Explore Explainable AI with QT

Sunday 5th January, 2020 - 13:37

Iris Kremer
University of Luxembourg
Email: iris.kremer.001@student.uni.lu

Amro Najjar
University of Luxembourg
Email: amro.najjar@uni.lu

Diego Agustin Ambrossio
University of Luxembourg
Email: agustin.ambrossio@uni.lu

***Abstract*—This paper presents the bachelor semester project made by Iris Kremer under the direction of her project academic tutor Dr. Amro Najjar and with the help of Dr. Agustin Ambrossio. This project seeks to use a robot as emotion sensitive music instrument and this way to explore the domain of explainable AI (XAI). The scientific deliverable produced during this project is an analysis of the role of emotions in the domains of human-robot interaction (HRI) and XAI, and the technical deliverable is a program that changes the music played with a robot according to the facial expression of the user. Therefore, the scientific dimension of this project focusses on XAI and on the role of emotions in robotics, while the technical aspect concerns facial emotion recognition and the robot operating system (ROS).**

1. Introduction

Recently, researchers in the domain of robotics have started to address questions about using robots in art. This idea is quite counter-intuitive, since art is an abstract concept that requires expressivity and sensibility to be produced, while robots, being machines after all, are currently known to be devoid of these capabilities, even though this is still under discussion [1]. However, the concept of using robots in art definitely has potential with the rapid advancements made in artificial intelligence or AI. A very recent and famous art piece produced by an AI is the portrait painting *"Edmond de Belamy"* featured in Figure 1. It was produced by a generative adversarial network, short GAN, developed by the French arts-collective *Obvious* [2]. We could think of implementing such technology in a robot to give it the capacity to produce art too.

Figure 1. *Edmond de Belamy* (2018) [3]

The technical improvement of robots has also broadened the possibilities of robots making music, especially with those who were developed for it. There now exist special robots playing all kinds of music instruments (percussive, string and wind) like the piano-playing hand from the University of Washington, the violin-playing robot from Ryukoku University and the Flutist from Waseda University. These and further robot musicians are presented in [4] alongside with an analysis on capabilities of robots to have what they call "musical intelligence", i.e. the expressivity, sensibility and further qualities required to produce music, and how those are implemented in robot musicians.

Staying in the domain of music, we also have robots developed to teach music. According to [5], robots are engaging music assistant teachers for children, because the children see the robot as their friend rather than their teacher, which motivates them to work with it. Robots are also efficient music teachers for children with autism, as demonstrated by the study [6].

However, until now, no robot was ever used as music instrument to play with. This semester project works with this goal in mind: using a robot as musical instrument. Furthermore, the general idea is to use the music instrument robot as tool to teach music to children, since robots are very engaging for them to learn.

Another bachelor semester project[1] works on program-

1. "Human-Robot Interaction", by Eliott Bonte

ming the robot QT such that a user can play music with it in the same way than with a theremin.
While having QT working as a music instrument is a significant contribution, a necessary and useful improvement is to endow QT with emotion recognition capacities. In art and especially in music, emotions undeniably play an essential role, because artists communicate emotions through their art. Emotions are also of primary importance in Human-Robot Interaction or HRI [7]; The ability of a robot to show sensibility to human emotion enhances a lot the communication with humans. In the context of a teaching robot for children, this factor becomes even more important, because, as has been shown in recent works, children tend to develop an emotional bond with the robot [5].

Therefore, this bachelor semester project has as goal to adapt the music played using QTrobot to the facial emotion of the musician playing.

The detailed project description, including domains and targeted deliverables, can be found in Section 2.1 of this paper, while the pre-requisites of the project are described in Section 3. The next sections then present the concrete deliverables of the project.

The goal of the project first requires the robot to be able to recognize human emotion on the players face. Furthermore, we must be able to explain QTs behaviour according to the emotion recognized, i.e. the changes in the music. Therefore, we will explore the domain of explainable AI, short XAI, and analyse the concrete impacts of the emotion factor in XAI, and inevitably also in HRI. The result of this research is presented in Section 4 and is the scientific deliverable of the project.

As for the technical deliverable of the project, a program has been produced during this semester project which fulfils the goal of the project to make an emotion sensitive instrument out of QTrobot. This program is presented in section 5 of this paper.

Finally, we conclude on the project and present future perspectives in Section 6.

2. Project description

2.1. Domains

2.1.1. Scientific.

This project addresses two scientific domains: explainable AI and human-robot interaction. These two domains are presented briefly in the following subsections.

Explainable AI

EXplainable AI, short XAI, is a set of techniques that enable humans to explain solutions given by an artificial intelligence (AI) to given problems [8]. These techniques aim to understand how decisions are made by so-called “black box” systems, like neural networks for instance [9]. They can rely on humans examining the AI system to explain its functioning, or on the AI itself explaining its behaviour to humans, for example through decision models.

Explainable AI is needed, because AI systems are becoming increasingly complex. Some are created through random weighting of artificial neurons like deep neural networks, and some are even capable of learning by themselves. But still, AIs could make mistakes. How can someone assess that an AI is reliable without understanding how it makes decisions? The same question holds for correcting and improving AI systems. With XAI, humans are able to understand and explain these systems and therefore also prove or disprove their reliability.

Even though it may seem concise at first glance, XAI is a broad field. In this project, we will focus on explainable AI in the context of robots [10], underlying the importance of understanding a robot’s behaviour and intentions, which of course can be achieved through XAI, and analysing how emotions can contribute to this field.

Human-Robot Interaction

Human-Robot Interaction, short HRI, is the field studying interactions and communication between robots and humans. It is a subcategory of Human-Computer Interaction.

Communication between human and robot can be proximate, which means that the human and the robot are in the same room, or remote, i.e. the human and robot communicate from a larger distance [11].

HRI is a very large field, so it would need several bachelor semester projects to cover it entirely. Therefore, this project focuses particularly on the importance of the emotion factor in HRI and so-called affective robots.

2.1.2. Technical.

The project also concerns three technical domains, which are facial emotion recognition, QTrobot and robot operating system. These domains are presented in the following subsections.

Facial Emotion Recognition

Facial emotion recognition is the process of identifying emotions on human faces. Humans perform facial emotion recognition on a daily basis and artificial intelligence can be trained to do the same.

