Computer Methods and Programs in Biomedicine 225 (2022) 107077

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

An adaptation algorithm for personalised virtual reality exposure therapy

Joris Heyse[a,*], Barbara Depreeuw[b,c], Tom Van Daele[b], Tine Daeseleire[b], Femke Ongenae[a], Femke De Backere[a], Filip De Turck[a]

[a] IDLab, Department of Information Technology Ghent University - imec, Technologiepark-Zwijnaarde 126, Ghent 9052, Belgium
[b] The Human Link, Grotesteenweg 93, Antwerp 2600, Belgium
[c] University Psychiatric Centre Duffel, Stationsstraat 22, Duffel 2570, Belgium

ARTICLE INFO

Article history:
Received 7 April 2021
Revised 31 March 2022
Accepted 17 August 2022

Keywords:
Virtual reality exposure therapy
Cognitive behaviour therapy
Decision support system
Personalisation

ABSTRACT

Background: Anxiety disorders are highly prevalent in mental health problems. The lives of people suffering from an anxiety disorder can be severely impaired. Virtual Reality Exposure Therapy (VRET) is an effective treatment, which immerses patients in a controlled Virtual Environment (VE). This creates the opportunity to confront feared stimuli and learn how to deal with them, which may result in the reduction of anxiety. The configuration of these VEs requires extensive effort to maximise the potential of Virtual Reality (VR) and the effectiveness of the therapy. Manual configuration becomes infeasible when the number of possible virtual stimuli combinations is infinite. Due to the growing complexity, acquiring the skills to truly master a VR system is difficult and it increases the threshold for psychotherapists to use such useful systems. We therefore developed a prototype of a supportive algorithm to facilitate the use of VRET in a clinical setting. This automatised system assists psychotherapists to use the wide range of functionalities without burdening them with technical challenges. Thus, psychotherapists can focus their attention on the patient.

Methods: In this paper both the prototype of the algorithm and a first proof of concept are described. The algorithm suggests environment configurations for VRET, tailored to the individual therapeutic needs of each patient. The system aims to maximise learning during exposure therapy for different combinations of stimuli by using the Rescorla–Wagner model as a predictor for learning. In a first proof of concept, the VE configurations suggested by the algorithm for three anonymised clinical vignettes were compared with prior manual configurations by two psychotherapists.

Results: The prototype of the algorithm and a first proof of concept are described. The first proof of concept demonstrated the relevance and potential of the proposed system, as it managed to propose similar configurations for the clinical vignettes compared to those made by therapists. Nonetheless, because of the exploratory nature of the study, no claims can yet be made about its efficacy.

Conclusions: With the increasing ubiquity of immersive technologies, this technology for assisted configuration of VEs could make VRET a valuable tool for psychotherapists.

1. Introduction

Anxiety disorders are highly prevalent, often chronic, mental health disorders. On average, one in four individuals in the United States and Europe suffer from such a condition across their lifetime [1,2]. Specific phobias, panic disorders, and Post-Traumatic Stress Disorder (Ptsds) are examples of anxiety and trauma-related disorders that can severely impact disability and impairment. Fortunately, effective treatments exist, like Exposure Therapy (ET), which is grounded in Cognitive Behaviour Therapy (CBT) [3,4]. During ET, the patient is repeatedly and systematically exposed to a feared stimulus in a safe context. For example, patients with arachnophobia are exposed to spiders and gradually learn to deal with the elicited fear. ET tends to make a patient less anxious or makes the anxiety less debilitating for specific stimuli, although this is not a necessity [5]. Understanding what makes the patient

[*] Corresponding author.
E-mail addresses: joris.heyse@ugent.be (J. Heyse), barbara.depreeuw@emmaus.be (B. Depreeuw), tom.vandaele@thomasmore.be (T. Van Daele), tine@thehumanlink.be (T. Daeseleire), femke.ongenae@ugent.be (F. Ongenae), femke.debackere@ugent.be (F. De Backere), filip.deturck@ugent.be (F. De Turck).

https://doi.org/10.1016/j.cmpb.2022.107077
0169-2607/© 2022 Elsevier B.V. All rights reserved.

anxious is essential to create a proper exposure exercise. If a patient is afraid of an external cue, e.g. driving a car on a freeway or in a tunnel or taking an elevator, they should be exposed to similar contexts. Complementary, interoceptive cues, e.g. feeling one's heartbeat or experiencing blurred vision, are often both the object of a patient's fears and essential symptoms of their condition. Therefore, elicitation of these cues is fundamental to conduct the proper exposure exercise. With these requirements, different forms of ET are possible, e.g. in vivo (in real life), imaginary, or in Virtual Reality (VR) [6,7]. To augment both the reach and effectiveness of existing treatments and services, this latter form of Virtual Reality Exposure Therapy (VRET), in which VR technology enhances exposure treatment, is increasingly being explored [8]. With VR technology, it is possible to simulate almost any scenario. This creates the opportunity for fine-tuned personalised VR environments for ET. However, configuring these environments is not trivial and introduces an additional workload for the psychotherapist.

This manuscript is the product of interdisciplinary research between psychotherapists and computer scientists. The main objective of this paper is the presentation of a prototype of a novel adaptation algorithm for VRET which generates personalised VR environments to treat patients with anxiety. The scalability of the algorithm is investigated in terms of execution time relative to the amount of input data. Furthermore, a proof of concept VRET application called PATRONUS has been designed and implemented in which the algorithm is integrated. The proof of concept focusses on fear of driving a car, claustrophobia, and panic disorders. In this proof of concept, data of three prior anonymised clinical vignettes are used to generate suggestions with the adaptation algorithm. These suggestions are compared with the manual configurations of two psychotherapists. The adaptation algorithm was not used in a clinical setting to steer the patients' therapy.

This paper presents the algorithm by describing the underlying theory and techniques from both the psychological and computer science point of view, making it relevant for experts from both domains. The remainder of the paper is structured as follows. Section 2 gives a theoretical foundation of the psychological learning mechanisms that are relevant for ET. Section 3 continues with the methodology for the design of the presented adaptation algorithm and proof of concept application. This section covers the outline of the system in which the adaptation algorithm is integrated as well as the knowledge and data management techniques used. In Section 3.4, the PATRONUS proof of concept, a VRET system that integrates the adaptation algorithm, is described in detail. The prototype of the adaptation algorithm is presented in Section 4.1, the technical details of the implementation are explained in Section 4.2. A scalability test on the execution time of the algorithm is presented in Section 4.3. The results of the comparison between the generated suggestions on three clinical vignettes are presented in Section 4.4. Finally, a discussion and conclusion is given in Section 5.

2. Related work

VR has a proven track record. Contrary to common belief, it has effectively been used in clinical practice for over two decades [9–11]. The therapeutic approach relying on VR, known as VRET, was initially primarily offered in specialised settings, i.e. treatment centres for war veterans in the United States suffering from PTSD [12]. One of the main advantages of VRET, compared to conventional ET, is the possibility to immerse patients in Virtual Environments (VEs) that are highly controllable and customisable. This is particularly interesting for the treatment of anxiety disorders. Despite the common ground of anxiety disorders, each patient is still unique. Across patients, there may not only be subtle differences between feared stimuli but also in terms of the pace at which the patient progresses. Determining when a patient is ready for *the next step* is therefore individually determined. Optimising VE and the pace at which they evolve and tailoring that to individual patients is therefore necessary. Personalised VEs tend to increase the potential of creating a sense of presence, offering patients the idea of really being in a different place than where they physically are [13]. To some extent, the technology itself can facilitate this by creating a highly immersive experience, e.g., accommodating a wide field of view in the virtual world or offering highly realistic VE [14]. However, research indicates that creating a feeling of being a part of a VE is not only a characteristic of the technology itself. Tailoring the VE to the particular fears of individual patients can play an important role as well [15].

Although found to be clinically effective, the technical nature of VRET and the associated costs did not allow this treatment form to become widely available yet. In recent years, VR systems ongoing commercialisation has opened up possibilities for more widespread use beyond these settings [16]. A lower hardware cost and increased accessibility of relevant software applications and platforms hold promise. Alongside established, extensive VRET platforms, e.g. Oxford VR[1] and Psious,[2] a low-cost stand-alone VRET-app with demonstrated clinical effectiveness has also made its way to market [17]. Nevertheless, this does not imply that no challenges remain for psychotherapists to implement VRET. Psychotherapists are not necessarily tech-savvy or are insufficiently familiar with the wide range of possibilities and features that broadly available VR systems increasingly support [18]. The current solutions are either very static with predefined environments or are dynamic with an overload of manually configurable options. The static environments are easy to use but lack the possibility to personalise the therapy to the patient's needs completely. The dynamic environments allow this personalisation at the cost of increased complexity, hindering psychotherapists from using these tools effectively. The latter can, in part, be overcome through vocational training, in which psychotherapists are informed and trained in interacting with VRET - and other - technological extensions to conventional practice [19]. As possibilities of VE are incrementally increasing, so do the possibilities for tailoring these environments. The increasing complexity implies that truly mastering a system will also require increasing and ongoing amounts of training, which introduces a novel threshold and limits the full potential for clinical practice. Furthermore, a VRET system should be easy to use, and the psychotherapist should possess the required knowledge to adopt a VR tool in therapy effortlessly. Otherwise, the VR system draws the psychotherapist's focus away from the patient, damaging the patient-therapist bond.

One way to deal with this challenge is through the development of supportive algorithms. Such algorithms could facilitate VR systems to consider clinical expertise without requiring psychotherapists to grasp the technical nuances fully. Nor do they need to know by heart the wide range of functionalities available in a system. More specifically, adaptation algorithms are needed to automatise this translation from expert observations and insights to environmental adaptations in VR. This will take away the burden of requiring psychotherapists to manually create the tailored environment they consider most suitable for each step of a VRET. At the same time, it can help them keep their focus during sessions to where it matters the most: on their patients.

A vast amount of lab research aims at uncovering the learning processes underlying ET. For this purpose, psychotherapists often use a Pavlovian fear condition paradigm to model the pathogenesis of anxiety disorders and ET [5,20,21]. This procedure consists

[1] https://ovrhealth.com.

[2] https://psious.com.

of two parts, starting with the fear acquisition to mimic the development of an anxiety disorder and then followed by fear extinction as a model for ET. In the fear acquisition, a neutral stimulus Conditioned Stimulus (CS) is repeatedly presented to the subject, followed by an aversive stimulus Unconditioned Stimulus (US). As a result, the CS acquires a dangerous meaning and starts eliciting fear reactions in anticipation of the US (routinely measured as enhanced physiological arousal, US-expectancy and increased negative valence of the CS). This is commonly termed the Conditioned Response (CR). The current dominant view is that during conditioning, the person forms a mental association between representations of the CS and the US [22]. Novel confrontations with the CS will thereby activate the representation of the US as well, which produces fear reactions. Subsequently, in fear extinction, the subject is repeatedly exposed to this CS in the absence of the US, and the conditioned fear reactions decrease. This fear extinction procedure is similar to what happens in ET. However, what drives this fear reduction?

A rationale why fear declines during extinction is that the US is expected but not delivered whenever the CS appears. This violates a patient's expectancy, which drives new learning to update false expectations. Violations of these false expectations of danger are therefore vital in extinction learning. Researchers have formalised this notion in formulas that stipulate how expectations are updated after each violation. As in the influential theory of Rescorla and Wagner [23], the standard approach is to calculate the difference between what a patient expects and what has actually happened. This allows evaluating the amount of learning on any given extinction trial.

