Self-Adaptive Games for Rehabilitation at Home

Michele Pirovano, Renato Mainetti, Gabriel Baud-Bovy, Pier Luca Lanzi, Nunzio Alberto Borghese

Abstract—Computer games are a promising tool to support rehabilitation at home. It is widely recognized that rehabilitation games should (i) be nicely integrated in general-purpose rehabilitation stations, (ii) adhere to the constraints posed by the clinical protocols, (iii) involve movements that are functional to reach the rehabilitation goal, and (iv) adapt to the patients’ current status and progress. However, the vast majority of existing rehabilitation games are stand-alone applications (not integrated in a patient station), that rarely adapt to the patients’ condition. In this paper, we present the first prototype of the patient rehabilitation station we developed that integrates video games for rehabilitation with methods of computational intelligence both for on-line monitoring the movements’ execution during the games and for adapting the gameplay to the patients’ status. The station employs a fuzzy system to monitor the exercises execution, on-line, according to the clinical constraints defined by the therapist at configuration time, and to provide direct feedback to the patients. At the same time, it applies real-time adaptation (using the Quest Bayesian adaptive approach) to modify the gameplay according both (i) to the patient current performance and progress and (ii) to the exercise plan specified by the therapist. Finally, we present one of the games available in our patient stations (designed in tight cooperation with therapists) that integrates monitoring functionalities with in-game self-adaptation to provide the best support possible to patients during their routine.

I. INTRODUCTION

Modern motion-sensing game controllers like the Nintendo Wii and balance board\(^1\), the Sony Move\(^2\), and Microsoft Kinect\(^3\) have revolutionized the way people play video games. These controllers capture players movements in the real world and convey them inside the game, transforming the players in the controllers themselves, making games much more intuitive to play and thus accessible to a broader audience. Accordingly, these devices have rapidly become a major source of inspiration for the researchers working in rehabilitation who immediately recognized the potential of these new technologies (see [1] for a recent survey). In this context, computer games appear as the best way to guide rehabilitation while limiting the typical boredom and fatigue affecting patients in their daily routine.

Rehabilitation games should be designed in tight cooperation with therapists to adhere to the constraints posed by the clinical protocols and to require actions that are functional to reach the rehabilitation goal. It should also be possible to tailor them to the rehabilitation goals set and to adapt them to the specific patient’s state (possibly in real-time) by smoothly changing the game parameters defining rehabilitation (e.g., the level of difficulty). Moreover, the movements and the performance of the patient should be tracked in order to assess and monitor the correct execution of the exercises. Finally, they should be designed to motivate patients and to make rehabilitation more engaging. These goals are not easy to reach as intensive rehabilitation requires regular daily sessions to recover functionality [2].

Related Work. In the last few years, researchers have proposed several integrated solutions for computer-assisted rehabilitation, both at hospital and at home [1], [3], [4]. All authors generally agree that rehabilitation applications (especially games) should adapt to the actual patient status and capability [5]. For instance, Burke et al. [6] list an adaptable difficulty level as an essential factor in designing games for rehabilitation. The vast majority of the rehabilitation applications are however limited in this respect in that they rarely include any sort of adaptation. Many application papers are based on selecting games developed for the entertaining market and using them for rehabilitation purposes. The control of these games is however unsuitable for the pace and the goals of rehabilitation [6], [7], yet very few authors designed rehabilitation games tailored to a specific group of patients in which both therapist and patient needs are explicitly considered [8]. In addition, little research has been carried out in making games self-adaptive: among the 200 papers considered in our recent review on rehabilitation applications and games [1] only eight of them mention some kind of adaptation. More specifically, the first adaptation scheme was based on heuristics [9] and it was aimed at spatially distributing targets at each trial according to the performance of the patient. Different authors proposed their own adaptation heuristics, based on various measures related to the patient’s performance [10]. Others improved upon the first techniques, using game physics to define parameters for adaptation [11] or adapting the game to the emotional state of the patient [12]. Similar techniques have been also employed independently by different authors in order to maintain the performance of the patient at an acceptable level [7]. It was not until 2011, however, that advanced methods of computational intelligence were applied to special games explicitly targeted to rehabilitation. Rossol et al. [13] introduced Bayesian networks, configured using the