Training an AI to recognize emotion on facial expressions involves machine learning procedures. The AI is trained to recognize features of each expression using large data sets with pictures and videos of human faces showing different emotions. The AI will parse this data and extract features characterizing different emotions on human faces to be able to recognize them. Then, the AI is tested on another set of data and the cycle may repeat until the

AI is well trained.

In this project, we make use of existing facial emotion recognition algorithms provided by the framework openCV, to enable the robot to recognize the emotions of the user in front of it through its front camera. This domain will also partly be addressed in the scientific part of the project concerning XAI, but has its main importance in the technical part.

QTrobot

QTrobot, short QT, is a robot developed by the company LuxAI. This robot's main purposes are first to teach autistic children basic human interaction through its children friendly interface and second to provide a good quality research tool for human-robot interaction.

Figure 2. QT assisting a child with autism. [12]

QT is the robot used in the technical part of this project. Therefore, the whole technical deliverable will revolve around it.

Robot Operating System

Robot operating system, short ROS [13], despite its name is not an operating system, but an open-source middleware for robots. It has a specific architecture, the so-called publisher and subscriber architecture, which is quite convenient for most robots. Therefore, this middleware is used in many robots.

ROS is a domain of this project because QT was built and developed on it, which means we inevitably had to use it and therefore needed to understand its architecture quite well. An explanation of the ROS architecture will be provided later in this paper, in Section 5.2.

2.2. Targeted Deliverables

2.2.1. Scientific Deliverable.

For the scientific part, the aim of this project is to explore the domain of explainable AI (and inevitably HRI) with focus on the role of emotion and QT as study case. Therefore, the scientific deliverable consists of an analysis of the emotion factor in the domains of HRI and XAI, especially how it improves communication between human and robots and enhances the capability of robots to explain their behaviour to humans.

This analysis will approach the topic both from a psychological and a technical point of view, which means that both social impacts and concrete implementation aspects of emotion in robotics will be discussed.

2.2.2. Technical Deliverable.

The general idea of the project is to use QT as a musical instrument that adapts the music played to the emotion of the player. Therefore, the technical deliverable of this project is a program for QT that analyses the emotion of the user in front of it through its front camera, takes the melody he or she plays and modifies it according to the emotion detected.

2.3. Constraints

This project required a robot to work with for the technical deliverable. We chose to use QT, because it has already some built-in features like facial and emotion recognition, which are very useful in this project.

Because of this choice, we were imposed to work with ROS, as already explained in section 2.1. This also meant that we could only use either C++, or Python as programming language to code the technical deliverable. We chose the latter.

Furthermore, we also needed to get familiar with working with QT in general, as in Figure 3, and especially understand QT's architecture. QT has two distinct, but connected computers, one in its head and one it its belly. This fact did not constraint the design of the code itself, but it complicated a lot the execution of the technical deliverable and therefore also the testing of our program. Further details on this aspect are given in Section 5.3.

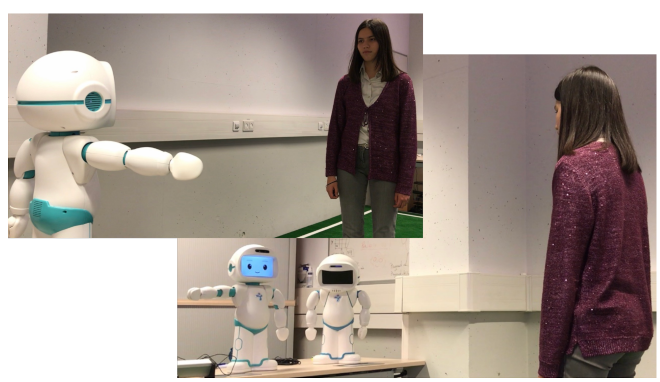

Figure 3. Interacting with QT.

3. Pre-requisites

3.1. Scientific Pre-requisites

This project does not have any scientific pre-requisite, except being able to read and analyse scientific documents, a know-how which was acquired during previous bachelor semester projects.

3.2. Technical Pre-requisites

3.2.1. Python.

The technical deliverable of this project is coded using the Python programming language. Therefore, basic Python knowledge is required to start this project. This knowledge was acquired before the start of the project.

3.2.2. Github.

The use of github in this project is not mandatory, but turns out to be extremely helpful to transfer code from our local machine onto QT and vice-versa. Therefore, we made use of the knowledge of Github we had beforehand.

3.2.3. Music Knowledge.

It is well known that various fields of computer science are overlapping on other fields. It is the case in this project, since aim of the technical deliverable is to use QT as music instrument that changes the music played according to the facial emotion of the user. Therefore, some basic knowledge of music is needed to figure out how to change the played music coherently with respect to the emotion of the player. Like the other prerequisites, this knowledge was already acquired before the start of the project.

4. Understanding XAI and the Emotion Factor in Robotics

4.1. Requirements

The goal of the scientific deliverable of this project is to understand the domain of explainable AI and the matters of the emotion factor in robotics. Furthermore, the knowledge acquired when answering these scientific questions will directly be applied in a case study that will analyse QT's behaviour according to the user's emotion when running the program developed as technical deliverable of this project (cf. Section 5). So, the scientific deliverable is divided into three parts.

4.1.1. Part 1 - Presentation of Explainable AI.

The goal of this part is not to provide an exhaustive overview of XAI, because this field is too broad. Instead, we only want to concentrate on explaining the importance of XAI for the domains of artificial intelligence and machine learning.

Note that a definition of XAI is already provided in Section 2.1 of this document, so there is no need to define XAI again in the scientific deliverable.

4.1.2. Part 2 - The Emotion Factor in Robotics, HRI and XAI.

In this project, we focus a lot on using emotions with a robot. However, we must also explain the interest of implementing emotions in AI systems and robots. We will provide explanations to this question with the following requirements:

1) Explain how implementing emotion recognition and expression in a robot enhances human-robot interaction.
2) Explain how emotion simulation in an AI system can contribute to explaining its behaviour.

4.1.3. Part 3 - Case Study: QT in this Project.

For the case study on QT, the aim is to apply the knowledge acquired in the previous parts of the scientific deliverable. Therefore, the following requirements were set:

1) Describe how QT recognizes emotions, including explanations on facial emotion recognition.
2) Explain QT's reaction to the emotion detected.
3) Discuss how it changes and enhances QT's interaction with the user.