Furthermore, the Rescorla–Wagner (RW) model allows for an idiosyncratic formalisation per patient per exposure exercise. In theory, such an approach would allow to personalise and optimise VEs. A therapist, however, cannot perform the calculations of the RW model to determine the amount of learning in real-time during VRET. Therefore, this paper presents an algorithm that uses the RW model that will play an essential role in supporting a therapist to maximise potential learning during exposure exercises. The algorithm suggests VEs that are personalised and optimised for each individual patient at every step during the therapy process.

3. Methods

In this work, a prototype of an algorithm for adapting VR environments is presented, specifically for VRET. The adaptations aim to optimise the configuration of these VEs to match the patient's needs in each therapy exercise, as indicated by their psychotherapist. The adaptations are performed based on profile information of the patient, which describes characteristics of the phobia, and context information, which describes what happens during exposure exercises. The presented adaptation algorithm is to be integrated in a system for administering VRET. In this study the PATRONUS application is designed as a proof of concept in which the adaptation algorithm is implemented. This section describes this system and the proof of concept in detail.

3.1. System overview

The adaptation system for VR content in ET consists of multiple software and hardware components. Fig. 1 gives a high-level overview of the system. The personalised adaptations are calculated in the *Adaptor* component, which uses data about the patient and the exercises they performed. All the data is gathered and stored in a *Knowledge base* that combines that data with prior knowledge from domain experts, e.g. the knowledge introduced in the background section of this work. The *Knowledge base* offers an access point for the *Adaptor* to get the information needed to make adaptations. The output of the *Adaptor* is multiple suggestions for new environments. The psychotherapist then selects one of these suggestions. Before presenting the environment to the patient through the *VR system*, the psychotherapist can still manually change the configuration. By generating multiple suggestions, the psychotherapist can still decide how to advance the therapy as there often is not only one single solution in exposure therapy but multiple valid approaches.

Because the psychotherapist has, at all times, complete control over what the patient sees, the adaptation system is a Decision Support System (DSS) [24]. It presents the information in an easy to digest format while leaving the final decision to a human expert. Before, during and after each exposure exercise, information is collected from the patient and stored in the *Knowledge base* for configuration of the next exercise.

3.2. Knowledge base

The *Knowledge base* is a collection point for all information and knowledge. This section describes which data is used, how data is modelled and how knowledge is extracted from that data.

3.2.1. Data

Data is fed to the adaptation algorithm to generate new personalised adaptations for the VE. On the one hand, the user's profile information identifies the characteristics of their anxiety, e.g. the patient is uncomfortable in busy places and apprehensive of bright lights. On the other hand, the used context data is the level of anxiety during exposure which aids in understanding the effect of the performed exercises. The patients express their level of anxiety using the Subjective Units of Distress Scale (SUDS). It expresses the patient's perception of discomfort as a number between 0 and 10, which is widely used for ET [25]. The context data allows designing exercises tailored to the patient's phobia at each step throughout the therapy.

The profile information is collected by having the patient fill out a VR parameter questionnaire. This questionnaire is specifically constructed for this work and depends entirely on the specific environments supported by the system. The questionnaire inquires the patient about their expected reactions to the set of stimuli in the VR system. All these questions have the following format: "What effect do you expect [*parameter configuration of the VE*] has on you?". Each question is answered with "Anxious", "Not anxious", or "I don't know". Examples are:

- "What effect do you expect driving through a tunnel has on you?"
- "What effect do you expect standing in a crowded elevator has on you?"
- "What effect do you expect hearing your own heartbeat has on you?"

The context information about the progress during exercises is collected through exposure logs which are commonly used in ET [6]. In these exposure logs, the patient answers a few questions after each exercise:

- "How afraid are you immediately after the exposure exercise?" (SUDS)
- "What was your peak anxiety level during the exposure exercise?" (SUDS)
- "Did that what you feared the most actually happen during the exposure exercise?" [Yes/No]

These questions have been designed to precisely extract the information needed to evaluate the RW model used by the adaptation algorithm.

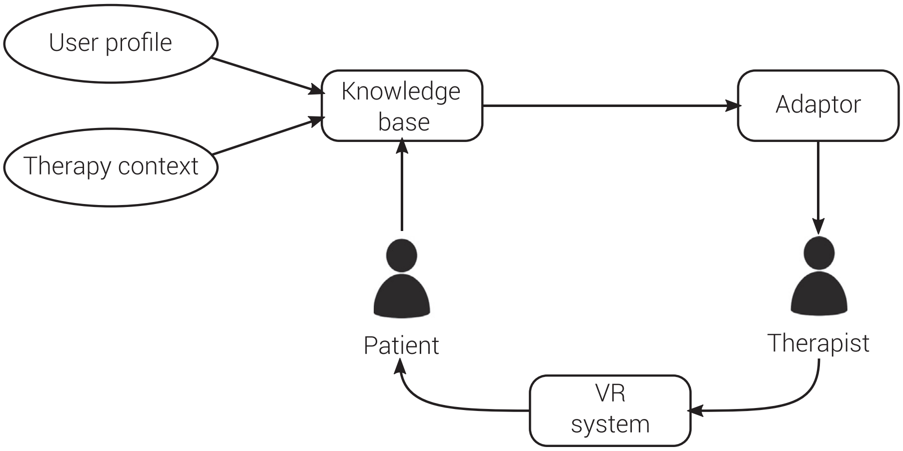


Fig. 1. Overview of the adaption system for VRET.

3.2.2. Ontology

Data in itself has little meaning. Only when adding context about the data, the information it holds becomes clear. Expert knowledge is required to add and interpret the context. The merger of these types of data and knowledge can ideally be performed through Semantic Web technologies [26], i.e. semantic ontologies, which provide formal descriptions of concepts, their properties and relationships between those concepts. These ontologies are built with domain experts through a co-creation approach [27]. Instances of the defined concepts are created from raw data, thereby mapping the data on the ontology. These are called individuals.

For this research, a new ontology has been developed specifically for VRET. To the authors' knowledge, no ontologies yet exist that describe the domain of CBT in general and VRET in specific. The ontology is split up into three logical layers, each modelling a different level of specificity. The top layer contains a general upper ontology, which defines concepts relevant to ET in general. The layer below consists of a VRET-ontology, which extends the upper ontology with concepts specific for ET in VR. The bottom layer contains several ontologies, each corresponding to a specific VE. The presented ontologies in each layer are publicly accessible from our repository[3] and are discussed in more detail below.

The upper ontology is modelled around the concept of hypotheses about the patient's fear and is shown in Fig. 2. These hypotheses model a relationship between a set of fear stimuli and responses, e.g. a tiny elevator with no windows and no alarm button elicits high anxiety in the patient. A hypothesis is tested through an exposure exercise. Doing the exercise reveals the actual response of the user. One target hypothesis is defined. It is a translation of one of the patient's goals into a configuration for the system. The *Adaptor* will generate intermediate hypotheses based on this target hypothesis that is used for new exercises. The ontology models both internal stimuli and external stimuli. These are properties of the environment or events happening during the exercise. The patient's responses to the exercise can be modelled as either behaviour characteristics, e.g. avoidance, panic and freezing up, or a SUDS. Fig. 2 shows some examples of individuals created for some of these classes in green. This represents instantiations of these concepts based on input data.

The VRET-ontology extends the upper ontology by providing VRET specific definitions for general concepts and introducing some new ones. The ontology is graphically presented in Fig. 3. A *VRScenario* embodies the hypothesis in VR. Therefore, it is a subtype of *Hypothesis*. A scenario is an entire configuration of a VE with all its properties and events that can occur. *Exercise*, *Stimulus* and *EnvironmentProperty* from the upper ontology have a VRET equivalent prefixed by the acronym "*VR*". The equivalents of *InternalEvent* and *ExternalEvent* are *VRInteroceptiveStimulus* and *VREvent*, respectively. These sub-classes are needed and used in this system because potentially other ontologies on the same level as the VRET-ontology, e.g. an ontology for in vivo ET, could extend these super-classes as well. In addition, the VRET-ontology contains a classification for stimuli based on the datatype of the stimulus parameter. The *Adaptor* requires this classification to query the *Knowledge base* about the stimuli present in each VE and how they can be configured. The main two classes are *VRDiscreteStimulus* for stimuli with discrete values and *VRContinuousStimulus* for stimuli with an infinite range of values. Both have some subtypes as well.

The ontologies in the lower layer depend entirely on the design of the VEs. They classify the stimuli used in each VE based on the types defined in the VRET-ontology. In essence, each ontology in the lower layer describes one type of VE and corresponds to one specific phobia to be treated with the system. Through this approach, the system is agnostic to the specific phobias it can treat. Thus, to support more types of VE, only the *Knowledge base* needs to be extended with a description of those VEs, while the rest of the system remains untouched. The adaptation algorithm is build to work with the classes in the upper layers. Thus, through inheritance, the adaptation algorithm works with any VE ontology that extends the upper layers with sub-classes. Fig. 4 shows an example of an ontology for a VE of a driving car. The concepts of the VRET-ontology, shown in grey, are extended to describe the specific attributes of the car VE.

3.2.3. Semantic reasoning

Semantic reasoning allows deriving logical consequences from the data and knowledge inside a *Knowledge base* [28,29]. There are two types of information that can be derived. On the one hand, ontology reasoning is used to infer to which classes an individual belongs. On the other hand, user-defined reasoning extracts high-level insights from low-level data. This is called user-defined reasoning, as the rules which dictate the reasoner are formulated by the end-user. In other words, the domain experts' knowledge is translated into rules such that the computer system can replicate part of its thought process.

The *Knowledge base* contains two user-defined rules, but it can easily be extended with more rules or more complex rules. The first rule is a simple rule used to define the property *has-*

[3] https://github.com/IBCNServices/PATRONUS-ontology.

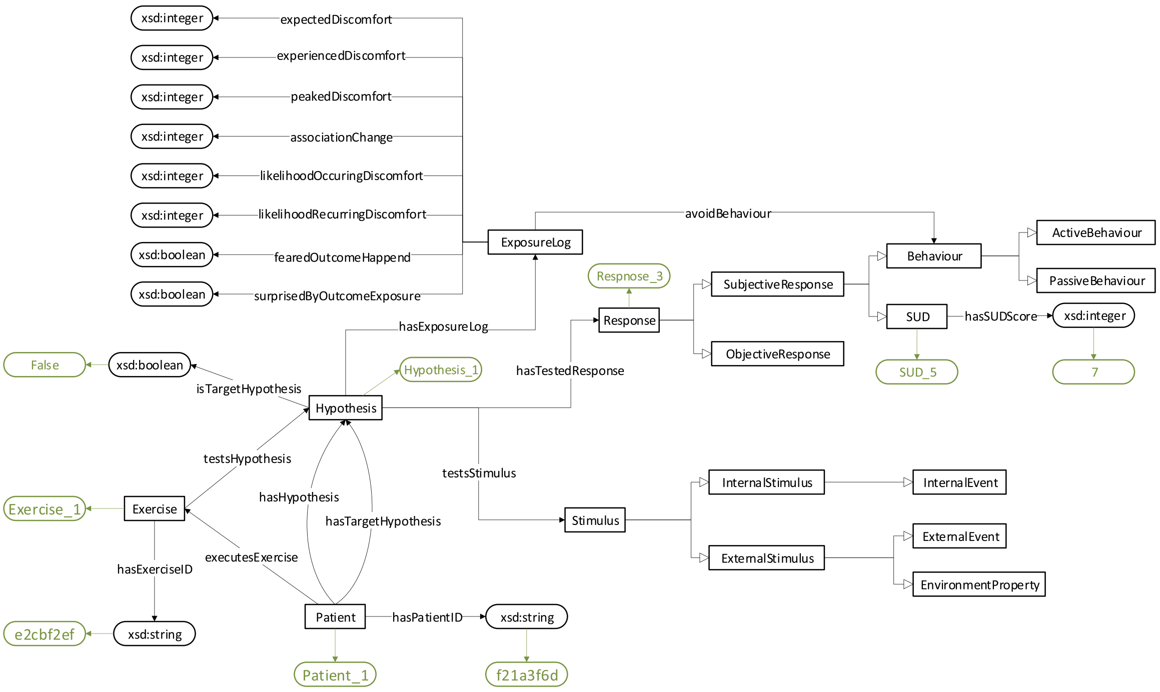


Fig. 2. This graphical presentation of the upper ontology shows the concepts (black boxes) and the relations between them (black arrows). The green boxes represent examples of individuals created for the corresponding classes.