\(^1\)http://www.nintendo.com
\(^2\)http://en.wikipedia.org/wiki/PlayStation_Move
\(^3\)http://en.wikipedia.org/wiki/Kinect
therapist’s knowledge, for the design of exercises suited to different patients. Computational intelligence has also been introduced in robots guided rehabilitation [14], [15], in which the motion and/or the force field generated by the robots to which the patient is attached can be adapted to the patient status.

In this paper, we show how Computational Intelligence can be used to realize a Game Engine targeted to Rehabilitation (IGER), that would make rehabilitation at home feasible and robust. The gameplay is continuously monitored using a fuzzy system (devised in cooperation with therapists) to avoid that patients assume wrong postures or perform wrong movements which may make rehabilitation harmful. Games are continuously adapted to the current patient status through a Bayesian framework that updates game parameters to provide an adequate difficulty level. Games have been designed to be fully configurable so that they can be tailored by the clinicians the patient needs and the rehabilitation goals set. Finally, we present one of the games developed by us, the Fruit Catcher. This was developed according to the requirements specified by the clinicians and results on test scenarios are reported and discussed.

II. A HIERARCHICAL PLATFORM FOR REHABILITATION AT HOME

The Intelligent Game Engine for Rehabilitation (IGER) has been developed as a component of the Rewire platform, developed under the FP7 framework. The project aims at developing a low-cost game-oriented platform that enables patients, discharged from the hospital, to continue intensive rehabilitation at home under remote monitoring by the hospital. The main idea of Rewire is to assemble off the shelf components in a robust and reliable way to allow deploying the platform massively at the patients’ homes.

The Rewire platform is constituted of three components: a patient station (PS), a hospital station (HS), and a networking station (NS). The HS allows clinicians to monitor remotely the ongoing rehabilitation at patient’s home. It also provides to the clinicians the tools to configure and schedule the rehabilitation games. Finally it supports a virtual community with the function of educating and motivating the patients. The NS is installed at the health provider site, at a Regional Level. It provides advanced data mining procedures aimed at discovering common features and trends of rehabilitation treatment among hospitals and regions.

The PS is installed at patients’ home. It is the core of the platform and has been designed to fulfill the need for friendliness and efficacy. Friendliness is achieved by hiding the therapeutic exercises underneath compelling video games, specifically created for rehabilitation. Efficacy is achieved through clinically valid exercises proposed by the clinicians and by the continual monitoring of the patients during their training. The PS has four modules. The hospital communication module is used by patients to interact with the clinicians at the hospital and to download game configurations. The lifestyle module collects data on the patient daily activity through a body sensor network. From these data the activity can be profiled [16]. The data can be used, along with environmental and physiological data, to tune the rehabilitation exercise level, assess potential risks, and advice clinicians on the therapy. The community module acts as a client of the patients community and allows patients to interact with other fellow patients online. Finally, the IGER module (Figure 1) guides the actual rehabilitation using 3D games and it is the focus of the present work.

III. THE GAME SYSTEM

The IGER module comprises two components, the game engine and the game control unit. The former provides all the gaming functionalities: input data, animation, collision detection, rendering and game logic. The latter contains the game control unit developed to match the needs of games for rehabilitation: it schedules the games, chosen and configured according to therapeutic needs, it adapts the game parameters to the actual patient performance so that an adequate challenge level is maintained, it supervises the gaming sessions and checks that the patient movement complies to the rehabilitation guidelines set by the therapist (Figure 1).