Note that this case study is too simple to demonstrate all aspects of XAI and the emotion factor, it is only an example application of the knowledge acquired through scientific research on this project.

4.2. Design

The design of the scientific deliverable follows almost directly from the requirements. For each part of the deliverable, research is performed and the knowledge acquired from this research is presented in the production section. The following sections present how the research has been conducted for each of the parts and which directions were followed.

4.2.1. Part 1 - Presentation of Explainable AI.

The first part focuses on XAI and answers the following question: "Why do we need XAI?".

The need of XAI can best be explained by some examples of applications where humans must trust an AI. Through these examples, the reasons why humans want and need to understand why AI systems act the way they do can be identified and illustrated.

Depending on the nature of the AI (i.e. how it is was constructed, how it works internally and how it communicates with humans), various challenges may be faced. In order to understand better the difficulties arising when we want to explain AI behaviour, article [14] distinguishes three different types of AI, which are briefly presented here:

1) Opaque systems, where all computations with the input performed by the AI to reach the output are hidden from the user, e.g. black-box systems.
2) Interpretable systems, where the user has access to the computations performed by the AI and its system. Here, with some field-specific knowledge, the user may infer and interpret some information explaining the AI's behaviour, but with complex systems like neural networks, this task is very difficult.
3) Comprehensive systems, that actively communicates information to the user that will help him understand the system's decisions.

These types are not officially recognized and there is currently no universal XAI technique that will always be able to explain a certain type of AI, but it is easier to address the problem on a case by case basis once we identified the type of AI we are analysing.

4.2.2. Part 2 - The Emotion Factor in Robotics, HRI and XAI.

According to the requirements, we want to study the emotion factor from a human-robot interaction aspect and from an explainable AI aspect, and the goal is to analyse how emotions contribute to these two fields.

However, instead of directly diving into these subjects, it is interesting to start with observing emotions from a biological point of view. More precisely, we begin with analysing why emotions have been selected during evolution, because this explains the benefits of emotions for animals, including us humans. Once we understand this, we can explain more easily the impacts of emotions in the domains of HRI and XAI. So, we structured this second part of the scientific deliverable in the following way:

1) Emotions and Evolution
2) Emotions and HRI
3) Emotions and XAI

4.2.3. Part 3 - Case Study: QT in this Project.

The last part of the scientific deliverable will use the knowledge acquired before and apply it to this project's technical deliverable by analysing how QT recognizes the user's facial expression and explain its response. It is important to say here that facial emotion recognition was already implemented in QT beforehand and this feature was only reused. However, we will still explain how this emotion recognition is implemented for QT. We will then analyse QT's response to detected emotion and explain how this response to emotion affects the interaction between user and robot. So, this part is divided in two sections:

1) Emotion Recognition
2) Response to Emotion and Effect on User Interaction

4.3. Production

4.3.1. Presentation of Explainable AI.

Explainable AI is undeniably a large field and this section, as specified in the requirements, has not as objective to give a complete overview of it, but should rather demonstrate the need of XAI in the domains of artificial intelligence, machine learning and robotics. A concise definition of XAI is already given in Section 2.1 of this document, please refer to it in case the term XAI appears to be unclear.

Artificial intelligence systems are continuously improving and become increasingly complex, especially AIs capable of learning by themselves and deep neural networks. However, AI systems still make some mistakes. One could argue that an AI making a few percentage of mistakes may still be way more efficient and precise than a human for some tasks but trusting an AI that is not completely accurate and for which no one is able to explain its functioning is risky.

Take the example of an AI that should distinguish benign, i.e. harmless, from malignant, i.e. cancerous tumours based on medical scans. Recent research has shown that some AI systems are already surpassing specialists on this task and that AI is a promising solution to this problem [15], [16]. However, both false positives and false negatives still occur, and their consequences are dramatic for the affected patients. We can reduce the mistakes percentage by improving the data training set, adding further relevant features to observe etc. but all these measures will not directly address the problem inside the AI system, neither will they assert its reliability [17]. Therefore, we absolutely need to understand exactly how and why the AI system reaches its decisions.

AI in the medicine domain is only one example among many others where we need to understand why an AI acts like it does. Another could be an AI determining the social status of a person (e.g. education, health, career success, criminality etc.). Here, we have to ensure that the AI does not base its decision on inappropriate criteria like ethnicity for instance, and determining whether this the case requires humans to understand it. One more example: For an AI system in an autonomous car, not only must the producer of the car be able to ensure that the car will not endanger anyone, but pedestrians, other autonomous vehicles and drivers of normal vehicles must also have some ways to predict the car's behaviour. For instance, if a pedestrian wants to cross the road, it must be able to recognize whether an autonomous car approaching will stop for him or not [10], [18].

To conclude, there are many reasons why humans want to understand how an AI works, depending on the situation. Some important ones are to ensure it is reliable, to improve it when it makes mistakes, to avoid discrimination and even to predict its behaviour.

4.3.2. The Emotion Factor in Robotics, HRI and XAI.

It may appear strange to talk about emotions in the domain of robotics, since robots are machines and therefore unable to have emotions like humans and some animals do. However, robots can be programmed to simulate emotions to a certain extent and trained to recognize human emotions through facial expression, body language, and voice tones. This directly implies that robots can be programmed to react according to their own and the others' emotions with an appropriate, human-like emotional response.

The aim of this section is to analyse some impacts the emotion factor makes in human-robot interaction and explainable AI.

Emotions and Evolution

Before we dive into the aspect of emotions in robotics, it is important to know about the importance of emotions for humans and animals to motivate the idea of implementing emotions in robots. Therefore, it is useful to have a quick glance at emotions in biology and more precisely at why we feel and express emotions.

We could spend more than an entire book explaining different emotions, their exact purposes and benefits for the animals, but many explanations for emotions converge to the fact that they were selected during evolution, because they efficiently increase the capability of animals to respond appropriately to environmental changes [19]. For instance, an animal that becomes scared when it sees its predator is much more likely to run away to save his life than an animal that does not feel anything when seeing it.

