, p. 804-805). The computed F-value (55.05) has to be greater than or equal to the critical value to indicate that the effect is significant, which is the case here. Additionally, the effect size $\eta^2 \in [0, 1]$ is computed, a higher value thereby indicating a greater effect (Howell, 2007, p. 325-328).

Subsequent to the main analysis which only reveals that there is at least one mean that differs significantly, a pairwise comparison is used in order to identify which mean differs significantly from each other. Table A.5 shows the results of the pairwise comparison. It is observable that the mean of the agent's outputs does not differ significantly from the mean of the human outputs because the p-level lies above 5%. Furthermore, it can be seen that the mean of the agent's outputs differs significantly from the mean of the random outputs because the p-level is below 1%.

Mean(I)	Mean(J)	Mean difference (I-J)	Std. Error	p-level
Agent (-.2038)	Human (-.1250)	-.079	.100	.439
	Random (.9639)	-1.168	.134	.000
Human	Agent	-.079	.100	.439
	Random	-1.089	.136	.000
Random	Agent	-1.168	.134	.000
	Random	-1.089	.136	.000

Table A.5: Pairwise comparison for the factor (predictability).

Category	Observed N	Expected N	Residual
Agent	40	52.5	-12.5
Human	65	52.5	12.5

Table A.6: Frequencies of the categorization for outputs stemming from human players.

A.4 Chi-Square-Test

The χ^2 test is used to test categorical data. More precisely, it is tested whether an observed number differs significantly from what is expected (Coolidge, 2006, p. 336). The null hypothesis of this test is that the observed number does not differ from the expected number for a population. In the case that the p-level is smaller than 5%, the null hypothesis is rejected. In chapter 3 the graphical outputs of human, agent and random players had to be categorized as stemming either from a human player or from the artificial agent. For each of the output types a χ^2-test was conducted. The result of the test is reported as seen in eq. (A.4) (human outputs).

$$\chi^2(\underbrace{1}_{\text{df = categories - 1}}, N = 21) = \underbrace{5.95}_{\chi^2\text{-value}}, \underbrace{p < .05}_{\text{p-level}} \tag{A.4}$$

Table A.6 shows the frequencies of the chosen categorization for this type of output. For the computation of the χ^2-value the observed N, the expected N and the number of different categories are needed (Field, 2009, p. 688). In order to test whether this effect is significant, the computed χ^2-value is compared to a critical value which has been computed for a specific significance level based on the same degrees of freedom (critical values can be found in χ^2-tables). For a p-level of 5% this critical value is 3.84 (Field, 2009, p. 808). As the computed $\chi^2 = 5.95$ is greater than the critical value, the null hypothesis is rejected and it can be concluded that this output type was categorized as stemming from a human player more often.

A.5 t-Test

The t-test is used to compare two means in order to test whether one mean differs significantly from the other. In chapter 7 a repeated measures t-test was used in order to test whether the interaction with the robot had a significant influence on the subjects' anxieties and attitudes toward robots. The results are presented in the form shown in eq. A.5.

$$t(\underbrace{18}_{\text{df}=N-1}) = \underbrace{2.13}_{\text{t-value}}, \underbrace{p < .05}_{p-level} \tag{A.5}$$

Due to the computation, the t-value becomes positive when the mean of the post-treatment is smaller than the mean of the pre-treatment and negative in the reverse case (Coolidge, 2006, p. 224). After the computation, the t-value is compared to a critical t-value in order to determine whether the difference is significant. In this case the t-value (2.13) exceeds the critical value 2.10 at $p = .05$ (see t-table in (Field, 2009, p. 803)). Consequently, there is a significant difference between the mean of the pre-treatment (2.74 ± 0.75) and the mean of the post-treatment (2.28 ± 0.94).

Bibliography

Bark, R., Dieckmann, S., Bogerts, B., and Northoff, G. (2005). Deficit in decision making in catatonic schizophrenia: an exploratory study. *Psychiatry Research*, 134(2):131–141.

Bechara, A., Damasio, A., Damasio, H., and Anderson, S. (1994). Insensitivity to future consequences following damage to human prefrontal cortex. *Cognition*, 50:7–15.

Bechara, A. and Damasio, A. R. (2005). The somatic marker hypothesis: A neural theory of economic decision. *Games and Economic Behavior*, 52(2):336–372.

Bechara, A., Damasio, H., and Damasio, A. R. (2000a). Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex*, 10(3):295–307.

Bechara, A., Damasio, H., Tranel, D., and Damasio, A. R. (2005). The iowa gambling task and the somatic marker hypothesis: some questions and answers. *Trends in Cognitive Sciences*, 9(4):159–162.

Bechara, A., Tranel, D., and Damasio, H. (2000b). Characterization of the decision-making deficit of patients with ventromedial prefrontal cortex lesions. *Brain*, 123(11):2189–2202.

Brand, M., Grabenhorst, F., Starcke, K., Vandekerckhove, M. M., and Markowitsch, H. J. (2007). Role of the amygdala in decisions under ambiguity and decisions under risk: evidence from patients with urbach-wiethe disease. *Neuropsychologia*, 45(6):1305–1317.