The exercises associated to each rehabilitation session are configured off-line at the hospital by the clinicians and customized for each patient. The clinicians can select which exercises and consequently which games to include in each training session (notice that there might be more games associated to the same exercise), the time allocated to each exercise, the number of repetitions, and the difficulty level. In addition, the clinicians can also select the best input devices based on the rehabilitation task and the patient status (see Section VI). For example, as illustrated in Figure 2, the therapist can select (panel A) the games used (in this case, the Fruit Catcher) and the associated input device (for instance the Wii Balance Board). The therapist can set the number of repetitions (5), or alternatively the total game time for the exercise, the number of trials for each repetition (10), and the parameters associated to the game that have direct impact on the movement speed and accuracy of the patient movement. For instance, in our

---

4http://www.rewire-project.eu
example, the therapist defines the maximum range from which fruits will fall at 0.1m, both on the left and on the right, the fruit dimension to be 5 times the default, a really high value, and the default gravity (i.e., 1g). The overall configuration is also shown graphically (panel B) to make the work of the therapist easier: any modification is reflected in real-time inside this visual interface. For instance, a change in the range of falling produces an immediate enlargement of the area from which the fruits fall.

IV. Fuzzy-Based Monitoring

The configuration interface is also used to define the rules that have to be complied to by the player during the game. For instance, while playing the Fruit Catcher, the patient may try to compensate her limited range of movement by stepping laterally, which is not allowed for this game. In this case, the monitor will detect the wrong motion of the feet and issue a warning, eventually stopping the game or the rehabilitation session and issuing an error.

The clinicians can also define constraints (panel C in Figure 2): for example, the patient may not move the feet from the platform more than 0.1 m away from the original position and the patient should not incline the head by more than 30 degrees in any direction. These constraints can be directly specified as numerical values or, in a more natural way, by modifying a symbolic representation of the patient’s skeleton, dragging visual handles to extend the limits (panel D). For each variable, the clinicians can also define a severity level that defines the importance of each constraint, basically deciding whether a constraint will be hard or soft, and thus will preferentially raise either an error or a simple warning. At the end, the clinician saves the configuration and sends it to the patient station along with the list of scheduled games. All the monitoring functionalities are maintained by the Knowledge-based Monitor that applies a set of fuzzy rules, previously specified by the clinicians, to produce an alarm level.

Rule Checking. The configuration parameters and the constraints are used with performance data and input measurements to determine whether the patient is doing something wrong and thus whether an alarm must be raised or not.

We have implemented two alternative monitoring tools. The first one allows us to define several monitors and attach them to the variables whose changes we want to keep under control according to the game configuration. These monitors can easily keep track of the variations in the value of these variables and, in case predefined limits are exceeded, an alarm is raised according to the predefined severity level specific to the monitor.

However, monitoring is often more complex and it is based on clinicians’ experience. This can be hardly converted into crisp reasoning and the most suitable framework to represent this appears that of fuzzy systems. In this case, the variables defined during the configuration phase are transformed into fuzzy variables associated to a set of classes of progressive severity. For instance, continuing the example in Section III, the constraint that the head must not rotate by more than 30 degrees is fuzzified into an input variable, the head tilt. Similar fuzzy variables can be defined for other body segments. We here define an additional constraint: that the patient must remain upright and thus her spine must not incline more than 10 degrees in any direction. The input variables and their membership functions are depicted in Figure 3. In addition to the input variables associated to body segments, the fuzzy monitoring system accepts also variables that take explicitly into account time, previous alarms and patient performance.

Using the severity level of each variable defined at configuration time, a set of fuzzy rules is constructed that map the input variables to the output fuzzy variable: the alarm level. The alarm level is at last defuzzified according to its set of membership functions as seen in Figure 4 in order to obtain a numerical alarm value (or, if needed, the label corresponding to the severity level).

Alarm Raising. When a constraint is violated, the system issues an alarm. There are five alarm severity levels (silent, logging, warning, error and shutdown), each level triggering a different behavior of the IGER module.