Expressive physical behaviour is also useful, since it enables to show an animal's own emotions to the others and possibly trigger a response from them. In other words, emotions play an essential role in communication and interaction between animals capable of feeling [7]. Especially humans constantly make use of "mindreading" through analysis of the other human's behaviours to understand them better. Emotion plays an important role in the determination of our behaviours and other humans will be able to read emotion through such behaviour signs. Being able to know what others think and feel enhances communication between individuals a lot, because we understand better what others communicate us and expect from us, how we can influence on their minds and how to communicate efficiently some information, such that they really understand what we want to communicate them [10]. A good example of this is a hungry baby crying because. Its mother will immediately react to this emotional behaviour of the baby and search to identify what it needs. She will try some things until she gives the baby some food. The baby will be happy, stop crying and eat, which will tell the mother that she found what her baby wanted.

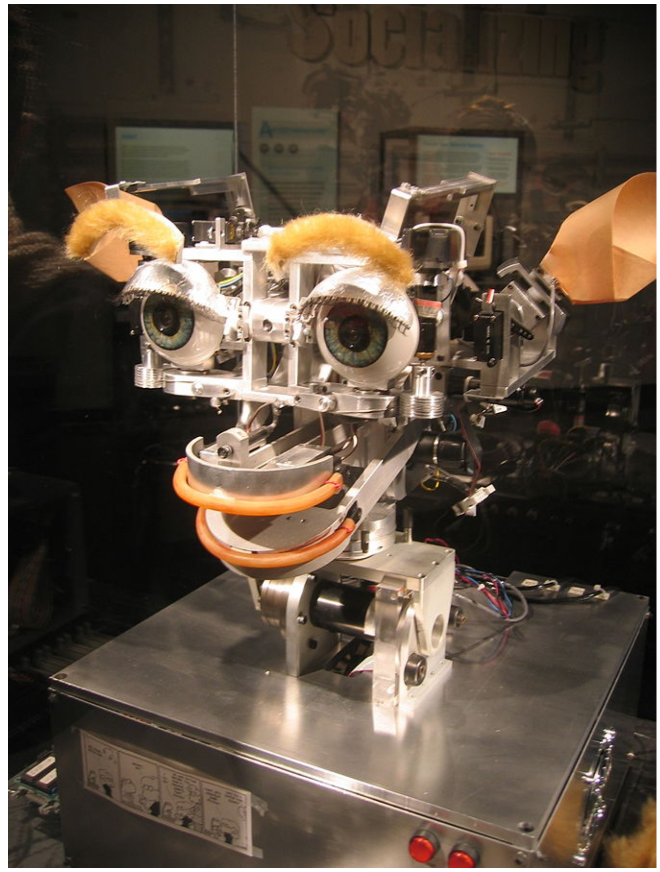

Figure 4. The robot Kismet [20].

We see that such emotional behaviours and responses play an essential role in human's everyday life and enhance our ability to understand others and communicate with them.

Emotions and HRI

Now that we know how essential emotions are in human relations, we can already guess the benefits of having emotion aware robots in human-robot interaction. Of course, emotional awareness is not useful for any robot in any field, but we can show that emotion aware social robots are much more appealing and comfortable for humans to work with than emotion unaware social robots.

The study [7] has analysed the reactions of humans to robot emotion with the robot Kismet, pictured in Figure 4, which was designed to be very expressive. This study showed concrete reactions from the human participants when they interacted with Kismet. The participants used the body language and facial expression of the robot to identify its "feelings" and adapted their voice tone according to their observations, for instance to solve misunderstanding. They also tended to mirror the robot's gestures, a psychological behaviour usually observed between humans called "mirroring". Finally, participants very often felt guilt and empathy towards the robot when the robot reacted with a sad or offended expression after they said something rude.

All these observations show that humans tend to consider the robot sensible as soon as it shows some emotional response, even thought we know that it does not really feel anything. Because of this fact, HRI can be improved by developing and training robots to communicate effectively their perceptions and intentions to humans through emotional behaviour. Furthermore, interacting with an emotion aware robot enables the human to refer to the robot as human-like entity and apply its knowledge of human interaction. This does not only increase the comprehension of the robot by the human, but also enables the robot to understand better the human though analysis of its emotional behaviour [10].

Therefore, we see that human-robot interaction with an emotion aware robot is highly enhanced with respect to emotion unaware robots, because both the human and the robot understand each other better, which directly leads us to the next part of the analysis.

Emotions and XAI

Since emotion aware robots are easier to understand for humans, this also suggests that emotions can contribute to explain an AI system.

As we have already seen in a previous section, humans do not act and make decisions only based on their goals, but also strongly according to their emotions. We can see this for instance by the fact that we use our feelings, like "scared", "happy", "sad" etc. naturally when we explain our actions. Making AI systems and robots simulating emotions will not only make them more human-like, they will also be able to use these emotions to make decisions and, more importantly, to explain these decisions with emotion descriptions or emotional behaviour. This is therefore a humanly understandable way to explain their behaviour, which is exactly what we seek in explainable AI [21].

Furthermore, it is important to notice that explaining the behaviour of an AI system or a robot through emotions is extremely convenient when it comes to explaining it to lambda users. This comes from the fact that every human understands emotion, at least to a certain extent, because they are inherently part of humans. A human using an emotion aware robot will understand it directly through the use of emotions and therefore be able to assess on its reliability without the need for scientific experts to demonstrate it. As long as the implementation of emotions is approximately true to reality, this is a great advantage for the use of AI and robots in the real world, like for our case study QT.

4.3.3. Case Study: QT in this Project.

With the knowledge acquired, we will now analyse QT in this project, according to the requirements and design described previously.

Emotion Recognition

Facial expression recognition is a feature which is implemented in QT via an application called Nuitrack.

Nuitrack uses computer vision techniques to extract information from images captured through a camera. It starts by determining a rectangle around the face of the person on the camera image and normalizing the coordinates inside this rectangle, such that all points inside the rectangle can be indicated by tuples (x, y) with x and y both between 0 and 1. It then analyses the face and determines the position of certain face features with points, as shown in Figure 5. Based on this information, the application can compute a confidence degree for each of the four emotions it is capable to recognize, i.e. neutral, angry, happy and surprised [22].

Information about the exact technique used to compute this confidence degree could not be found, but one possibility is machine learning algorithms.