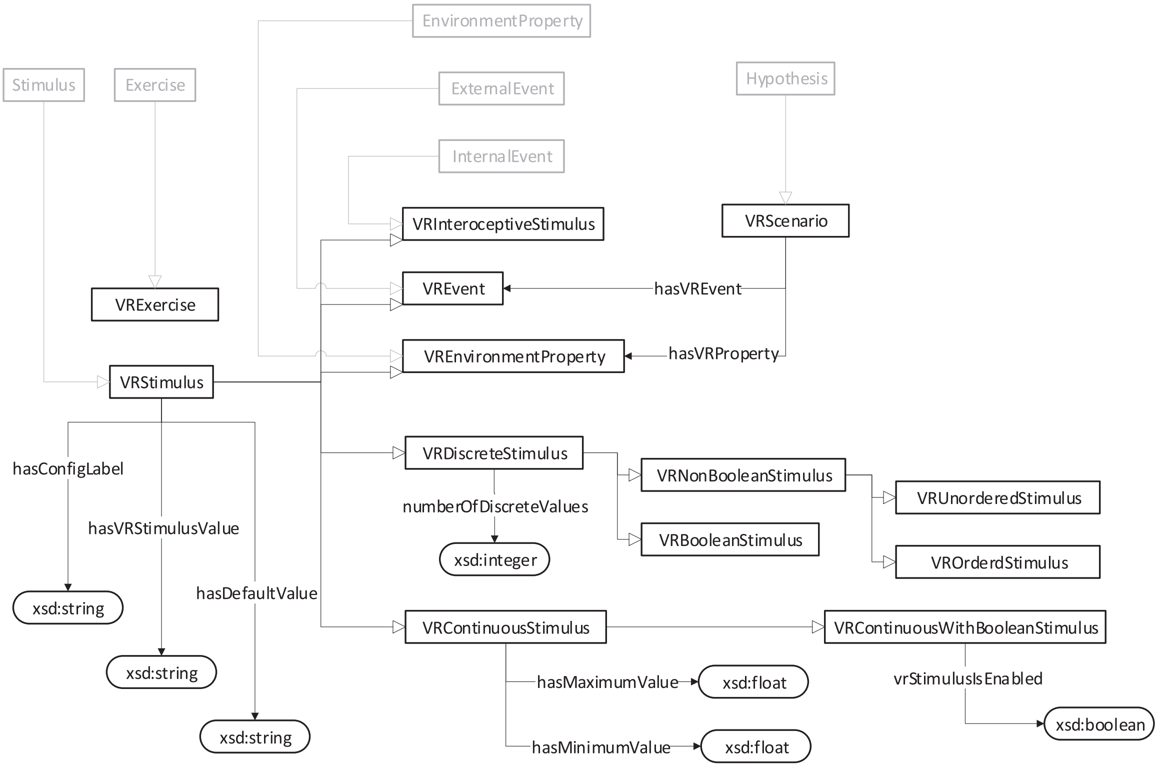


Fig. 3. The concepts and relations of the VRET-ontology that extend the concepts of the upper ontology (grey).

TargetHypothesis. This relation holds when a patient has some hypothesis (*hasHypothesis*), and that hypothesis is marked as the target hypothesis (*isTargetHypothesis*). The second rule calculates the value of the association change defined by the RW model, which is needed in the *Adaptor* and is discussed in more detail in the next section. This value can be calculated from data collected through the exposure log. This is an excellent example of how the *Knowledge base* contains expert knowledge. The *Knowledge base* knows how to calculate the association change based on the available data.

3.3. Adaptor

The *Adaptor* is the component that contains the actual adaptation algorithm for personalising the environment. It collects the required data from the *Knowledge base* and provides four sug-

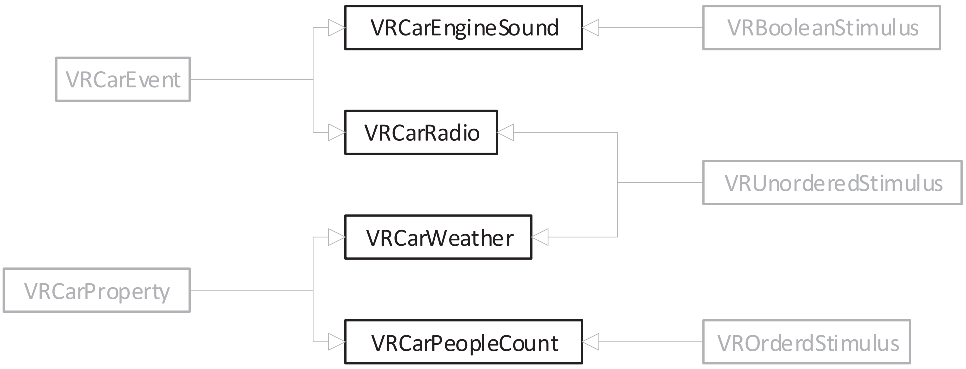


Fig. 4. An example of a small ontology that describes a VE of a driving car. The environment has four parameters that can be configured (CS). They extend the VRET-ontology concepts (grey).

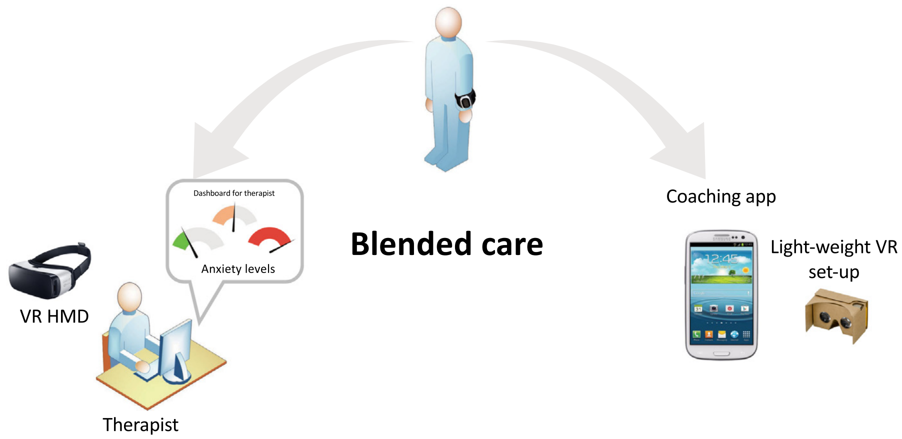


Fig. 5. The PATRONUS system is a patient-centred blended care solution for anxiety treatment through VRET with the psychotherapist and at home.

gested configurations for the following exercises to the psychotherapist. The choice for four configurations is primarily pragmatic: it creates sufficiently distinct environments while keeping the choice for a specific environment manageable for the psychotherapist. The *Adaptor* employs the theory and model proposed by Rescorla and Wagner [23] for calculating the explicit configurations. The mechanisms of the algorithm are discussed in detail in Section 4.1

3.4. Proof of concept

The VE for which the adaptation algorithm is generating suggestions are part of the *VR system*. In this work, a proof of concept application for VRET has been developed which fulfils the role of the *VR system*. This proof on concept application is called PATRONUS[4] and is the product of an interdisciplinary collaboration between computer science engineers, user researchers, IT experts and psychotherapists. The proof of concept application with which the adaptation algorithm is interacting is further discussed in this subsection. Special attention is given to the virtual environments of the proof of concept.

3.4.1. PATRONUS system

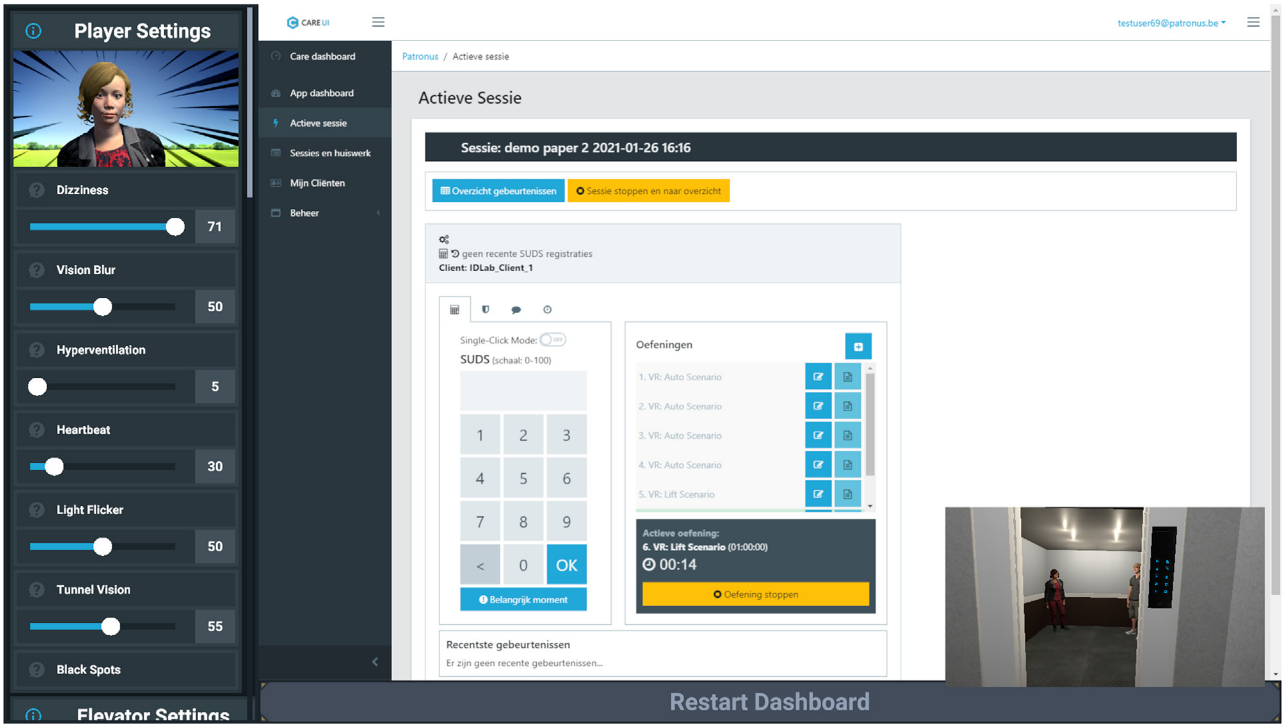
The PATRONUS system aims to create a patient-centric blended healthcare solution for anxiety treatment through ET. The approach consists of therapy at the psychotherapist's office and longitudinal follow-ups through homework exercises on a mobile coaching app, as presented in Fig. 5. During the face-to-face sessions, the patient receives ET through the use of VR. The VE are personalised to the specific needs of the patient. In between sessions, the patient can continue exposing themselves to anxiety-inducing environments through homework exercises on their mobile phone. The patient always plans the homework exercises in consultation with the psychotherapist. These homework exercises are instructions to experience real-life situations or VRET exercises similar to those performed during the face-to-face sessions.

At the centre of the PATRONUS system is a dashboard application, as shown in Fig. 6. Through this dashboard, the psychotherapist can record information about the patient, configure and start VR exercises for ET and create homework exercises that synchronise with a mobile application on the patient's smartphone. The psychotherapist can also consult the progress of the patient from the dashboard. When configuring a VR exposure exercise, the psychotherapist can either manually configure a new environment or have the adaptation algorithm generate four suggestions for the patient.