When the alarm level is zero, no action is performed. The alarm level is set at the silent level. If the alarm reaches the...
logging level, a notification of the alarm cause is added to the game log. This can be useful for subsequent analysis by the clinicians or for data mining purposes. If the warning level is reached, a warning is issued to the patient, prompting her to correct her bad behavior. The event is also logged. This level is basically associated to violating soft constraints. In the case of mildly dangerous situations or specific erroneous movements, the error severity level may be reached. In addition to issuing a warning and logging the event, the game is paused. The player is then given an explanation of the cause of the error and the option to resume or repeat the game. This level basically identifies a hard constraint. The shutdown severity level is reserved for very dangerous situations. The game is stopped and a warning is immediately sent to the Hospital Station prompting the clinicians to contact the patient.

The warnings are managed by the game control unit that shows them to the player in the form of visual feedback inside the game (text or icons) and audio feedback (e.g., warning sounds). The icons and sounds are especially useful as they provide immediate and easily understandable feedback to the patient; they can be associated to messages specific to the constraint (or constraints) that has been violated. For instance, a message which says that the head inclination is risky for the patient’s own safety during the game would clarify what is going wrong with exercising.

V. BAYESIAN-BASED IN-GAME ADAPTATION

Adaptation (Fig. 1) is based on the estimate of a set of parameters and the subsequent adaptation of their value as a function of the interaction history with the game. The adequacy of the actual game parameters can be inferred by measuring the hit ratio inside the game. This ratio can be associated with the subjective difficulty of the game (and thus of the exercise) for a given patient. In this work, we have implemented two alternative algorithms for real-time performance-based parameter estimate, that can be performed either on a per-repetition basis or on a per-trial basis.

The first algorithm performs a simple adaptation based on the performance of the patient in one repetition, similar to previous implementations ([9], [11]). Before a game starts, the $n$ game parameters $\mathbf{x} = [x_1, x_2, \cdots, x_n]$ that can be adapted, and a target performance level, $p_{end}$, are set. At the end of each repetition, we compute the performance $p_R$ as the ratio between the number of successful trials $N_s$ and the total number of trials $N_{tot}$ in the repetition:

$$p_R = \frac{N_s}{N_{tot}}$$

This represents the hit ratio. The parameters $\mathbf{x}$ are then updated as follows. If the ratio $p_R$ is higher than $p_{end}$ the patient is over-performing and $\mathbf{x}$ is therefore incremented, increasing the difficulty of the next repetition:

$$\mathbf{x} = \mathbf{x} + d\mathbf{x}$$

If $p_R$ is lower than $p_{end}$ the patient is under-performing and $\mathbf{x}$ is decreased making the game easier at the next repetition:

$$\mathbf{x} = \mathbf{x} - d\mathbf{x}$$

As a result, the game will modify its challenge level to match the skills of the patient, avoiding frustration on the one hand and maintaining the proper challenging level on the other.

Our second parameter estimate algorithm is built upon the Quest Bayesian adaptive method [17], typically used in psychophysics to adapt a psychometric threshold for audial, visual or tactile stimuli. The threshold can be adapted according to the result of previous trials, which is usually expressed as a binary outcome. This method has been modified here to adapt the game difficulty in real-time: contrary to the previous algorithm, the selected parameters are modified here on a per-trial basis where each trial ends with the outcome of the interaction with a game element and it is of binary type: either success or failure. To adapt the parameters, we first define the probability density function (pdf) of the parameter $x$, $f_x(x)$. This is assumed here as a Gaussian function centered in the $x_{prior}$ value and with standard deviation $\sigma_{prior}$, both provided at configuration time. $x_{prior}$ represents the a priori initial value of the parameter, for instance the initial range from which fruit falls in the Fruit Catcher game (Section VI) as chosen by the therapist (see Section IV). Given a certain desired success rate, $p_{end}$, we aim to compute the value of $x$, $x_{end}$, that is expected to be the threshold at which the posterior pdf, $f_{x|D}(x|D)$, converges to $p_{end}$. $f_D(D)$ represents the priori pdf distribution of the outcome of the previous trials.