Response to Emotion and Effect on User Interaction

We described above the implementation of emotion recognition by QT. However, if QT does not change its behaviour according to the emotion recognized, the benefit of detecting the user emotion is null. Therefore, it is useful

Figure 5. Set of 31 points detected by Nuitrack on a human face [22].

to implement something we could call an “emotional response”, which means an adaptation of QT’s behaviour depending on the emotion of the user. This is exactly what we did in the technical deliverable of this project, described in Section 5.

The type of response we implemented is that QT changes the notes played by the user, depending on the emotion recognized. The user plays a melody and QT shifts the notes to match better the user’s emotion. This response implies that QT is taking part in the process of playing music, so user and robot actually play together. This increases the level of interaction between human and robot, which is especially beneficial for children.

Note that QT was designed to be very expressive, so implementing further, different emotional responses is totally feasible. For instance, writing a program that makes QT respond to emotion by showing an appropriate face and simulate an emotion from his side is very easy with the built-in facial expressions of the robot, like the ones shown in Figure 6.

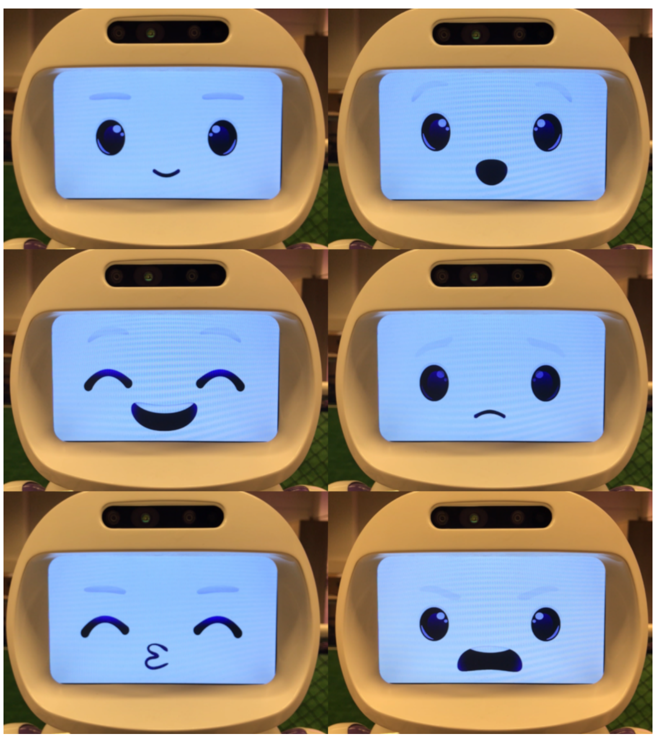

Figure 6. Some faces of QT. From top-left to bottom-right: neutral, surprise, happy, sad, kiss, angry.

4.4. Assessment

For each of the three parts of the scientific deliverable, we defined some requirements in Section 4.1 which must be fulfilled in order to consider the scientific deliverable as complete. In this section, we will verify whether all requirements are met.

In the first part, we emphasized the need of XAI in different domains where AI is used by showing some examples and explaining different benefits of understanding why the AI behaves the way it does. This meets perfectly the requirement for this part.

The second part explains in some details on how implementing emotion simulation in AIs and robots enhances human-robot interaction and makes AI systems more understandable for users, especially for lambda users, who do not have sufficient domain-specific knowledge to understand more scientific or technical forms of explanations. Thus, the two requirements for this section are fulfilled.

Finally, in the third part, we analyse how QT recognizes emotions using the application Nuitrack. We also explain the emotional response of QT implemented in the program developed as the technical deliverable of this project, and demonstrate how this response benefits to the interaction between user and robot. So, the three requirements set for this last part are also met.

Since all requirements are fulfilled, the scientific deliverable is completed successfully.

5. QT as Emotion Sensible Instrument

5.1. Requirements

The aim of this project is to make QT an emotion sensible instrument. Therefore, the technical deliverable is a python code working on QT that changes the music played with the robot according to the facial expression of the player.

To be able to assess that the goal is met, some requirements were set for this deliverable. First, we defined the functional requirements:

1) The program must be easily compatible with any program that turns QT into a music instrument.
2) The program must take the note played by the user and the emotion recognized on the user’s face as input.
3) The program must consider which emotion is detected on the user’s face and, to a possible extent, modify the note such that it reflects this emotion, e.g. if the user looks happy, the modification of the notes played should make the music sound happier.
4) Then, the program must output a note, which is the original note modified according to the emotion detected, and make QT play it instead of the original one.

We also defined some non-functional requirements, which are not necessary, but enhance the user experience:

1) When QT is plugged to an external screen, it should show its camera view in a window on the screen, such that the user can see what the robot sees.

2) The code must be easily adjustable and extensible, e.g. the notes modification and emotions detected can be changed easily, and further functionalities can be added to the program without having to rethink it through.

5.2. Design

The algorithm of this project's technical deliverable is quite straightforward and has no real constraint except the technical capabilities of the robot. However, the structure of the program was constrained by one important factor: the ROS architecture. Therefore, it is essential to understand it before we present the architecture of the program itself.

5.2.1. Algorithm.

The basic algorithm of the program is very simple. The program must constantly analyse the data obtained through the robot's front camera and detect the emotion of the user using facial expression recognition. This information must be constantly updated, since the user can change his or her emotion at any time.

Furthermore, the program must take as input the notes that the user is playing using QT, change it according to the emotion detected and play the new note instead of the initial one.

The program is supposed to make the music played by the user sound appropriate according to the emotion detected. However, this is not really possible, because the program has no control over the sequence of notes played by the user, only over each individual note. However, we can still try to define a certain way of changing the music that should help the user play music that fits better his or her emotion.

Therefore, the program is changing the notes by performing a shift of a certain number of semitones forward (higher) or backward (lower). Since QT is capable to read four different emotions on human faces, we have four shifting values to define: neutral, angry, happy and surprised. We defined them as follow for our program:

1) Neutral: 0, because neutral means no particular emotion. Therefore, the music played by the user should not be modified.
2) Angry: -2 (i.e. two semitones backwards), because in music, angry emotions are often translated into lower tones in the music.
3) Happy: 2, since a happy emotion evokes a brighter mood and higher notes in music.
4) Surprised: 4, because the surprised emotion is an intenser feeling that lasts less time than the other emotions considered recognized by QT. Since the surprised emotions here has no negative conotation, it can be associated with an unusual big forward shift of the notes, resulting is very high notes.