Brand, M., Labudda, K., and Markowitsch, H. J. (2006). Neuropsychological correlates of decision-making in ambiguous and risky situations. *Neural Networks*, 19(8):1266–1276.

Breazeal, C. (1999). Robot in society: friend or appliance. In *Proceedings of the 1999 Autonomous Agents Workshop on Emotion-Based Agent Architectures*, pages 18–26.

Breazeal, C. (2003). Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, 59(1):119–155.

Burghouts, G. J., Heylen, D., Poel, M., op den Akker, R., and Nijholt, A. (2003). An action selection architecture for an emotional agent. In *FLAIRS Conference*, pages 293–297.

Coolidge, F. (2006). *Statistics: A Gentle Introduction.* Sage Publications.

Damasio, A. (1994). *Descartes Error: Emotion, Reason, and the Human Brain.* Putnam Press.

Dautenhahn, K., Woods, S., Kaouri, C., Walters, M. L., Koay, K. L., and Werry, I. (2005). What is a robot companion - friend, assistant or butler. In *In Proc. IEEE IROS*, pages 1488–1493.

Doya, K. (2000). Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, 10(6):732–739.

Dugard, P., Todman, J., and Staines, H. (2010). *Approaching Multivariate Analysis: A Practical Introduction.* Routledge, second edition.

Dunn, B. D., Dalgleish, T., and Lawrence, A. D. (2006). The somatic marker hypothesis: A critical evaluation. *Neuroscience & Biobehavioral Reviews*, 30(2):239–271.

Fellows, L. K. and Farah, M. J. (2005). Different underlying impairments in decision-making following ventromedial and dorsolateral frontal lobe damage in humans. *Cerebral Cortex*, pages 58–63.

Field, A. (2009). *Discovering Statistics Using SPSS: (and Sex, Drugs and Rock'n'roll).* ISM (London, England). SAGE.

Fishbein, D., Hyde, C., Eldreth, D., London, E. D., Matochik, J., Ernst, M., Isenberg, N., Steckley, S., Schech, B., and Kimes, A. (2005). Cognitive performance and autonomic reactivity in abstinent drug abusers and nonusers. *Experimental and Clinical Psychopharmacology*, 13(1):25.

Harrington, K. I., Olsen, M. M., and Siegelmann, H. T. (2011). Communicated somatic markers benefit both the individual and the species. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 3272–3278. IEEE.

Hoefinghoff, J., Hoffmann, L., Krämer, N., and Pauli, J. (2013a). Deciding like humans do. In *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference*, pages 557–562. AAAI Press.

Hoefinghoff, J. and Pauli, J. (2012). Decision making based on somatic markers. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pages 163–168. AAAI Press.

Hoefinghoff, J. and Pauli, J. (2013). Reversal learning based on somatic markers. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 498–504.

Hoefinghoff, J., Steinert, L., and Pauli, J. (2012). Implementation of a decision making algorithm based on somatic markers on the nao robot. In *Autonomous Mobile Systems 2012*, Informatik Aktuell, pages 69–77. Springer.

Hoefinghoff, J., Steinert, L., and Pauli, J. (2013b). An easily adaptable decision making framework based on somatic markers on the nao-robot. *Kognitive Systeme*, 1.

Hoogendoorn, M., Merk, R., Roessingh, J., and Treur, J. (2009). Modelling a fighter pilot's intuition in decision making on the basis of damasio's somatic marker hypothesis. In *Proc. of the 17th Congress of the Int. Ergonomics Association*. CD-Rom.

Howell, D. (2007). *Statistical Methods for Psychology*. Thomson Wadsworth, sixth edition.

James, W. (1884). What is an emotion? *Mind*, 9:188–205.

Ko, C.-H., Hsiao, S., Liu, G.-C., Yen, J.-Y., Yang, M.-J., and Yen, C.-F. (2010). The characteristics of decision making, potential to take risks, and personality of college students with internet addiction. *Psychiatry research*, 175(1):121–125.

Kovalchik, S. and Allman, J. (2006). Measuring reversal learning: Introducing the variable iowa gambling task in a study of young and old normals. *Cognition and Emotion*, 20:714–728.

Kramer, J. and Scheutz, M. (2007). Development environments for autonomous mobile robots: A survey. *Autonomous Robots*, 22(2):101–132.

Laing, G. (2004). *Digital Retro: the Evolution and Design of the Personal Computer*. Sybex.

Mata, C. D. and Aylett, R. (2005). Having it both ways–the impact of fear on eating and fleeing in virtual flocking animals. In *Modelling Natural Action Selection: Proceedings of an International Workshop, Edinburgh, Scotland*, pages 152–157.

Mori, M. (1970). The uncanny valley. *Energy*, 7(4):33–35.

Noël, X., Brevers, D., and Bechara, A. (2013). A neurocognitive approach to understanding the neurobiology of addiction. *Current Opinion in Neurobiology*, 23(4):632–638.