This method is based on three assumptions that are satisfied in our case. First, the function that relates the parameter to the performance in the game must have the same shape under all conditions. This can be obtained by considering an adequate function (see later). Second, the patient threshold should not vary from trial to trial. Third, individual trials must be statistically independent. The last two assumptions are satisfied, since the adaptation is carried out inside a single session of a game and thus patient’s skill improvement can be safely ignored.

Considering explicitly the dependency between $x$ and $D$, through the Bayes theorem, we can obtain the a posteriori pdf as

$$f_{x|D}(x|D) = \frac{f_x(x)f_D(x|D)}{f_D(D)}$$

$x_{end}$ is thus the expected value of $f_{x|D}(x|D)$. On the right side of equation 2 the term $f_D(D)$ is constant. Thus, we are
left with the other two terms

\[ Q(x) = f_x(x) f_D|x (D|x) \]  \hspace{1cm} (3)

where \( Q(x) \) is the Quest function. By defining the performance function, \( p_x(x) \), that determines the probability that, given the threshold \( x \), using parameter value \( x_i \) the trial \( i \) will be a success, we can rewrite the Quest function as

\[ Q(x) = f_x(x) \prod_{i=1}^{n} p_{r|x}(x_i) \]  \hspace{1cm} (4)

Where \( p_{r|x}(x_i) \) is the probability of trial \( i \) tested at the selected parameter \( x_i \) and with outcome \( r_i \) which is either 0 (miss) or 1 (hit). Since we are working under the assumption that \( p_x(x_i) \) maintains the same shape between trials, but just shifts on the axis of the parameter range according to its threshold \( x \), we can rewrite it through a canonical function \( \Psi(x_i - x) = p_x(x_i) \) and thus define a success function \( S(x_i) \) and a failure function \( F(x_i) \) as follows:

\[ S(x_i) = \Psi(-x_i) \]
\[ F(x_i) = 1 - \Psi(-x_i) \]  \hspace{1cm} (5)

Thanks to this, the Quest function \( Q_i(x) \) at trial \( i \) can be obtained from \( Q_{i-1}(x) \) by shifting \( S(x_i) \) (or \( F(x_i) \), if the outcome was a failure) by intensity \( x_i \) and multiplying the result:

\[ Q_i(x) = Q_{i-1}(x) \cdot \begin{cases} S(x - x_i) & \text{if success} \\ F(x - x_i) & \text{if failure} \end{cases} \]  \hspace{1cm} (6)

At each trial \( i \), the parameter value \( x_i \) is chosen as the mean of the posterior pdf \( f_x,D|x(D|x) \) (Eq. 2) as suggested by [18]. The Quest function \( Q(x) \) is then modified and the next trial can start.

We can apply the modified Quest method to maintain a chosen performance level \( p_{end} \) for the player by adapting the game’s parameters. As a result, the value of \( x_i \) of each trial will depend on the previous value of \( x \) and on the actual threshold, \( x_{end} \). The actual value, that can be for instance the range from which fruits are falling in the Fruit Catcher game can be seen as a measure of the game’s difficulty.

In our approach, as suggested by [17], we use the Weibull function for \( p_x(x_i) \)

\[ p_x(x_i) = \delta \gamma + (1 - \delta) \cdot (1 - (1 - \gamma)e^{-\beta x_i}) \]  \hspace{1cm} (7)

where \( \delta \) is the percentage of failure at maximum intensity and should be a small value that accounts for the so called finger mistakes, \( \gamma \) is the probability of success at zero intensity and it depends on the type of parameter \( x \), and \( \beta \) is the slope of the psychometric function. In our case, \( \delta \) is set to the very small value of 0.01 while \( \gamma \) is different for different game parameters. For the falling range in the Fruit Catcher game, \( \gamma \) represents the probability of success when the range goes to infinity and this has been set to a very small value of 0.001. The value of \( \beta \) was set to 3.5 as suggested by [18].

