HAL open science

TactiPEd: Easy prototyping of tactile patterns

Sabrina Panëels, Margarita Anastassova, Lucie Brunet

To cite this version:

Sabrina Panëels, Margarita Anastassova, Lucie Brunet. TactiPEd: Easy prototyping of tactile patterns. 14th International Conference on Human-Computer Interaction (INTERACT), Sep 2013, Cape Town, South Africa. pp.228-245, 10.1007/978-3-642-40480-1_15 . hal-01501746

HAL Id: hal-01501746

https://inria.hal.science/hal-01501746

Submitted on 4 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

TactiPEd: easy prototyping of tactile patterns

Sabrina Panëels[1], Margarita Anastassova[1] and Lucie Brunet[1]

[1] CEA, LIST, Sensorial and Ambient Interfaces Laboratory
91191 - Gif-sur-Yvette CEDEX, FRANCE

{sabrina.paneels, margarita.anastassova, lucie.brunet}@cea.fr

Abstract. We present the design and evaluation of a tactile editor, TactiPEd for the rapid and easy prototyping of vibrotactile patterns. It is based on the graphical metaphor of the shape of the device, which is used for the tuning of the main tactile characteristics, including amplitude, frequency and duration of tactile sequences. The editor includes file systems functionalities using the XML format along with playing and recording functionalities. The editor was thoroughly evaluated: a usability evaluation was conducted with 9 participants, the designed metaphor-based patterns were analyzed for insights on cross-device design and finally the editor was tested with several devices. TactiPEd was successfully and easily used with little training and enabled users to design patterns in little time. The resulting patterns shared common characteristics across the devices for a given metaphor.

Keywords: tactile feedback, tactile pattern authoring, vibrotactile pattern.

1 Introduction

The development of wearable tactile devices, in the forms of bracelets [1, 2, 3], vests [4] or belts [5], has been increasing, in particular for mobile and wearable computing. By providing an alternative channel for communication, these devices enable the transmission of information in an eyes-free, ear-free and discrete manner, useful not only in contexts where other modalities are not available but also when they are heavily used, all the while letting the user focus on the surrounding environment. For instance in mobility, it frees the user from staring continuously at his smartphone screen for information whilst walking [6] and thus enables him to stay aware of the potential dangers incoming from traffic of cars and people. The haptic modality has successfully been used for example for providing navigation cues [4, 5, 6], status information about mobile phone applications (e.g. activity of email inbox [1]) or for enhancing the education and learning [7], among many other existing applications.

However, a lot of effort is still spent on designing appropriate, intuitive and discriminable vibrotactile cues, through numerous development iterations and user evaluations [8], often specific to one device and one application, and usually requiring specific device and programming knowledge. Therefore few research prototypes

reach the market or the industry, though they can potentially benefit the areas of accessibility, inclusivity and user mobility. In fact, the general public is only confronted to basic haptic interaction (e.g. vibrations from mobile phones) whereas interface design has been giving increasing importance to user experience [9]. Solutions to integrate user experience in design frameworks are currently being investigated. One such solution lies in making the prototyping of interactions accessible to all, thus enabling a greater involvement of the user, for example using participatory design [10, 11]. This could potentially foster the creativity and intuitiveness [11].

Therefore, to ease the design and testing, and to help promote tactile research and make it more widely accessible, an interface is needed that enables the rapid prototyping and the easy authoring or tuning of patterns, not only by developers but also non-developers, such as ergonomists, designers and users, without the need for specific signal processing or signal communication knowledge. Such an interface should also support several types of devices including multi-actuator devices, i.e. a device with several actuators. Few such interfaces are currently available and widely used. In order to fill this gap, this paper proposes a novel tactile editor, which is based on using parameters that are common knowledge, namely activation, duration, amplitude and frequency as well as a graphical layout of the device for tuning these. The details of its design, implementation and evaluation are further described in the following sections after presenting existing interfaces.

2 Tactile Authoring and Prototyping

2.1 Direct Signal Representation

A few editors have been developed that are based on the direct representation of the vibration signal as editable waveforms. Enriquez and Maclean [12] developed the Hapticon Editor, for the creation and editing of haptic icons for a 1-DOF force-feedback device (e.g. a rotary knob). A pattern could be created by concatenating simple waveforms where length, frequency and amplitude were specified for each waveform and their shape modified through control points. Each icon could be subsequently played on the knob for testing. This work was extended and led to the Haptic Icon Prototyper [13], which uses a more graphical approach with a streamlined interaction sequence for composition of basic haptic icons (see left of Fig. 1).

Similarly, in the vibrotactile domain, Ryu and Choi [8] developed the posVibEditor [8] (see right of Fig. 1). It enables the prototyping of vibrotactile icons and can be used with multiple vibration motors. It also supports the creation of a library of patterns using XML-based pattern files. On the commercial side, Immersion developed the Haptic Studio (previously known as MOTIV Studio), a tactile editor for mobile platforms [14]. It provides templates of haptic effects (e.g. periodic effect) and a timeline interface for their combination into complex vibrotactile patterns. It also has a convenient feature of automatic vibrotactile pattern generation from music files in the MIDI format. It supports a variety of actuators (vibration motor, voice coil, Piezo element), i.e. mechanical devices that convert energy into motion and thus in these cases vibration, but it is unclear if it can be used for a device with multiple actuators.