Nomura, T., Suzuki, T., Kanda, T., and Kato, K. (2006a). Measurement of anxiety toward robots. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 372–377. IEEE.

Nomura, T., Suzuki, T., Kanda, T., and Kato, K. (2006b). Measurement of negative attitudes toward robots. *Interaction Studies*, 7(3):437–454.

Picard, R. W. (1995). Affective computing.

Pimentel, C. and Cravo, M. (2009). Don't think too much! - Artificial somatic markers for action selection. In *Int. Conf. on Affective Computing and Intelligent Interaction*, volume 1, pages 55–62, Amsterdam. IEEE.

Poel, M., op den Akker, H., Nijholt, A., and van Kesteren, A.-J. (2002). Learning emotions in virtual environments. In Trappl, R., editor, *Cybernetics and Systems 2002*, pages 751–755, Vienna. Austrian Society for Cybernetic Studies.

Salichs, M. A. and Malfaz, M. (2012). A new approach to modeling emotions and their use on a decision-making system for artificial agents. *Affective Computing, IEEE Transactions on*, 3(1):56–68.

Stolper, E., Van de Wiel, M., Van Royen, P., Van Bokhoven, M., Van der Weijden, T., and Dinant, G. J. (2011). Gut feelings as a third track in general practitioners' diagnostic reasoning. *Journal of General Internal Medicine*, 26(2):197–203.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Thomaz, A. L., Berlin, M., and Breazeal, C. (2005). An embodied computational model of social referencing. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 591–598. IEEE.

Tsankova, D. D. (2009). Emotional intervention on an action selection mechanism based on artificial immune networks for navigation of autonomous agents. *Adaptive Behavior*, 17(2):135–152.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, pages 433–460.

Velásquez, J. D. (1998). When robots weep: emotional memories and decision-making. In *AAAI/IAAI*, pages 70–75.

Vitay, J. and Hamker, F. (2011). A neuroscientific view on the role of emotions in behaving cognitive agents. *KI - Künstliche Intelligenz*, 25:235–244.

Wada, K., Shibata, T., Saito, T., Sakamoto, K., and Tanie, K. (2005). Psychological and social effects of one year robot assisted activity on elderly people at a health service facility for the aged. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2785–2790. IEEE.

Watkins, C. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.

Watson, C., Kirkcaldie, M., and Paxinos, G. (2010). *The Brain: An Introduction to Functional Neuroanatomy*. Elsevier Science.

Wendemuth, A. and Biundo, S. (2012). A companion technology for cognitive technical systems. In *Cognitive Behavioural Systems*, volume 7403 of *Lecture Notes in Computer Science*, pages 89–103. Springer.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342.

Zuse, K. (1993). *The Computer - My Life*. Springer.

Nomenclature

α	Cronbach's alpha (coefficient of internal consistency)
χ^2	Chi-square test
Δ	Weighting value for the exponential smoothing
η^2	Effect size
$\hat{w}$	Weighting of collected knowledge
κ_i^t	Reliability of collected knowledge for stimulus s_i
λ	Quadratic function which is used for the computation of the weightings w and $\hat{w}$
$\mathbb{N}^+$	Natural numbers without zero
$\mathbb{Z}$	Integers
$\max(R_{s_i})$	Maximal possible reward
$\min(R_{s_i})$	Minimal possible reward
μ	Quadratic function which is used for the computation of the weightings w and $\hat{w}$
$\overline{r_{i,j}^t}$	Current accumulated reward
$\overline{r_{i,j}^{\text{init}}}$	Initial value for the last accumulated reward
$\overline{r_{i,j}^{t-1}}$	Last accumulated reward
$\sigma_{i,j}^t$	Somatic marker for stimulus s_i and action a_j
τ	Number of treatments
θ_i^t	Frustration level (threshold) for stimulus s_i

$\vec{\kappa}$	Vector that contains a reliability value for each stimulus
$\vec{\theta}$	Vector that contains a frustration level value for each stimulus
$\widehat{\kappa_i^{t+1}}$	Temporary reliability value during the computation
$\widetilde{r_{i,j}^t}$	Scaled current reward
A	Set of executable actions
A'	Resultant subset of actions after the emotional selection part
a_j	Action that could be executed
c	Reward resolution
F	F-value of variance analysis
i	Running index for stimuli
j	Running index for actions
$M_{\lvert S\rvert\times\lvert A\rvert}$	Matrix that contains all somatic markers
m	Number of actions
M_i	Vector which contains all somatic markers for stimulus s_i
N	Number of subjects or artificial agents within an experiment
n	Number of stimuli
o	Number of rewards
p	Probability of committing the Type I error
$r_{i,j}^t$	Current reward
r_{max_i}	Reward with the maximum magnitude
R_{s_i}	Set of possible rewards for a specific stimulus
S	Set of recognizable stimuli
s_i	Stimulus that could be recognized
t	Discrete time index
w	Weighting of new knowledge

List of Figures

List of Tables