Since our aim is to achieve a fixed performance level, we modify the performance function so that the expected percentage of successes is equal to the performance level \( p_{end} \). We thus interpolate \( p_x(x_i) \) in order to find the correspondent parameter value \( x_{end} \). The performance function thus becomes

\[ p_x(x_i) = \delta \gamma + (1 - \delta) \cdot (1 - (1 - \gamma)e^{-\beta x_i}) \]  \hspace{1cm} (8)

Fig. 6 shows the time course of one of the adapted parameters for the Fruit Catcher game, the fall range. \( x_{prior} = 1.0m \) and \( \sigma_{prior} = 0.8mm \) inside the virtual environment, were set at configuration time. \( p_{end} \) is 70%. \( x \) is first increased because of the good performance of the patient but is lowered afterwards when the performance starts decreasing. The game control unit continuously adapts the parameter value, that eventually converges to the value \( x_{end} \) that guarantees that, on average, \( p_{end} \) trials will end with a success for that patient in her actual status. We remark here that the hit ratio for the present experiment, computed as the average success in all trials (Eq. 1), is \( p_R = 0.6571 \), a value that is expected to converge to the chosen \( p_{end} \) with a greater number of trials, due to the convergence of \( x \) to \( x_{end} \).

**Interaction between Monitoring and Adaptation.** The work of the Parameter Estimation Sub-module is strictly supervised by the Knowledge-based Monitor, that can block or mitigate the adaptation.
If the alarm level is close to zero, the parameters are modified and sent to the game engine, resulting in a full direct adaptation. However, if the alarm level is at a high level, the modifications suggested by the Parameter Estimation Sub-module are considered only if they are in the direction to make the game easier. Otherwise, the modification of the parameters is reduced, in case the alarm is at a medium level, or even discarded in the case that the alarm is at a high level, thus performing an indirect adaptation. This could also mean that the change of the parameter value was too large, since the alarm level depends also on the increment given to the parameters.

To increase game variability, we have introduced a controlled degree of randomization inside the game. For example, in the Fruit Catcher game, the lateral falling position of each fruit is chosen randomly in each trial, inside the current range. The same is true for the number of fruits falling on the left or on the right side. This randomization decreases the monotonousness of the game, according to game design principles [19] and it allows the patients to perceive the game as different each time they approach it. In this latter case the adaptation cannot be made on a per-trial basis, but has to be carried out always on a per-repetition basis, also when Quest Bayesian adaptation is used.

VI. A SELF-ADAPTIVE REHABILITATION GAME

We tested our first prototype on healthy subjects to evaluate its usability. The tests were carried out using Fruit Catcher, one of the games developed in strict contact with the therapists. Fruit Catcher is very simple as it is aimed at recovering basic postural control by improving balancing ability while increasing the perception of the body scheme.

In Fruit Catcher (Figure 7), the player must catch fruits falling from the top of a tree. The player stands below the tree with a basket placed on the top of the head and can move the body laterally to catch the falling fruits with the basket. When a fruit falls into the basket, the player’s score increases and an audiovisual feed-back is given. When the fruit falls on the ground, the score is decreased. The fruits ripen then fall down from the tree from different heights and from different lateral positions. The basket size, the fruit size and weight, the falling frequency, the horizontal range and the number of falling fruits for each repetition, are all parametrized. The game can be played either with the Nintendo Wii Balance Board or Microsoft Kinect according to the exercise goals set and the patient status. A video showing the test cases can be downloaded at http://borghese.dsi.unimi.it/Research/LinesResearch/Virtual/Virtual.html.

A. Rehabilitation Scenario 1 - Weight Shifting

This exercise is performed using the Wii Balance Board, although it can be also done using the Kinect. During the weight shift exercise, the fruits fall either on the left or the right side of the screen and the patients are required to shift their balance to catch the fruits with the basket, requiring a displacement of the whole body while keeping the feet still on the ground. It is worth noting that this scenario shares several similarities with other mini-games available in Wii Sports and Wii Party but, in this case, its pace and difficulty level as well as the player posture are carefully monitored to avoid that wrong postures are assumed by the patient making rehabilitation even harmful. In this scenario, the game parameters were set by us (acting as therapists) at the following value: initial fruit range 1.0m, maximum fruit range 1.5m, both to the left and the right. In addition, two additional constraints are introduced on the feet position and head tilt (Section IV): the feet must be flat on the ground (hard constraint), the head should not be tilted by more than 30 degrees (soft constraint) and the upper body should not be tilted by more than 10 degrees.