3.4.2. The virtual environments

The VEs are designed to make them easily adaptable, either manually or through an algorithm. Each environment has some base elements and characteristics which cannot be changed. These define the overall purpose of the environment. For each environment, a set of parameters is defined, which further refines the elements and characteristics of the environment. These parameters are configurable and allow the environment to be adapted. In this implementation, each parameter is considered a potential stimulus that elicits fear, i.e. CS. The following sections discuss specific configurable parameters.

[4] https://www.imec-int.com/en/what-we-offer/research-portfolio/patronus.



Dizziness 71

Vision Blur 50

Hyperventilation 5

Heartbeat 30

Light Flicker 50

Tunnel Vision 55

Black Spots

Actieve Sessie

Sessie: demo paper 2 2021-01-26 16:16

geen recente SUDS registraties
Client: IDLab_Client_1

Single-Click Mode: OFF
SUDS (schaal: 0-100)

Oefeningen
1. VR: Auto Scenario
2. VR: Auto Scenario
3. VR: Auto Scenario
4. VR: Auto Scenario
5. VR: Lift Scenario

Actieve oefening:
6. VR: Lift Scenario (01:00:00)
00:14

Recentste gebeurtenissen
Er zijn geen recente gebeurtenissen...

Fig. 6. The dashboard application is the interface for the psychotherapists to use the entire PATRONUS system.

The PATRONUS system incorporates two distinct environments. There is an elevator environment and a car environment. Both environments can be used for people with panic disorder and claustrophobia. The car environment also focuses on fear of driving.

Apart from the environment-specific parameters, a small set of general adaptation parameters are provided. These are the so-called *interoceptive* parameters. These do not change anything about the environment itself. They are designed to elicit a specific interoceptive sensation in the patient by modifying the perception of the environment.

The *elevator* and *car* environments, and *interoceptive* parameters are discussed in more detail in the following sections.

Elevator environment

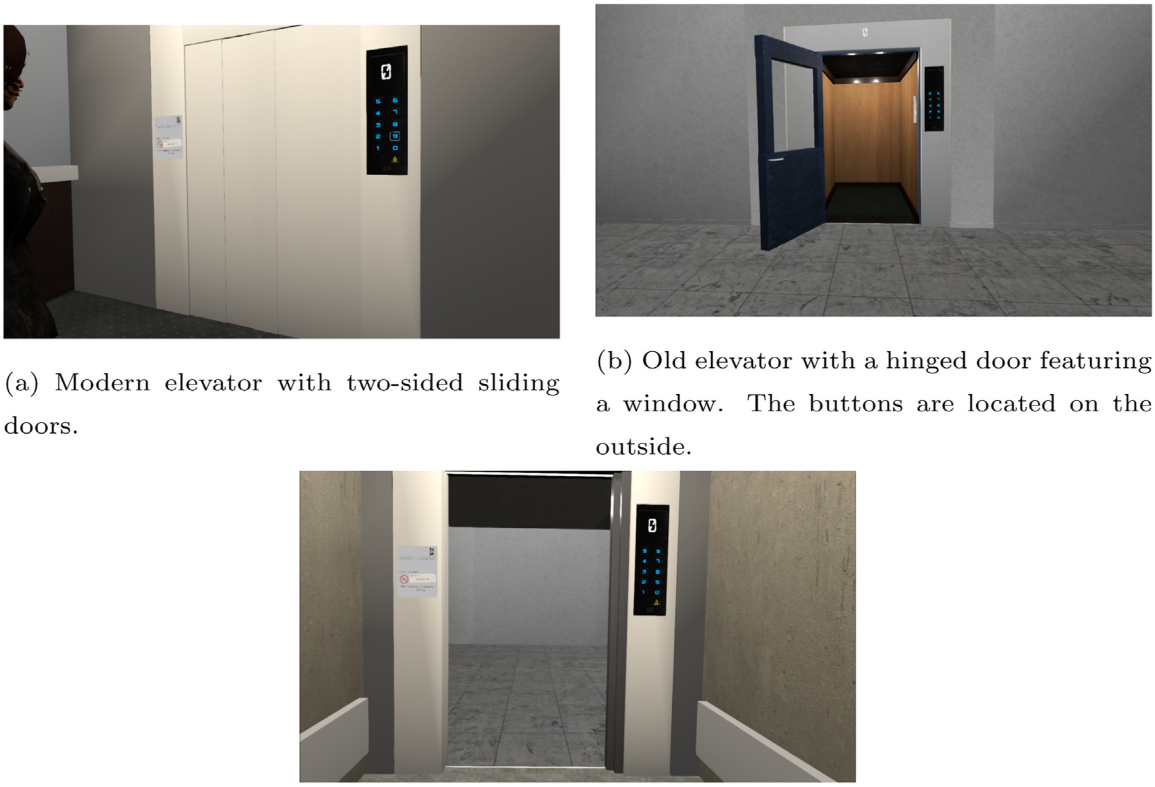
The user can interact with the elevator environment by walking around freely and pressing the buttons in the elevator. In this scenario, the user controls what happens as they decide to enter the elevator or go to a specific floor. Interaction happens through hand-held controllers, which are represented by hands in the VE. The look and feel of the elevator are crucial because this can influence the patient's anxiety level. Therefore, three base types of elevators are supported, a modern-looking elevator (Fig. 7a), an old elevator (Fig. 7b) and a service elevator (Fig. 7c). They can then be further tailored by choosing a type of door from a one-sided sliding door (Fig. 7c), a two-sided sliding door (Fig. 7a), or a hinged door (Fig. 7b), adding windows (Fig. 7b), and changing the size of the elevator. Also, more subtle elements can be configured, such as whether the floor buttons are inside or outside the elevator, whether an alarm button is present or how much noise the elevator makes. The number of people in the elevator can be changed to simulate a crowded space. To further enhance the sense of presence, subtle visual cues are incorporated for the elevator's speed through moving light streaks around the doors and the sensation of the elevator starting and stopping through some camera shake. Lastly, three parameters simulate defects in the elevator as these are essential stimuli for some patients. Random shaking of the elevator is simulated through camera shaking, the lights in the cabin can dim at random moments, and the elevator could get stuck while moving between floors.

Car environment

The patient is seated in a virtual car on the freeway. There is little interaction possible from the user with the VE. The user can only look around while being submitted to what is going on in the environment. To provide a better sense of presence in the environment, the patient holds two controllers matched to hands on the steering wheel. In this car environment, many parameters can be altered to increase or decrease the patient's level of anxiety. The first is the type of freeway: an open freeway (Fig. 8a), a tunnel (Fig. 8b) or a bridge (Fig. 8c). The traffic on the road is configurable as well: the amount of traffic, the speed level, the presence of motorcycles (Fig. 8b), and the lane in which the car is driving. Also, the time of day and the weather conditions can be altered. In this VE, it can be nighttime or daytime, and it can be sunny or rainy and clear or foggy. A set of parameters focuses on the situation in the car itself and discerns between visual and auditive stimuli. For the visual parameters, the patient can be seated behind the steering wheel or sit in the passenger's seat in the front or the back. Additionally, the number of passengers in the car can also be modified. There is also an option for enabling a blinking warning light on the dashboard to signal a defect. Finally, four more parameters provide acoustic stimuli to increase or decrease the patient's level of anxiety, as some patients are more susceptible to sounds. The patient can hear a car horn, wailing sirens and engine sounds. Also, the radio can be turned on and off, either broadcasting traffic information or instrumental music.

Interoceptive parameters

Dizziness is achieved by applying motion on the camera view of the virtual environment, which causes a misalignment between the physical position of the user's head and the position in the virtual environment. A filter on the visual feed applies a blur to simulate a sudden vision blur for the user. Tunnel vision is also a filter applied on the visual feed. This filter narrows down the field of vision resulting in a small area through which can be seen, while

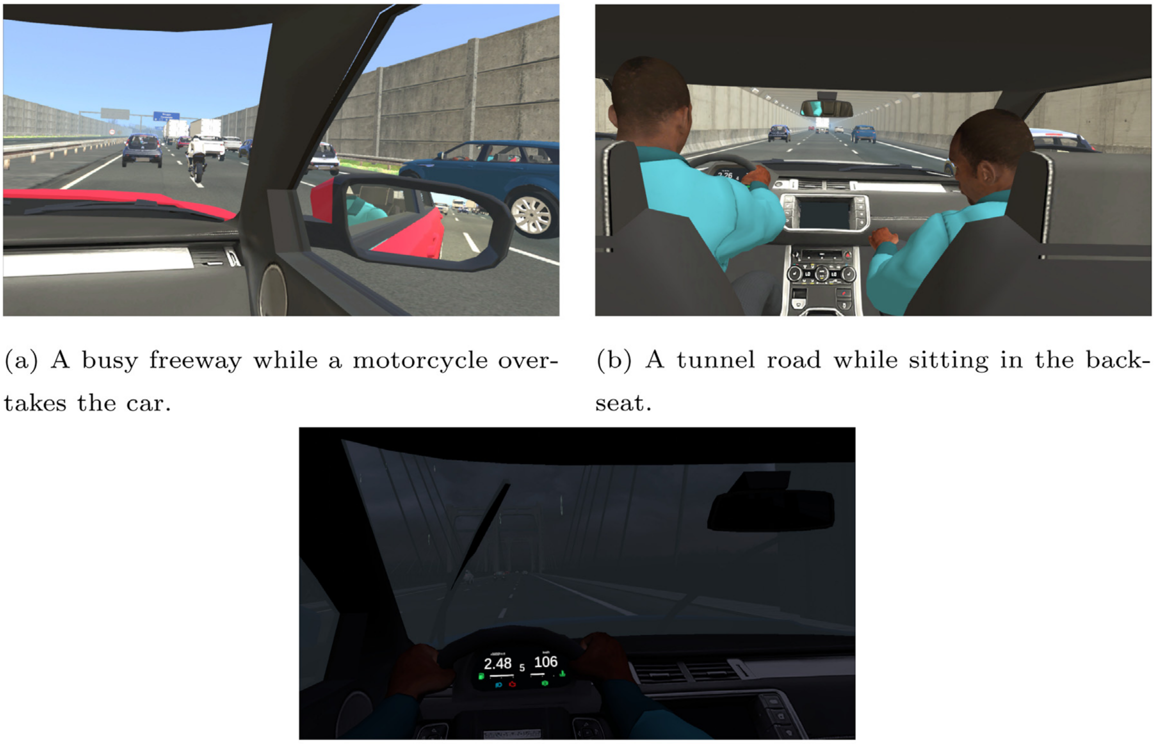


(a) Modern elevator with two-sided sliding doors.

(b) Old elevator with a hinged door featuring a window. The buttons are located on the outside.

(c) Service elevator with a one-sided sliding door.

Fig. 7. The look and feel of the elevator depends on the combination of many parameters such as the elevator interior, the door type, and the buttons' position.