These shift choices are thought through, but still subjective. As long as they correspond to the purpose of adapting the music to the emotion of the user, other shifting patterns may also be chosen for such a program.

The structure of the program was chosen to match this algorithm, but before we present it, we need to understand the ROS architecture.

5.2.2. ROS Architecture.

As already mentioned in section 2.1, ROS has a very specific architecture, which is known as the publisher and subscriber architecture. In ROS, its implementation is based on nodes, messages and topics.

Nodes are individual processes that perform computation and use ROS to communicate with other nodes. They can be viewed as modules. Systems using ROS are usually composed of many nodes, because ROS is designed to be modular, and all these nodes can be represented in a graph to model the interactions between them.

Messages are simple data structures with fields, each containing information of certain types (e.g. integer, string, boolean). These are the pieces of information exchanged between nodes to communicate.

These exchanges occur via topics. Nodes communicate information to other nodes by publishing messages on topics, and listen to information sent by other nodes by subscribing to topics and this way receiving and treating messages sent by them. [23]

All communications, including publisher, subscriber, topics and message transmissions, are monitored by the ROS Master, a special node which must be started first in order to use ROS. This means that while programming with ROS, we don't have to worry about handling the communications between the nodes. The ROS Master will automatically do it for us. [24]

This architecture is convenient for many robots, because robots are built out of many individual components that must be able to execute independently from each other, but still communicate state changes and other information to other components. [25]

5.2.3. Program Structure.

The structure of the technical deliverable follows the ROS architecture. The program is divided into three python files, each of them being one node, and each of these nodes communicates with each other by publishing messages on and listening to topics.

The first node is called emotionReader. It is responsible for detecting the emotions from the user's facial expression through the front camera of the robot.

The second node is the emotionConverter node, which will change the individual notes played by the user according to the emotion detected.

The last node is called musicPub and is there to publish notes onto a topic to which the emotionConverter node is listening. This node is actually not part of the program, but it was needed for testing purposes. Furthermore, it also serves as a placeholder for any other node system that would publish notes to be played on QT and which could therefore be used together with this project's technical deliverable.

These nodes will work together in a system by communicating over the topics, sending messages with relevant information to the subscribed nodes. More precisely, emotionReader will publish the emotion detected on the user's face on a topic (/QTInstrument/emotion) and the musicPub node regularly publishes a note on another topic (/QTInstrument/music). The emotionConverter node subscribes to both these topics and receives the messages sent by the two other nodes. Using these two information, i.e. the emotion and the note played, emotionConverter will shift the note received by a certain number of semitones (higher or lower, depending on the case) and publish the newly obtained note to a built-in node of QT that will play the note. Figure 7 is a schema of the system described.

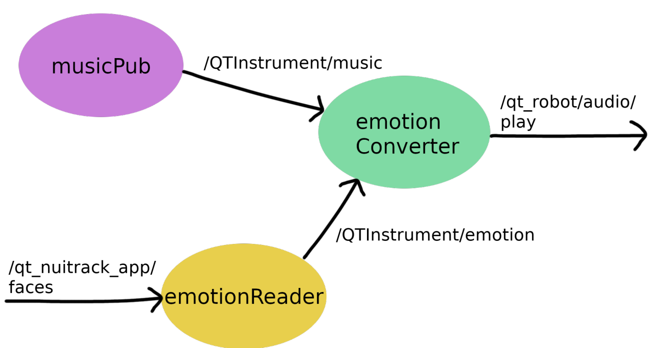

Figure 7. Schema of the system structure showing nodes and publisher-subscriber relations.

5.3. Production

The production of the technical deliverable has two aspects, which are presented in the following subsections. The first one is the development process and the second one is the code produced.

5.3.1. Development Process.

The technical deliverable was produced and tested on QTrobots from the AI Robolab at the University of Luxembourg. We started using a QT from the second-to-last version, which is the blue one pictured in Figure 3, and continued later with a QT of the most recent version[2], pictured in Figure 8, because the idle movements of the pink QT were deactivated, so they did not bother during testing of the program anymore. For simplicity, the blue one was called QTBoy during the project and the pink one QTGirl. The two versions have no noticeable difference for us and the code produced for this project's technical deliverable works fine on both versions of QT, so which one we use is irrelevant.

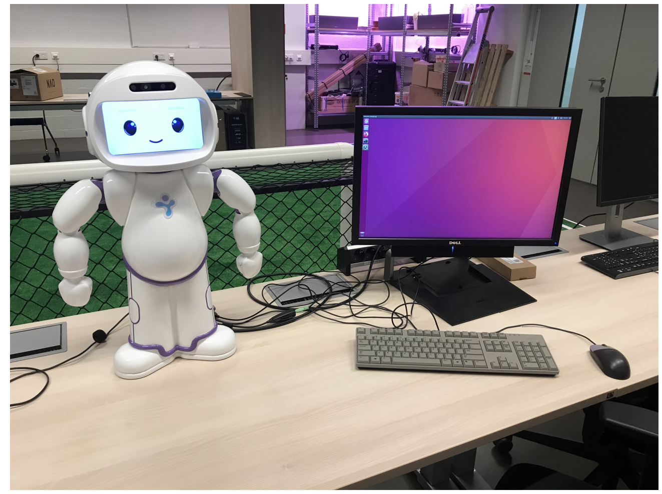

Figure 8. Workspace with QT.

Github was used to transfer the code from QT to external computers and vice-versa, to enable to work on the code without needing the robot and test it later on QT.

Furthermore, as already mentioned in Section 2.3, QT has a special internal architecture that complicated the execution of the code on QT and therefore also the testing procedure. Indeed, QT possesses two computers: One in the head, called QTxxx[3], and one in the belly, called QTPC. These two computers are connected together via an ethernet connection. The robot is fully functional with QTxxx only, but QTPC was added to add more processing power for some applications that required it[4].

However, during the production of the deliverable, when it came to testing the code, it turned out that we could not display QT's camera view on the desktop from QTxxx, because we connected to QTxxx through ssh from QTPC, and QT's speaker was inaccessible from QTPC. Therefore, we had to split the program to run the emotionConverter node on QTxxx and the other two nodes on QTPC, and make both computers run on the same rosmaster node, such that the nodes of our program could communicate with each other on the same topics. This way, the program runs correctly.