During the first trials, the player starts collecting fruits at a low difficulty level. The adaptation module identifies that the patient is slightly overperforming and, after a first oscillation, adapts the falling range by increasing it to about 1.2m (Figure 6). If the patient plays the game correctly, e.g., without moving his feet or tilting his head, no warning appears and the game keeps adapting the difficulty while still satisfying the constraints specified by the therapists. After a few additional trials, the player’s performance starts to drop: the difficulty level auto-adjusts to match the new performance level to avoid frustrating the patient, while the alarm level is raised towards the logging severity level (Figure 4). The event is logged to record that the player’s performance dropped, a piece of information that can be later useful for the therapist to further tune the game parameters.

B. Rehabilitation Scenario 2 - Stepping

This exercise extends the previous one by allowing patients to move their feet and thus it targets healthier patients. The game guides the patient with the same mechanism used before: the fruits fall laterally and must be collected. The patient is required to move laterally while keeping balance and the body vertical. While the previous rehabilitation exercise could be performed either using the Wii Balance Board and Microsoft Kinect, here the therapist can only select the latter as the movement of the whole body has to be tracked. To this aim we have developed our own C++ library, built upon the Microsoft
Kinect SDK, to obtain the 3D position of the joints of the patient skeleton. From these we compute the joints rotation and map them to the skeleton of a 3D avatar in real-time, thus creating a motion capture device. Since the game now requires the patient to move laterally, the falling fruit maximum lateral range is increased to $1.5m$ and the constraint on the feet position is removed.

In this case, the subjects have tried to simulate a mild hemiparesis by reducing their movement capabilities on the left side of their body. As before, the game starts easy and the adaptation modifies the difficulty level to match the player performance. Due to the (simulated) player impairment on the left side, the patient has more difficulty in catching fruit falling on his left side. The control unit notices this and decreases the falling range on the patient’s left side while at the same time increases the number of fruits falling on that side, so to exercise more the hemiparetic side without stressing it too much. If the player erroneously starts to tilt his head and/or trunk due to the weakness of the left side (even while maintaining a consistent performance), the Knowledge-based Monitor is activated. The alarm level is raised and a first warning is presented using visual/audio feedback (Figure 8). At the same time, the Knowledge-based Monitor stops the adaptation, analyzes the alarm level, and decides that the falling range will not be increased until the erroneous posture is rectified. If the player keeps tilting his head or trunk erroneously, eventually approaching the maximum limit, the combined analysis of the alarm history (duration and level) and the severity of the situation (the amount of tilt) increases the alarm level even more. Thus, an error is issued to the patient, pausing the game and explaining what is wrong. The player is then enabled to correct his posture and he can resume the game to finish the rehabilitation session with ease. All these events are logged and notified along with the game interaction and movement data to the therapist who will use these information for further tuning.

VII. DISCUSSION

The key elements to offer efficient games for rehabilitation have been implemented inside the IGER module, as described in Section V. This allows maintaining a suitable challenge level, by analyzing the patient performance, and monitoring the safety and correctness of the exercises. IGER is based on implicit intelligence to estimate and adapt the game difficulty level on a per-trial or per-session basis and on explicit intelligence to monitor the rehabilitation session, implementing a virtual supervisor to support the therapy.

In particular, implicit intelligence is aimed at maintaining a given success rate, that is kept usually very high (> 90%) so that the patient is, on one side, stimulated to do its best not to fail, on the other side he never gets into frustration. This is very different from most commercial games which often require that the player achieves a certain proficiency level inside the game to progress. The real-time adaptation to the patient’s skills has a double beneficial effect: it controls the amount of physical stress of the patient and provides a tailored challenging environment. Coupling the adaptation with the added randomness, we obtain ever-changing games which can maintain the focus of the player for a longer time.