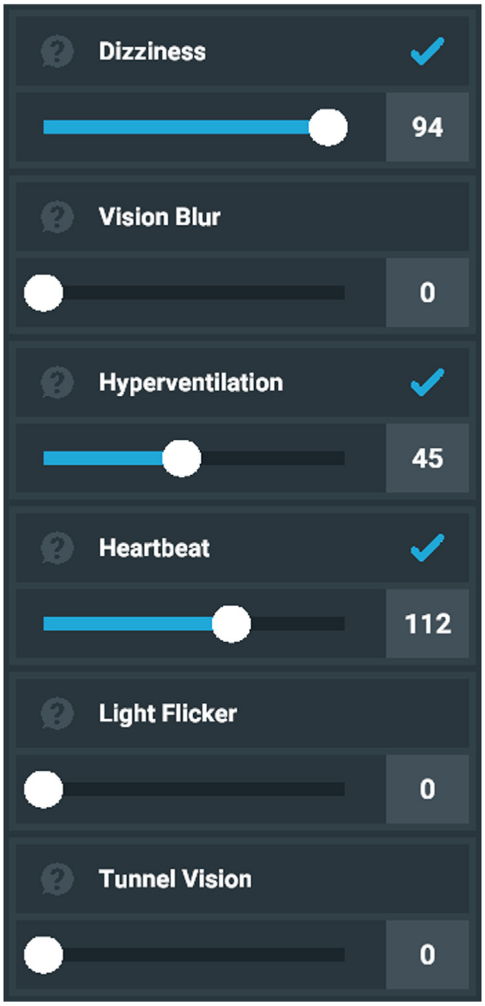


(a) A busy freeway while a motorcycle overtakes the car.

(b) A tunnel road while sitting in the backseat.

(c) A road across a bridge by night.

Fig. 8. The combination of parameters such as the road type, the seating position and the time of day influence the representation of the car environment.



Dizziness ✓

94

Vision Blur

0

Hyperventilation ✓

45

Heartbeat ✓

112

Light Flicker

0

Tunnel Vision

0

Fig. 9. The interoceptive stimuli are adjustable in the dashboard view during an exposure exercise.

the rest is black. Another effect causes light flickers which create short bursts in which the view turns completely white. These flickers happen at random points in time. The last two are based on an acoustic cue. The sound of a heartbeat can be simulated. This creates the illusion that the patients hear their heartbeat. Lastly, intentionally causing hyperventilation is a common exercise in ET and elicits interoceptive sensations. Hyperventilation can be triggered by letting the patient breathe deeply in and out at a steady and fast pace. In the VE, a beeping sound indicates the pace. All interoceptive parameters can be set in the dashboard, see Fig. 9, through a toggle button and a slider indicating the intensity.

4. Results

The primary result presented by this paper is the prototype of an adaptation algorithm for VRET. The algorithm creates four suggestions for possible new configurations of VE based on the needs of the patient by relying on the theoretical foundation provided by the RW model and data of the patient and prior exercises. This prototype is further described in this section. Furthermore, a scalability test on the execution time of the algorithm is performed in the next part of the section. Finally, the suggestions resulting from this algorithm are compared to prior manual configuration of two psychotherapists from three anonymised clinical vignettes.

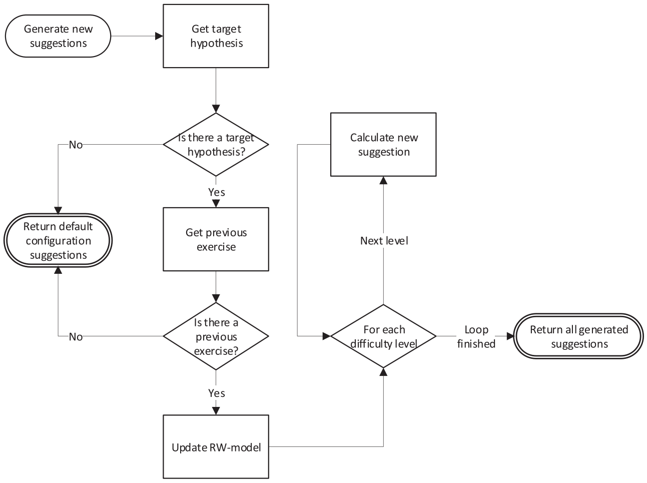

Generate new suggestions
Get target hypothesis
Is there a target hypothesis?
No
Yes
Return default configuration suggestions
Get previous exercise
Is there a previous exercise?
No
Yes
Update RW-model
Calculate new suggestion
Next level
For each difficulty level
Loop finished
Return all generated suggestions

Fig. 10. Flow diagram of the adaptive algorithm based on the Rescorla–Wagner model.

4.1. Adaptation algorithm

The adaptation algorithm relies on a plethora of heterogenous data of the patient, its history and the available VE for the VRET. This information is consolidated in a knowledge base as discussed in Section 3. This knowledge base is critical for the adaptation algorithm. Fig. 10 presents the process of the adaptation algorithm to generated the four suggestions based on the information it has available.

The process starts by querying the knowledge base for the target hypothesis and a base hypothesis. The target hypothesis has been constructed from the VR parameter questionnaire. The base hypothesis corresponds to the hypothesis that was used in the last exercise. It is called the base hypothesis or configuration because it forms the starting point for the suggestions of the next exercise, i.e. for each new exercise, the configuration of the previous exercise is altered. If no target or base hypothesis exists, default suggestions will be provided to the psychotherapist as there is not enough data available to construct sensible suggestions. In this case, the process terminates early.

The next step is updating the RW model [23]. The following two formulas describe the model.

$$\Delta V_X^{n+1} = \alpha_X \beta \left(\lambda - \sum_{s \in S} V_s \right), \tag{1}$$

$$V_X^{n+1} = V_X^n + \Delta V_X^{n+1}. \tag{2}$$

Eq. (1) models the change in association, ΔV_X for trial $n+1$ between the CS X and the US whereas, Eq. (2) expresses the total association strength between CS X and the US after $n+1$ trials. In Eq. (1), α_X is the salience of X, β is the rate parameter for the US, λ is the maximum association possible for X, and S is the collection of all stimuli presented during the trial.

The knowledge base provides the information needed to perform the update step. Specifically, it provides estimations for the values of α_X, β, λ and V_{tot}. The theory of Rescorla and Wagner states that α_X and β need to remain the same for each exercise. The parameter β denotes the rate at which a patient forms new associations. Therefore, the value of β is associated with the number of exposure exercises needed before the patient shows a reduced SUDS for a particular stimulus. For the construction of this algorithm, we assume a value of 0.2 for the product of α_X and β in Eq. (1). Finding the optimal values for α_X and β is outside the scope of this research. This work uses an estimated value as a default. The value for λ can be extracted from the exposure log directly. The answer to the question "Did that what you feared the most actually happen during the exposure exercise?" is either *yes* or *no*, which translates to a value of 1 or 0, respectively, for λ. Lastly, V_{tot} is the fear association of the patient for the combination of all stimuli in the current exercise. This value can also be directly obtained from the exposure log. The observed peak discomfort provides this value.

With the updated RW model, the four suggestions can be calculated. Each suggestion has an estimated difficulty as a number between 0 and 1 denoted *difficulty_level*. An even distribution of difficulties ranging from 0.5 to 0.95 is chosen, resulting in a diverse range of suggestions that are not too simple but still challenge the patient. Adjusting the difficulty level for each of the four suggestions can result in more diverse configurations if needed.

The difficulty value influences the number of stimuli for which a new value is chosen and the value of the new stimulus itself. As mentioned before, a new configuration is created by changing the configuration of the previous exercise, which was called the base hypothesis. Each stimulus from the base configuration changes with a probability p, as defined in Eq. (3), and stays the same with a probability of $1 - p$. Therefore, a higher difficulty value leads to more stimuli with a new value, while a higher association with the previous exercise leads to fewer stimuli with a new value.

$$p = difficulty\_level * (1 - V_{tot}) \tag{3}$$

The new value of a stimulus is based on s, according to Eq. (4). This value indicates the distance between the value of the stimulus and the target value of that stimulus. A value of 1 results in precisely the target value, while 0 results in the value furthest away from the target.

$$s = difficulty\_level / V_X \tag{4}$$

The target value t is the value given to a stimulus in the target hypothesis. The method for calculating the exact new value u of a stimulus depends on the datatype of its parameter and the target value of the stimulus. The four supported types are parameters with boolean values, parameters with an ordered set of possible values, parameters with an unordered set of possible values and parameters with continuous values.

For stimuli with boolean parameters, the value of u is calculated as Eq. (5) formulates.

$$u = \begin{cases} true & s > 0.5 \\ false & s \leq 0.5 \end{cases} \tag{5}$$

If the parameter can take a value from a set of ordered options, s is first transformed to match the range between 0 and t, and then the value is rounded to the nearest integer number. This value then represents the selected option from the set as Eq. (6) states.

$$u = \text{round}(s \cdot t) \tag{6}$$

When the set of options is unordered, the value of s has little significance. Therefore, the algorithm always selects the exact target value.

Finally, for continuous values, the new setting is calculated by multiplying s by the target value and adjusting for the minimum value of the range as stated in Eq. (7).

$$u = (s \cdot (t - value_{min})) + value_{min} \tag{7}$$

As mentioned before, the value of s indicates the relative difference between t and u. When $s = 0$, the difference should be maximal. In these formulas, the assumption was made that the target value is always close to 1, indicating a high intensity. Therefore, the opposite would be 0, indicating a low intensity. In practice, this assumption does not always hold as the target could also be exposed to a low-intensity stimulus because it elicits fear. If the target is close to 0, the new value of the stimulus can be calculated using the previous formulas as $1 - u$.

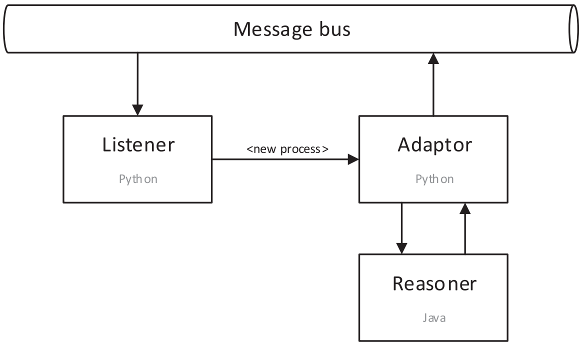


Fig. 11. The implementation overview of the different components of the adaptation system and how the message bus is used for communication with any other part of the system.

4.2. Implementation

The system in which the adaptation algorithm gets integrated is responsible for collecting data from different sources (e.g., manual input, questionnaires, previous exercises) and rendering the VE based on the configuration. Therefore, the algorithm and the knowledge base need to interact with the system to use the collected data and instruct it how to configure the VEs. The technical implementation details of the interaction with the system are discussed here.

Fig. 11 illustrates the three components that enable the integration of the adaptation algorithm. These components communicate with any other part of the system through a message bus. A continuously running listener process written in Python monitors the bus for new incoming messages for the adaptions system. The message on the bus contains an identifier indicating for which patient the adaptations should be calculated. The message also contains the locations of the relevant data for that patient. For every incoming message, a new process is started which performs the calculations for the adaptions. The control flow is driven by the *Adaptor* component as depicted in Fig. 1, which is also implemented in Python and interacts with a Java-based semantic reasoner called Pellet [30]. Once the suggestions are generated, these are pushed back on the message bus for the following components to handle them.

4.2.1. Ontology representation and reasoning

Ontologies need a formal description for a computer system to be able to interpret it. The ontology description uses OWL 2 [31], the most recent version of Web Ontology Language (OWL). OWL is a recommendation of the W3C. The *Adaptor* uses OWL API [32] for handling the OWL ontologies at runtime, while SPARQL is used to query the information from the ontologies. The Protégé [33] editor software was used to create the OWL ontologies. Listing A.1 presents a fragment of the VRET-ontology using OWL.