Because of these additional complications, a user guide ("README.md") in markdown explaining step by step how

2. From 2019

3. "xxx" denotes a 3 digit number that differs depending on QT's version. It was 110 for QTBoy and 132 for QTGirl.

4. According to Dr. Agustin Ambrossio, who participated in building QT at LuxAI.

to set up and run the program is provided with the technical deliverable.

5.3.2. Code Presentation.

The entire code was written with the help of the documentation [26] and its structure is inspired by code samples in [27].

With the program architecture described in Section 5.2, it just remains to implement it. However, before coming to the implementation of the nodes, it had to be decided by which means QT would emit individual notes. Since it was not clear how QT could emit an arbitrary sound by itself, the only solution we came up with was to create short audio files for each note we want QT to be able to play. This procedure is quite cumbersome, but does not go against the requirements.

One folder per octave we wanted to reach was created, each containing the audio files for the notes of the octave in waveform audio file (wav) format. These were named from 1.wav to 12.wav according to the semitone they corresponded to, with 1 being note C and 12 note B, as pictured in Figure 9. This naming convention was chosen to facilitate the implementation in the code, because computations are done with numbers.

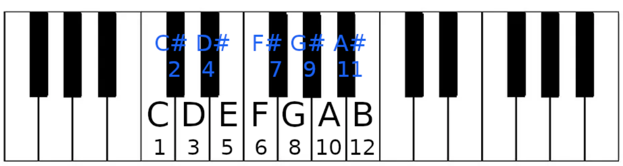

Figure 9. Numbers and corresponding notes on a piano.

Remark that from this project's point of view, only a few notes, not even an entire octave, are sufficient to demonstrate how the program works. However, working with the general objective in mind to use QT as an instrument, we decided to think further and considered an implementation that takes multiples octaves into account.

Now that we know how to make QT emit notes, we can present the program. As already presented in the design section, it consists of three nodes, each of them having its own functionality. For each of the node, we created a python file in which we initialize the node and encode the features we wanted to provide. The following sections describe in details the implementation of each node.

Node 1: emotionReader

The role of the emotionReader node is to capture the emotion of the user and to publish it on the topic /QTInstrument/emotion. For a simpler code implementation and to avoid having to handle anything related to the emotion itself in another node, this node will actually publish directly the shift defined to correspond to the emotion detected on the topic. This way, the node listening to /QTInstrument/emotion, in our case emotionConverter, will just have to perform the computation to shift the note.

Detecting the emotion of the user (among the four emotions that QT is able to recognize) is a feature that is already implemented in QT by default. But the information about the emotion is sent in a message along with many other information that we do not need on the topic /qt_nuitrack_app/faces. So, emotionReader subscribes to this topic and whenever it receives a new message from this topic, it extracts all information about the emotion detected and processes it further.

Indeed, the information on the user's emotion is presented in form of a certain confidence degree for each of the four possible emotions [22]. Therefore, the node must compare all percentages to determine the highest one and deduce that this is the emotion shown by the user. Finally, the node publishes the shift corresponding to the emotion detected on the topic /QTInstrument/emotion.

An additional functionality of this node is to show the camera view of QT in a window on the desktop, to give some visual feedback to the user on what QT sees. Therefore, the node subscribes to the topic /camera/color/image_raw and uses the framework openCV to make the image appear in a window on the desktop.

Node 2: musicPub

The role of the musicPub node is only to repeatedly publish a note to the topic /QTInstrument/music, such that the emotionConverter has an input note to shift and publish. It is only there for testing and as placeholder for any other program that could make an instrument out of QT.

However, we needed a way to have the octave number and the tone in one same message, to avoid synchronization problems that would be caused by sending two different messages (on one same or two different topics). One possibility would have been to create a message type with two fields, one for the octave and the other for the tone, and to send messages of this type on the topic. This solution was not chosen, because there is an easier solution, which is simply to send integer messages in the format ONN, where O is the digit corresponding to the octave and NN are the two digits corresponding to the note of the octave. For instance, the message "310" corresponds to octave 3, note A.

Of course, any program that someone may want to use in place of this node must publish the notes in the same format described above, because the emotionConverter node, which will handle these messages, is expecting the message to be formatted this way.

Node 3: emotionConverter

The role of the emotionConverter node is to modify the note received on the /QTInstrument/music topic by the shift received on the /QTInstrument/emotion topic, and to publish the new note, more exactly the name of the audio file corresponding to the new note, on the topic /qt_robot/audio/play, which will make QT play the audio file.

Therefore, emotionConverter subscribes to both the topics /QTInstrument/music and /QTInstrument/emotion, and handles callbacks for both of them:

- Whenever it receives a new message on the topic /QTInstrument/emotion, indicating the shift by which the next note must be changed, it stores this shift in a variable.
- Whenever it receives a note, more precisely an integer representing the note, on the topic /QTInstrument/music, it will extract the tone by taking the integer modulo 100 and the octave by using integer division by 100. Then, it will modify the note by the shift currently stored, adjust the octave in case the shift made the note reach an octave higher or lower. Finally, it will find the folder corresponding to the new octave and the file name for the new note, and send the path to this file on the topic /qt_robot/audio/play.

The nodes system is now fully operational and the technical deliverable is complete.

5.4. Assessment

The goal is to create a program that could change music played with QT according to the facial expression of the user in front of it. To measure whether this goal was met, we fixed some requirements in Section 5.1 that must be fulfilled in order to consider the technical deliverable correct. In this section, we will go over the functional and non-functional requirements and ensure that all of them were met.

The program was designed modular, following the ROS architecture, which automatically makes it very easy to extend it with further features. Furthermore, the node musicPub is a placeholder for any program that makes QT play music, as long as this program sends messages in the described format over the topic /QTInstrument/music. Therefore, the technical deliverable of the project is easily compatible with other programs, because it was designed for this purpose. So, the first functional requirement and the second non-functional requirement are both met.

The other requirements concern the features of the program. The node emotionReader is extracting the emotion of the user and deduces a corresponding shift of the note played, which was defined during the design of the program to correspond well to the emotion. The node emotionConverter will receive the shift and the note played by the user, shift the note and send it over another topic that will make QT play the note. So, the algorithm and implementation for the program fulfill the functional requirements 2, 3 and 4.