In the present version the $a\ priori$ expected value of the threshold, $x_{\text{prior}}$, is defined at configuration time. This value can be very different from the threshold to which the adaptation module converges, $x_{\text{end}}$. In this case, the therapist would have largely underestimated or overestimated the patient ability; the value of $x_{\text{end}}$ would allow the therapist to better tune the rehabilitation program for that patient in the future. In our tests, we set the prior threshold $x_{\text{prior}}$ according to the outcome of few repetitions of the game played by healthy subjects. We aim at carrying out more tests in the next future with a larger player population to achieve a better estimate of $x_{\text{prior}}$ and $\sigma_{\text{prior}}$.

The shape of $p_x(x_i)$ in Eq. 7 is given here $a\ priori$ as a Weibull distribution. Although its parameters could be defined by analyzing the behavior of normal players in the game, the values proposed in [18] have allowed a very good and smooth adaptation of the game parameters for the tests we conducted and can thus be considered a good starting point.

The knowledge-based monitor overviews the patient’s behavior. This acts as a virtual therapist and pays attention to the way in which movements are performed. A balance between giving advice and being not too intrusive is obtained through a fuzzy combination of the rules that are associated with the different constraints defined by the therapist and with the frequency of previous intervention of the monitor. There might be the case in which a parameter is increased too fast and too much and the patient to keep up with the game pace, starts assuming the wrong posture or executing the movements in the wrong way. The knowledge based control, in this case, reduces the parameters to bring back that game at a reasonable difficulty level. Again, the monitoring is aimed at controlling that the game prompts the patient to execute the movements correctly while leaving to the adaptation module the role of...
finding a proper value for the parameter to guarantee the adequate level of challenge. Here we use only one output variable for the monitor module, the alarm level, but several output variables can be inserted in the monitoring module, in case.

By continuously monitoring the patient movement, according to the knowledge defined by the therapist, the game system can react instantaneously with direct and clear feedback and thus avoids the appearance of over-compensatory or erroneous movements during the exercise that would lead to the failure of the rehabilitation program. Such knowledge can be inserted inside the IGER module by defining, with an intuitive interface, the constraints on the exercises, while the inner workings of the control unit are hidden to the therapist, allowing him to work in a more traditional way. The monitoring system is also aimed to be as least intrusive as possible, providing an adequate amount of feedback at all times, thus balancing the need for the movements to be corrected and the focus of the patient on the game. This is possible thanks to the fuzzified alarm level and to the explicit consideration of the alarms previous history.

To allow the selection of the most adequate input device, according to the patient’s possibilities, an input Abstraction Layer has been inserted inside the IGER. This allows us to interface different input devices with the same game. Our prototype currently supports the Sony PlayStation 3 Eye camera, the Microsoft Kinect camera and its microphone array, the Wii Balance Board, and two haptic devices (the Omni Phantom and the Novint Falcon). More devices at a time can be used as input, when needed (cf. Section VI).

Thanks to the control unit module, the therapist does not need to be always present during the exercises, making home rehabilitation a real option. The role of the therapist is still fundamental as he has to prescribe adequate personalized rehabilitation programs and precisely tweak the exercises, by modifying their parameters, to match actual patient status and progression.

VIII. CONCLUSION

We have shown here how different types of intelligent system can cooperate to provide the control of the gameplay while monitoring, at the same time, the player actions. These two functionalities would make rehabilitation at home a real option. Moreover, the Patient Station, implemented according to these principles, is a good match with the guidelines set forth by the EC on personalized health systems. Besides the rehabilitation domain, continuous adaptation of the game level and monitoring of the player activity can both be used to create new generation games that could incorporate the capability of adapting to the player current skills to make gaming experience more engaging.

ACKNOWLEDGMENTS

This work has been partially supported by the FP7 Project REWIRE, grant N. 287713, of the European Union (EU).

REFERENCES