The user-defined rules also need a formal definition for the reasoner to interpret them. The Semantic Web Rule Language (SWRL) [34,35] can be used for this. It makes it possible to describe logical rules for OWL ontologies. An example of a SWRL rule is as follows:

$$ExposureLog(?e) \wedge experiencedDiscomfort(?e, ?x)$$
$$\wedge\ expectedDiscomfort(?e, ?y) \wedge swrlb{:}subtract(?z, ?x, ?y)$$
$$\longrightarrow associationChange(?e, ?z)$$

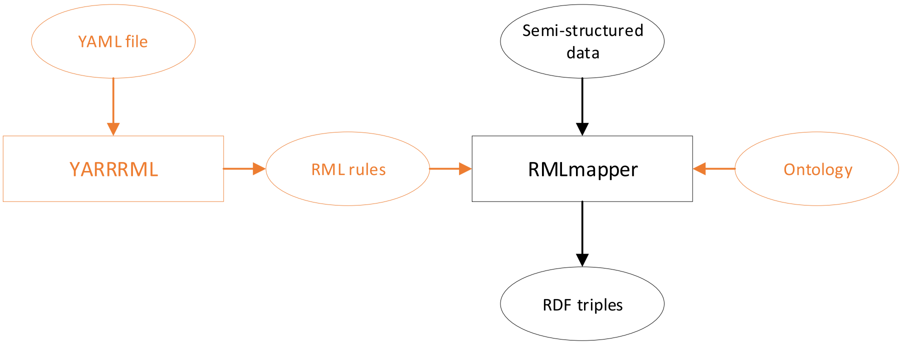


Fig. 12. Overview of the data mapping process showing the two processor units (squares) and the data objects (ovals) that flow through it. The orange process is only executed once before execution time. Similarly, the orange data objects are created once at design time. The black process is executed repeatedly at runtime resulting in new data objects.

4.2.2. *Data mapping*

Raw incoming data is mapped onto the ontology. This means creating instances of the concepts defined in the ontology and assigning relationships between these instances based on the incoming data. Fig. 12 depicts this data mapping process. The raw data enters the adaptation system in a semi-structured format, namely JSON. The result of this data mapping is a set of Resource Description Framework(RDF) triples. A triple consists of a subject, a predicate and an object, e.g. `patient` (subject) `hasHypothesis` (predicate) `hypothesis` (object). In this way, every relation of every instance can be defined. The data in semi-structured JSON format can be automatically mapped using a tool called *RMLmapper*, which uses the RDF Mapping Language(RML) [36] to define rules for mapping from one structure to another.

The RML syntax is rather complex. Therefore, an additional tool is used for generating these rules from a much simpler syntax. YARRRML [37] is an application that accepts mapping rules in YAML syntax and outputs the corresponding RML rules.

Constructing the RML rules using YARRRML is only performed once during implementation. The mapping into RDF triples using RMLmapper is performed for every new data entering the system. This is illustrated by the orange and black components, respectively, in Fig. 12.

To illustrate this process, the following example is presented. The input is the JSON-file in Listing 1, which contains only an ID of the patient. The input data is mapped according to the mapping rules defined in Listing A.2. These rules are in YAML syntax. Therefore, they are first translated into RML syntax, which results in the rules shown in Listing A.3. The output of the process for this example is two triples shown in Listing 2. The first triple states that there is some entity which is of type *Patient*. The second triple indicates that this new entity has the property *hasPatientID*, which provides it with an ID.

```
{
    "ClientId": "f21a3f6d"
}
```

Listing 1. This is an example of a possible input JSON file. It only contains an ID of a patient.

4.3. *Evaluation of the execution time*

It is hardly feasible to collect large amounts of data from real-life experiments with actual patients for evaluating computational complexity and performance, nor would it be ethically appropriate to do so with a first protoype of the algorithm. Therefore, these tests ran in a simulated environment with randomised data. Specifically, the execution time of the algorithm for an increasing amount of input data is evaluated. The amount of input data directly correlates to the number of previous exercises. Therefore, the tests are evaluated as a function of the number of previous exercises directly. The content of the processed data does not influence the execution time, thus, randomised data is used.

The number of input exercises for the simulation range from 1 to 50. The execution metrics presented are averaged over 50 repetitions. All simulations ran on a 2,4 GHz Dual-Core Intel Core i5 with 8GB of memory. The code is written in Python without any parallel computing.

The execution of the entire algorithm is divided into five steps, the *pre-processing* step, the *RML-mapping* step, the *reasoning* step, the *updating* step of the RW model and the *suggestion generation* step. The execution time for each step is reported individually. A category is added for *other* processing, including downloading data and querying the knowledge base to ensure the sum of all steps equals the overall execution time. Fig. 13 shows the stack plot of the execution time as a function of the number of previous exercises. Table 1 presents the exact values of each step's average execution time and standard deviation for 1, 10, 20, 40 and 50 previous exercises.

Every step of the algorithm scales linearly with the number of previous exercises. The *pre-processing* step and the *other* processing take up the least amount of time and are almost negligible compared to the other four classes. For an input of 50 exercises, these four classes account for 99.2% of the total execution time. In total, 14.07% of the execution time is spent on data mapping, 38.54% is spent on reasoning, 6.62% is spent on updating the RW model, and 39.98% is spent on generating suggestions. All steps together scale at a rate of 0.27 s per additional previous exercise. The standard deviations are reasonably low for each step. The reasoning step does show the most deviation at max 953 ms.

Fig. 14 compares the slope of each of these four dominating computation steps against each other. The *RW updating* step is the least affected by the amount of input data, while the *reasoning* and *suggestion generation* steps are affected the most. The *reasoning* step has a very high initial overhead. However, it has a relatively small slope. After approximately 39 exercises, the execution time of the *generation* step will dominate over that of the *reasoning* step. The high initial overhead of the reasoner is due to the slow startup time of the reasoning process, which is performed at the

```
base:patient_f21a3f6d a uet:Patient.
base:patient_f21a3f6d uet:hasPatientID "f21a3f6d"^^xsd:string.
```

Listing 2. The output of the mapping process are two triples.

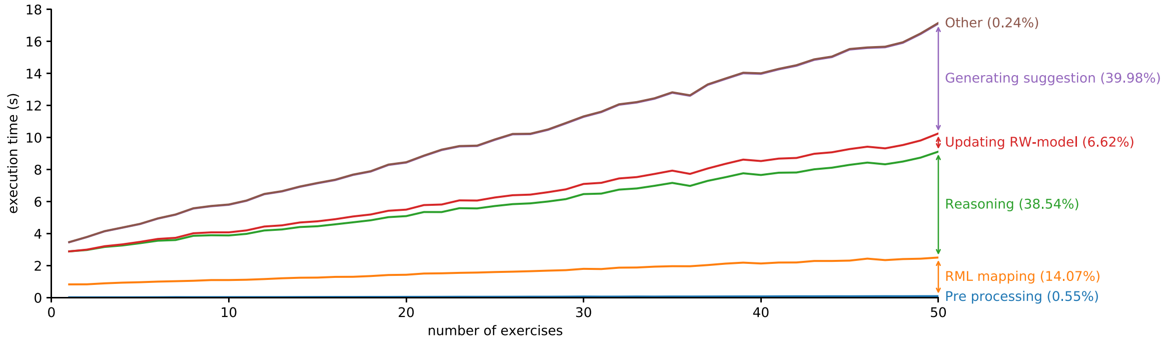


Fig. 13. The majority of the total execution time of the algorithm, namely, 99%, is accounted for by four steps of the algorithm: RML-mapping step (14%), reasoning step (39%), updating step of the RW model (7%) and the suggestion generation step (40%) for 50 exercises. Pre-processing and the "other processing" category are almost negligible, making up the final 1%.

Table 1
Each test condition is executed 50 times, the average execution time and standard deviations for each step are presented in seconds for 1, 10, 20, 30, 40 and 50 previous exercises.

Number of exercises	Pre processing		RML mapping		Reasoning		Updating RW model		Generating suggestion		Other	
	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	0.023	0.003	0.806	0.104	2.060	0.245	0.023	0.003	0.568	0.047	0.012	0.004
10	0.035	0.005	1.066	0.052	2.785	0.126	0.191	0.036	1.720	0.143	0.021	0.001
20	0.050	0.009	1.380	0.081	3.661	0.499	0.405	0.073	2.934	0.285	0.027	0.008
30	0.071	0.048	1.730	0.265	4.666	0.499	0.633	0.103	4.186	0.335	0.033	0.016
40	0.080	0.004	2.054	0.098	5.525	0.685	0.870	0.137	5.441	0.468	0.038	0.018
50	0.095	0.006	2.412	0.133	6.610	0.953	1.135	0.171	6.857	0.553	0.041	0.016

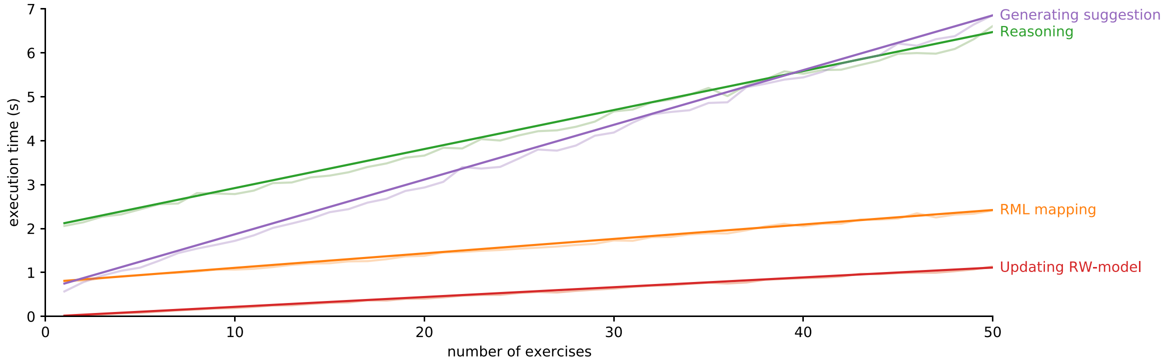


Fig. 14. For each step, the average execution times (faded colours) are shown to which a linear function is fit (bright colours). Every step of the algorithm scales approximately linearly with the number of previous exercises. However, the reasoning and generation steps are the most affected by the amount of input data, as the execution time increases significantly more than the other two dominating computation steps with increased input data.

start of each simulation run. OWL-DL reasoning is a slow process that scales poorly for large amounts of data [38].

In the presented system, the *Adaptor* component lacks the capability of persistent storage. This has an impact on the data *mapping* step and the calculations for updating the RW model. Specifically, intermediate results of both steps cannot be stored. By introducing persistent storage and storing intermediate results, these two steps can be optimised. Since only new information from the last exercise needs processing, the *mapping* step would reduce to mapping the data of the last exercises. The execution time would be constant at approximately 800ms, as can be deduced from Fig. 14. The same applies to the update step of the RW model. The time needed for this step would be constant at approximately 20ms, as shown in Fig. 13. Assuming these optimisations are applied, the execution times can be estimated to be reduced by 15.8% for 50 previous exercises. This results in a scaling factor of 0.22 s per previous exercises. Additionally, the reasoning process would benefit from storing intermediate results. The reasoning would be limited to data from new exercises by storing the materialised ontology for each patient. Therefore, the gain in execution time is more challenging to estimate and is therefore not included in the presented time estimation above.

4.4. Clinical vignettes

Through tests on data of clinical vignettes, the overall quality of the generated suggestions is evaluated. Specifically, VRET therapy sessions were held with clinical patients without the use of the adaptation algorithm. The data of these therapy sessions was collected as part of a prior study, for which the protocol was approved by the Medical Ethical Committee of Ghent University Hospital [EC/2019/0893]. During these sessions, three patients suffering from panic disorder or specific phobia – fear of driving diagnosed via a structured interview, Mini International Neuropsychiatric Interview (MINI) [39] – received ET with manually configured VR exposure exercises on the PATRONUS system. These patients had the following characteristics: Patient 1, male, 34 years, diagnosed with panic disorder; Patient 2, male, 68 years, diagnosed with specific phobia and fear of driving; Patient 3, female, 55 years, diagnosed with specific phobia. The two psychotherapists perform-

Table 2
The similarity of each generated suggestion compared to the used configurations. The values in bold indicate the highest similarity for each configuration. The values in italic indicate the choice of the psychotherapist during the post-study.