Finally, emotionReader has a secondary purpose, which is to display QT's camera view on the desktop. This fulfills non-functional requirement number 1.

All requirements have been met, so the technical deliverable is complete.

Acknowledgment

I, Iris Kremer, would like to thank my project academic tutor Dr. Amro Najjar for helping and guiding me through this semester project. Many thanks also to Dr. Agustin Ambrossio for his help on QT. I also want to thank Eliott Bonte, with whom I worked part of the time due to our complementary projects.

Then, I also owe thanks to the BiCS management team for establishing the bachelor semester projects in BiCS [28], [29].

Finally, I would like to thank my family and my friends for their support and love.

6. Conclusion

This paper presented the third bachelor semester project made by Iris Kremer, under the direction of her project academic tutor Amro Najjar. This project produced one scientific and one technical deliverable. The scientific deliverable aimed to present explainable AI and analyse the emotion factor in HRI, XAI and robotics. As for the technical deliverable, a program was developed which enables to use QT as emotion sensible instrument. Both scientific and technical deliverable meet all their respective requirements and therefore, the project is considered as a success.

As for the scientific deliverable, we could continue the presentation of XAI by analysing some existing XAI techniques that enable us to explain some AI systems, even though there do not exist many of them yet. Furthermore, using emotions in robots may pose some ethical issues when such emotion aware robots are used in society. Robot ethics is therefore also an axis to investigate in the future.

For the technical deliverable, a possible future improvement is, if possible, to include a launch file that would start the entire program at once. Furthermore, there may be a better solution than using short audio files to make QT emit notes, so this axis could be investigated further.

On the long term, we can combine the program developed in this project with another program that makes an instrument out of QT. We could then for instance consider creating some music lessons on QT, intended for children to learn music.

References

[1] Margaret A Boden. Creativity and artificial intelligence. *Artificial Intelligence*, 103(1-2):347–356, 1998.

[2] Obvious. http://obvious-art.com/. Online; accessed 30th of November 2019.

[3] Artificial intelligence software. Edmond de belamy. https://commons.wikimedia.org/wiki/File:Edmond_de_Belamy.png, 2018. Online; accessed 11th of November 2019.

[4] Mason Bretan and Gil Weinberg. A survey of robotic musicianship. *Communications of the ACM*, 59(5):100–109, 2016.

[5] Jeong-Hye Han, Dong-Ho Kim, and Jong-Won Kim. Physical learning activities with a teaching assistant robot in elementary school music class. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 1406–1410. IEEE, 2009.

[6] Alireza Taheri, Ali Meghdari, Minoo Alemi, Hamidreza Pouretemad, Pegah Poorgoldooz, and Maryam Roohbakhsh. Social robots and teaching music to autistic children: Myth or reality? In *International Conference on Social Robotics*, pages 541–550. Springer, 2016.

[7] Cynthia Breazeal. Affective interaction between humans and robots. In *European Conference on Artificial Life*, pages 582–591. Springer, 2001.

[8] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2, 2017.

[9] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93, 2019.

[10] Thomas Hellström and Suna Bensch. Understandable robots-what, why, and how. *Paladyn, Journal of Behavioral Robotics*, 9(1):110–123, 2018.

[11] Human robot interaction. http://humanrobotinteraction.org/1-introduction/. Online; accessed on the 5th of January 2020.

[12] LuxAI. Qt for autism. http://luxai.com/why-qtrobot-for-children-with-autism-works/. Online; accessed 18th of November 2019.

[13] Official ros website. https://www.ros.org/. Online; accessed 30th of November 2019.

[14] Derek Doran, Sarah Schulz, and Tarek R Besold. What does explainable ai really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*, 2017.

[15] Yuzheng Wu, Maryellen L Giger, Kunio Doi, Carl J Vyborny, Robert A Schmidt, and Charles E Metz. Artificial neural networks in mammography: application to decision making in the diagnosis of breast cancer. *Radiology*, 187(1):81–87, 1993.

[16] Artificial intelligence better than humans at spotting lung cancer. https://www.medicalnewstoday.com/articles/325223.php#1. Online; accessed 13th of December 2019.

[17] Randy Goebel, Ajay Chander, Katharina Holzinger, Freddy Lecue, Zeynep Akata, Simone Stumpf, Peter Kieseberg, and Andreas Holzinger. Explainable ai: the new 42? In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 295–303. Springer, 2018.

[18] Milecia Matthews, Girish Chowdhary, and Emily Kieson. Intent communication between autonomous vehicles and pedestrians. *arXiv preprint arXiv:1708.07123*, 2017.

[19] Edmund T Rolls. *Emotion explained*. Oxford University Press, USA, 2005.

[20] Wikimedia commons, picture of kismet. https://commons.wikimedia.org/wiki/File:Kismet_robot_20051016.jpg.

[21] Frank Kaptein, Joost Broekens, Koen Hindriks, and Mark Neerincx. The role of emotion in self-explanations by cognitive agents. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 88–93. IEEE, 2017.

[22] Nuitrack face detection. http://download.3divi.com/Nuitrack/doc/Instance_based_API.html. Online; accessed on the 23rd of December 2019.

[23] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[24] Ros tutorials on the ros wiki. http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes. Online; accessed 7th of December 2019.

[25] Somphop Limsoonthrakul, Matthew N Dailey, Methee Srisupundit, Suwan Tongphu, and Manukid Parnichkun. A modular system architecture for autonomous robots based on blackboard and publish-subscribe mechanisms. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 633–638. IEEE, 2009.

[26] Ros wiki - qtrobot documentation. http://wiki.ros.org/Robots/qtrobot. Online; accessed last on the 22nd of December 2019.

[27] Luxai qtrobot examples repository. https://github.com/luxai-qtrobot/tutorials. Online; accessed last on the 22nd of December 2019.

[28] Bics semester projects reference document. https://www.dropbox.com/sh/h83eajpydshiqat/AACqnGS8xFFnbhFcuD3TZwYQa/BSP/bics-bsp-reference-document.pdf?dl=0. Technical report, University of Luxembourg (2018).

[29] Bics bachelor semester project report template. https://github.com/nicolasguelfi/lu.uni.course.bics.global. University of Luxembourg, BiCS - Bachelor in Computer Science (2017).