Exercise	Suggestions			
	1	2	3	4
1	0.53	***0.66***	0.53	0.50
2	*0.81*	**0.83**	0.62	0.82
3	0.77	0.78	0.77	**0.79**
4	**0.81**	0.79	0.78	0.77
5	0.58	**0.60**	0.55	*0.49*
6	**0.46**	**0.46**	*0.44*	0.38
7	**0.45**	**0.45**	0.40	*0.34*
8	***0.83***	0.79	0.75	0.81
9	**0.66**	0.65	0.60	0.62

ing the therapy are experienced in their domain and well familiar with the PATRONUS system (psychotherapist 1, 38 year old, psychotherapist 2, 31 years, with experience between 10 and 15 years of professional experience), both are IRB approved for participating in the study. This experience provides them with good knowledge of how to create the best VE for the patients manually. During a post-therapy evaluation, the psychotherapists were presented with multiple sets of four suggestions for each exercise they actually performed with the patient during the therapy sessions. In other words, each set of suggestions was calculated with the data which would be available at that time to simulate the actual setting.

On the one hand, these suggestions are compared with the manual configurations of the psychotherapists from the actual exercise. On the other hand, the psychotherapists were asked to indicate the most appropriate suggestions. This approach was chosen to prevent the algorithm from influencing the psychotherapist, thereby introducing bias into the results. All tests from this evaluation are applied to the car environment because none of the participating patients required therapy using the elevator environment. However, environment design should not significantly impact the results as the algorithm is environment agnostic. In other words, the algorithm should adapt the values of any VR parameter or combination of parameters.

The similarity of the two configurations increases with the number of parameters with the same value between these two. For this evaluation, this similarity is expressed as a number between 0 and 1, where a value of 0 means two configurations are as dissimilar as possible while a value of 1 denotes two identical configurations. Table 2 presents the similarity of the four suggestions generated by the adaptation algorithm against the manual configuration of each exercise. For each row, the suggestion with the highest similarity is indicated in bold. In most cases, the highest similarity is between 0.6 and 0.8. This result indicates that the suggestions are indeed close to what the psychotherapist would do. For some configurations, the similarity is the same compared to two different suggestions. That is because the actual configuration was equally similar to both, not because both suggestions are identical. This occurs for exercises that have relatively low similarity to all suggestions. It suggests that the presented suggestions are not in line with the psychotherapists approach during the therapy.

The results in Table 2 present the similarity scores of the environment parameters and interoceptive parameters combined for nine different exposure exercises covering three patients. Each exercise was assessed by one of the two psychotherapists. The tests show a significant difference in the average similarity for interoceptive parameters and environmental parameters. The similarity is 0.88 on average for the interoceptive parameters, while for the environmental parameters, it is 0.58. For specific suggested configurations, the similarity on the interoceptive parameters equals 1, meaning the configurations are identical. It is clear that appropriate personalised interoceptive parameter values are much easier to predict. However, if this increased accuracy is due to the smaller configuration space of these parameters compared to the environment parameters or better predictability of interoceptive parameters is unclear from the data.

The cells with a value in italic in Table 2 indicate the suggestions the psychotherapist assessed as the most appropriate during the post-therapy evaluation. As can be seen, for some exercises, the psychotherapist decided that none of the suggestions was good enough (Ex. 3, 4 and 9). For other exercises (Ex. 1 and 2), the choice of the psychotherapist matched the suggestion with the highest similarity, i.e. the values in bold. In all other cases, the choice of the psychotherapist during the post-study did not match with the highest similarity suggestion of the actual therapy sessions. However, in those cases, the difference in similarity to the actual configuration between the psychotherapist's choice and the closes matching suggestion is minor. On average, a difference of 0.09 in similarity, with a minimum of 0.02 and a maximum of 0.21, was obtained.

5. Discussion and conclusion

This work presents a prototype of an adaptation algorithm for personalised VE in VRET. A proof of concept application for VRET enabled the integration of the algorithm. Data from three clinical vignettes enabled the comparison of the manually configured environments and the generated suggestions. The results of this comparison indicate that the proposed system has merit, as therapists choose similar configurations to at least one generated suggestion. Indeed, the results suggest that the output of the adaptation algorithm is valuable to the experienced psychotherapists in this study. However, future studies are needed to discover if less experienced psychotherapists would benefit even more from the algorithm.

From a computational point of view, the tests showed that the system scales linearly with the number of previous exercises with a factor of 0.22 s per exercise. Although this is quite steep, the computation for 50 previous exercises is below 15 s. The domain experts estimate that the actual required number of VR exercises can vary but rarely exceed 50. Therefore, the execution time measured for 50 exercises can be considered an upper bound, which is acceptable for therapy application.

The relevance for potential application in clinical practice has also been a key focus throughout this work. Researching this relevance required a multidisciplinary mindset and resulted in continuous collaboration with psychotherapists. They provided a theoretically sound foundation for the resulting adaptation algorithm and implemented a proof of concept in practice. This helped to demonstrate the algorithm's potential better and pushed this research further than mere theoretical conceptualisation.

Nonetheless, further improvements are required to unlock the full potential. The application of the RW model should be further investigated. Specifically, research on the optimal values for α_x and β is needed. As α_x depends on the impact of stimulus x on each user, this value could potentially be calculated for each user/stimulus pair individually. Analogously, the parameter β models a property of the user itself, namely, the rate at which patients form new associations. Both parameters could be calculated from the personality treats of the patients.

Furthermore, no claims can be made concerning effectiveness or efficacy at this point. Future studies should be set up with large samples of participants, both psychotherapists and clients, and with a rigorous methodology. The development and evaluation methodology proposed by Birckhead et al. [40] can serve as an excellent framework to further iterate on the design of this algorithm and its VR application.

It is clear from the results that the therapist makes different decisions in some cases than those suggested by the adaptation algorithm. On the one hand, fully quantifying and automating the complex process of ET is probably not possible nor feasible. This is why the adaptation algorithm should primarily be considered as a decision support system to help guide psychotherapists. On the other hand, many solutions proposed by the system could have been proven to be relevant, as it is unlikely that only a single perfect configuration of the VR environment can be made at any given time. Nevertheless, the quality of the system can be further improved by learning from the decisions made by psychotherapists, especially as they also become increasingly familiar with VRET, the functionalities and the potential. Through self-learning technologies, the system would learn to adapt to the intentions of the therapist as well. Furthermore, with more qualitative input data for the algorithm, better suggestions become possible. Where a discrete expectancy violation metric is used in the current design of the algorithm, a continuous expectancy rating can provide a quantitative and qualitative improvement in future studies.

As immersive technology is becoming increasingly accessible, it is bound to spark the interest of psychotherapists as well. One particular technique that has already been proven valuable and effective in research is VRET. We expect that - with further research - adaptation algorithms like the one described in this paper will play a vital role in facilitating successful uptake in clinical practice. These algorithms have the potential to reduce the challenge for psychotherapists to master the operation of the systems fully and to make use of technology for its primary purpose: to facilitate offering high-quality and effective care to those in need of support.

Funding

This research was funded by the imec.ICON project PATRONUS, co-funded by VLAIO, imec, The Human Link, PreviewLabs, Bazookas and Van Roey Automation [HBC.2017.0156].

Statements of ethical approval

This study was approved by the Ghent University Hospital Ethical Committee [EC/2019/0893].

Code repository

All source files of the ontology presented in this work are publicly available on our repository under the CC-BY-4.0 licence: https://github.com/IBCNServices/PATRONUS-ontology.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

Acknowledgement

This research was funded by the imec.ICON project PATRONUS, co-funded by VLAIO, imec, The Human Link, PreviewLabs, Bazookas and Van Roey Automation [HBC.2017.0156].

Appendix A. Listings

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://idlab.ugent.be/ontology/VRETPatronus.owl#"
  xml:base="http://idlab.ugent.be/ontology/VRETPatronus.owl"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://idlab.ugent.be/ontology/
      VRETPatronus.owl">
    <owl:imports rdf:resource="http://idlab.ugent.be/ontology/
        UpperET.owl"/>
  </owl:Ontology>

  <owl:ObjectProperty rdf:about="http://idlab.ugent.be/ontology/
      VRETPatronus.owl#hasVREvent">
    <rdfs:domain rdf:resource="http://idlab.ugent.be/ontology/
        VRETPatronus.owl#VRScenario"/>
    <rdfs:range rdf:resource="http://idlab.ugent.be/ontology/
        VRETPatronus.owl#VREvent"/>
    <rdfs:range rdf:resource="http://idlab.ugent.be/ontology/
        VRETPatronus.owl#VRInteroceptiveStimulus"/>
  </owl:ObjectProperty>

  <owl:Class rdf:about="http://idlab.ugent.be/ontology/VRETPatronus
      .owl#InternalEvent">
    <rdfs:subClassOf rdf:resource="http://idlab.ugent.be/ontology/
        UpperET.owl#InternalStimulus"/>
  </owl:Class>

  ...
</rdf:RDF>
```

Listing A.1. Fragment of the VRET-ontology.

```
prefixes:
  base: "http://idlab.ugent.be/ontology/Patronus/individual/"
  uet: "http://idlab.ugent.be/ontology/UpperET.owl#"

mappings:
  patient:
    sources:
      - [’data.json˜jsonpath’, ’$’]
    s: base:patient_$(ClientId)
    po:
      - [a, uet:Patient]
      - [uet:hasPatientID, $(ClientId), xsd:string]
```

Listing A.2. The mapping rules in YAML syntax instruct to create a new invidual of type *Patient* and give it an ID.

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix ql: <http:// semweb.mmlab.be/ns/ql#>.
@prefix map: <http://mapping.example.com/>.
@prefix uet: <http://idlab.ugent.be/ontology/UpperET.owl#>.
@prefix base: <http://idlab.ugent.be/ontology/Patronus/individual/>
    .

map:map_patient_0 rml:logicalSource map:source_0;
    a rr:TriplesMap;
    rdfs:label "patient";
    rr:subjectMap map:s_0;
    rr:predicateObjectMap map:pom_0, map:pom_1.
map:om_0 a rr:ObjectMap;
    rr:constant uet:Patient;
    rr:termType rr:IRI.
map:om_1 a rr:ObjectMap;
    rml:reference "ClientId";
    rr:termType rr:Literal;
    rr:datatype <http://www.w3.org/2001/XMLSchema#string>.
map:pm_0 a rr:PredicateMap;
    rr:constant rdf:type.
map:pm_1 a rr:PredicateMap;
    rr:constant uet:hasPatientID.
map:pom_0 a rr:PredicateObjectMap;
    rr:predicateMap map:pm_0;
    rr:objectMap map:om_0.
map:pom_1 a rr:PredicateObjectMap;
    rr:predicateMap map:pm_1;
    rr:objectMap map:om_1.
map:s_0 a rr:SubjectMap;
    rr:template base:patient_{ClientId}.
map:source_0 a rml:LogicalSource;
    rml:source "config_report.json";
    rml:iterator "$";
    rml:referenceFormulation ql:JSONPath.
```

Listing A.3. These RML-mapping rules describe the same as the mapping rules in YAML syntax.

References

[1] J.-P. Lépine, The epidemiology of anxiety disorders: prevalence and societal costs, J. Clin. Psychiatry 63 (Suppl 14) (2002) 4–8.

[2] J. Alonso, M.C. Angermeyer, S. Bernert, R. Bruffaerts, T.S. Brugha, H. Bryson, G. Girolamo, R. Graaf, K. Demyttenaere, I. Gasquet, J.M. Haro, S.J. Katz, R.C. Kessler, V. Kovess, J.P. Lepine, J. Ormel, G. Polidori, L.J. Russo, G. Vilagut, J. Almansa, S. Arbabzadeh-Bouchez, J. Autonell, M. Bernal, M.A. Buist-Bouwman, M. Codony, A. Domingo-Salvany, M. Ferrer, S.S. Joo, M. Martinez-Alonso, H. Matschinger, F. Mazzi, Z. Morgan, P. Morosini, C. Palacin, B. Romera, N. Taub, W.A.M. Vollebergh, Prevalence of mental disorders in Europe: results from the European study of the epidemiology of mental disorders (ESEMeD) project, Acta Psychiatr. Scand. 109 (s420) (2004) 21–27, doi:10.1111/j.1600-0047.2004.00327.x.

[3] D.H. Barlow, S.D. Raffa, E.M. Cohen, Psychosocial treatments for panic disorders, phobias, and generalized anxiety disorder, in: A Guide to Treatments that Work, second ed., Oxford University Press, New York, NY, US, 2002, pp. 301–335.

[4] B. Depreeuw, L.A. Andrews, S. Eldar, S.G. Hofmann, The treatment of panic disorder and phobias, in: Evidence-Based Psychotherapy, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2017, pp. 63–93, doi:10.1002/9781119462996.ch4.

[5] M.G. Craske, K. Kircanski, M. Zelikowsky, J. Mystkowski, N. Chowdhury, A. Baker, Optimizing inhibitory learning during exposure therapy, Behav. Res. Ther. 46 (1) (2008) 5–27, doi:10.1016/j.brat.2007.10.003.

[6] M.G. Craske, M. Treanor, C. Conway, T. Zbozinek, B. Vervliet, Maximizing exposure therapy: an inhibitory learning approach, Behav. Res. Ther. 58 (2014) 10–23, doi:10.1016/j.brat.2014.04.006.

[7] K. Meyerbröker, P.M. Emmelkamp, Virtual reality exposure therapy in anxiety disorders: a systematic review of process-and-outcome studies, Depress. Anxiety 27 (10) (2010) 933–944, doi:10.1002/da.20734.

[8] D.D. Ebert, T. Van Daele, T. Nordgreen, M. Karekla, A. Compare, C. Zarbo, A. Brugnera, S. Øverland, G. Trebbi, K.L. Jensen, F. Kaehlke, H. Baumeister, Internet- and mobile-based psychological interventions: applications, efficacy, and potential for improving mental health: a report of the EFPA e-health taskforce, Eur. Psychol. 23 (2) (2018) 167–187, doi:10.1027/1016-9040/a000318.

[9] D. Freeman, S. Reeve, A. Robinson, A. Ehlers, D. Clark, B. Spanlang, M. Slater, Virtual reality in the assessment, understanding, and treatment of mental health disorders, Psychol. Med. 47 (14) (2017) 2393–2400, doi:10.1017/S003329171700040X.

[10] P. Best, A. McKenna, P. Quinn, M. Duffy, T. Van Daele, Can virtual reality ever be implemented in routine clinical settings? A systematic narrative review of clinical procedures contained within case reports for the treatment of PTSD, Front. Virtual Real. 1 (2020) 563739, doi:10.3389/frvir.2020.563739.

[11] S.Y. Park, S.M. Kim, S. Roh, M.-A. Soh, S.H. Lee, H. Kim, Y.S. Lee, D.H. Han, The effects of a virtual reality treatment program for online gaming addiction, Comput. Methods Programs Biomed. 129 (2016) 99–108, doi:10.1016/j.cmpb.2016.01.015. https://linkinghub.elsevier.com/retrieve/pii/S0169260716000079

[12] A. Rizzo, M.J. Roy, A. Hartholt, M. Costanzo, K.B. Highland, T. Jovanovic, S.D. Norrholm, C. Reist, B. Rothbaum, J. Difede, Virtual reality applications for the assessment and treatment of PTSD, in: S.V. Bowles, P.T. Bartone (Eds.), Handbook of Military Psychology, Springer International Publishing, Cham, 2017, pp. 453–471, doi:10.1007/978-3-319-66192-6_27.

[13] R. Baños, C. Botella, M. Alcañiz, V. Liaño, B. Guerrero, B. Rey, Immersion and emotion: their impact on the sense of presence, CyberPsychology Behav. 7 (6) (2004) 734–741, doi:10.1089/cpb.2004.7.734.

[14] M. Slater, Immersion and the illusion of presence in virtual reality, Br. J. Psychol. 109 (3) (2018) 431–433, doi:10.1111/bjop.12305.

[15] P. Lindner, A. Miloff, W. Hamilton, L. Reuterskiöld, G. Andersson, M.B. Powers, P. Carlbring, Creating state of the art, next-generation virtual reality exposure therapies for anxiety disorders using consumer hardware platforms: design considerations and future directions, Cogn. Behav. Ther. 46 (5) (2017) 404–420, doi:10.1080/16506073.2017.1280843.

[16] P. Bun, F. Gorski, D. Grajewski, R. Wichniarek, P. Zawadzki, Low - cost devices used in virtual reality exposure therapy, Procedia Comput. Sci. 104 (2017) 445–451, doi:10.1016/j.procs.2017.01.158.

[17] T. Donker, I. Cornelisz, C. van Klaveren, A. van Straten, P. Carlbring, P. Cuijpers, J.-L. van Gelder, Effectiveness of self-guided app-based virtual reality cognitive behavior therapy for acrophobia: arandomized clinical trial, JAMA Psychiatry 76 (7) (2019) 682, doi:10.1001/jamapsychiatry.2019.0219.

[18] P. Lindner, A. Miloff, E. Zetterlund, L. Reuterskiöld, G. Andersson, P. Carlbring, Attitudes toward and familiarity with virtual reality therapy among practicing cognitive behavior therapists: a cross-sectional survey study in the era of consumer VR platforms, Front. Psychol. 10 (2019) 176, doi:10.3389/fpsyg.2019.00176.

[19] T. Van Daele, M. Karekla, A.P. Kassianos, A. Compare, L. Haddouk, J. Salgado, D.D. Ebert, G. Trebbi, S. Bernaerts, E. Van Assche, N.A.J. De Witte, Recommendations for policy and practice of telepsychotherapy and e-mental health in Europe and beyond, J. Psychother. Integr. 30 (2) (2020) 160–173, doi:10.1037/int0000218.

[20] S. Lissek, A.S. Powers, E.B. McClure, E.A. Phelps, G. Woldehawariat, C. Grillon, D.S. Pine, Classical fear conditioning in the anxiety disorders: a meta-analysis, Behav. Res. Ther. 43 (11) (2005) 1391–1424, doi:10.1016/j.brat.2004.10.007.

[21] P. Eelen, B. Vervliet, Fear conditioning and clinical implications: what can we learn from the past?, in: M.G. Craske, D. Hermans, D. Vansteenwegen (Eds.) Fear and Learning: From Basic Processes to Clinical Implications, American Psychological Association, Washington, 2006, pp. 17–35, doi:10.1037/11474-001.

[22] B. Vervliet, M.G. Craske, D. Hermans, Fear extinction and relapse: state of the art, Annu. Rev. Clin. Psychol. 9 (1) (2013) 215–248, doi:10.1146/annurev-clinpsy-050212-185542.

[23] R. Rescorla, A.R. Wagner, A theory of pavlovian conditioning : variations in the effectiveness of reinforcement and nonreinforcement, Classical Conditioning: Current Research and Theory, 1972.

[24] R.A. Greenes, Clinical Decision Support, Elsevier, 2014, doi:10.1016/C2012-0-00304-3.

[25] R.E. McCabe, Subjective units of distress scale, in: Phobias: The Psychology of Irrational Fear, ABC-CLIO, LLC, Santa Barbara, California, 2015, pp. 361–362.

[26] D.B. Yamada, V.T. Yoshiura, N.S. Brandão Miyoshi, I.B. de Lima, G.Y. Usumoto Shinoda, R.P.C. Lopes Rijo, J.M. de Azevedo Marques, M.M. Cruz-Cunha, D. Alves, Proposal of an ontology for mental health Management in Brazil, Procedia Comput. Sci. 138 (2018) 137–142, doi:10.1016/j.procs.2018.10.020.

[27] F. Ongenae, P. Duysburgh, N. Sulmon, M. Verstraete, L. Bleumers, S. De Zutter, S. Verstichel, A. Ackaert, A. Jacobs, F. De Turck, An ontology co-design method for the co-creation of a continuous care ontology, Appl. Ontol. 9 (1) (2014) 27–64, doi:10.3233/AO-140131.

[28] X.H. Wang, D.Q. Zhang, T. Gu, H.K. Pung, Ontology based context modeling and reasoning using OWL, in: IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, IEEE, Orlando, FL, USA, 2004, pp. 18–22, doi:10.1109/PERCOMW.2004.1276898.

[29] D. Isern, D. Sánchez, A. Moreno, Ontology-driven execution of clinical guidelines, Comput. Methods Programs Biomed. 107 (2) (2012) 122–139, doi:10.1016/j.cmpb.2011.06.006.

[30] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, Y. Katz, Pellet: a practical OWL-DL reasoner, J. Web Semant. 5 (2) (2007) 51–53, doi:10.1016/j.websem.2007.03.004.

[31] D.L. McGuinness, F. Van Harmelen, OWL web ontology language overview, W3C Recomm. 10 (10) (2004) 1–12.

[32] M. Horridge, S. Bechhofer, The OWL API: a java API for OWL ontologies, Semant. Web 2 (1) (2011) 11–21, doi:10.3233/SW-2011-0025.

[33] N. Noy, M. Sintek, S. Decker, M. Crubezy, R. Fergerson, M. Musen, Creating semantic web contents with protege-2000, IEEE Intell. Syst. 16 (2) (2001) 60–71, doi:10.1109/5254.920601.

[34] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: a semantic web rule language combining OWL and RuleML, W3C Memb. Submiss. 21 (79) (2004) 1–31.

[35] X. Tang, M. Xiao, B. Hu, D. Pan, Exchanging knowledge for test-based diagnosis using OWL ontologies and SWRL rules, Procedia Comput. Sci. 131 (2018) 847–854, doi:10.1016/j.procs.2018.04.279.

[36] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, R. Van de Walle, RML: a generic language for integrated RDF mappings of heterogeneous data, in: 7th Workshop on Linked Data on the Web, Proceedings, Event-Place: Seoul, Korea, 2014, p. 5.

[37] P. Heyvaert, B. De Meester, A. Dimou, R. Verborgh, Declarative rules for linked data generation at your fingertips!, in: The Semantic Web: ESWC 2018 Satellite Events, in: Lecture Notes in Computer Science, vol. 11155, Springer International Publishing, Cham, 2018, pp. 213–217, doi:10.1007/978-3-319-98192-5_40.

[38] R. Zgheib, S. Kristiansen, E. Conchon, T. Plageman, V. Goebel, R. Bastide, A scalable semantic framework for IoT healthcare applications, J. Ambient Intell. Humaniz. Comput. (2020), doi:10.1007/s12652-020-02136-2.

[39] T. Overbeek, K. Schruers, E. Griez, M.I.N.I., mini international neuropsychiatric interview, nederlandse versie 5.0.0, 1999.

[40] B. Birckhead, C. Khalil, X. Liu, S. Conovitz, A. Rizzo, I. Danovitch, K. Bullock, B. Spiegel, Recommendations for methodology of virtual reality clinical trials in health care by an international working group: iterative study, JMIR Ment. Health 6 (1) (2019) e11973, doi:10.2196/11973.