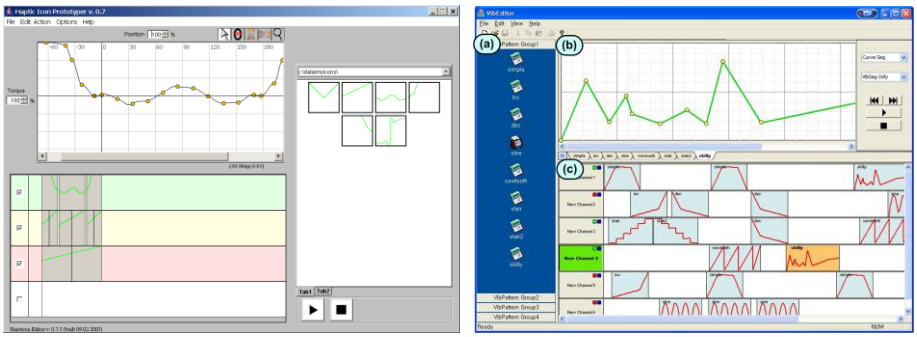

Fig. 1. On the left, the Haptic Icon Prototyper [13], with the waveform editor at the top, the streamlined sequence of signals at the bottom and the palette of haptic icons primitives on the right. On the right, the PosVibEditor [8], with the pattern manager (a) on the left of the image, the editor at the top (b) and the multichannel timeline interface (c).

These low-level interfaces enable finer control over the output due to a direct signal-based mapping. However, they are too low-level and as Lee and Choi [15] mention, they are far from being intuitive and time-efficient and rather ill-suited for non-experts who may lack the understanding required for creating the desired result. This is why another set of interfaces have been geared towards indirect representations or metaphor-based editing of tactile icons.

2.2 Indirect Representation – Metaphor based

Audio – Musical Metaphor. Inspired by the similarity between sound and tactile vibration, Lee et al. aimed to provide a higher-level interface, the VibScoreEditor [16] (see left of Fig. 2), and therefore more widely accessible by using a musical metaphor. The vibrotactile score metaphor can represent the vibration pitch, by vertical location on the staff lines, the duration by the shape of the note and the strength by an integer inside the head of the note. The metaphor and authoring tool were thoroughly evaluated [15]: both programming and scripting by experts and waveform editing (using the PosVibEditor [8]) by non-experts were compared to using the tool. For all qualitative and quantitative measures, the vibrotactile score performed better than the other design methods, both for experts and non-experts. However, the drawbacks of the VibScoreEditor raised by the non-experts participants include the issue of users unaccustomed to musical notation as well as the difficulty to read strength and duration simultaneously. We believe these issues would be exacerbated with multiple actuators, not currently supported by the editor.

Jonas [17] used another graphical representation of music by drawing inspiration from music and video editing software with their timeline interface. In this regard, he developed a tactile editor with a track-based graphical user interface for non-experts, who can simply create and edit patterns by switching motors on and off for certain durations and at certain intensities (see right of Fig. 2). It also supports multiple actuators through Arduino and MakeController platforms. Participants positively evaluated

the tool underlining that the timeline metaphor helped them design and test the vibrotactile patterns.

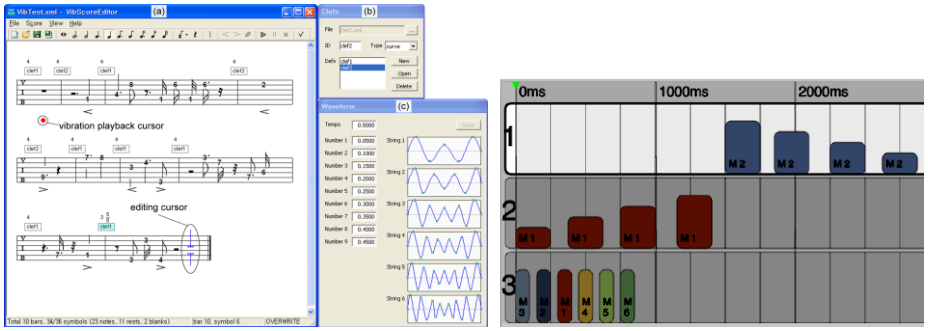

Fig. 2. Left: the VibScoreEditor [16] uses a musical metaphor for editing (main window) that is an abstraction of the physical parameters associated with a vibrotactile signal (right windows). Right: the Tactile Editor [17].

Another radically different and original approach to authoring haptic content is the TECHTILE toolkit [18], which aims to be a prototyping tool for easy and fun design and education of haptic media. It is based on a direct mapping from the audio signal to the vibration. Through a microphone, voice-coil actuators and a signal amplifier, users can easily record the sensation via audio and transfer it to a haptic sensation. First experience with haptics is surely improved, but fine-tuning still requires using audio editing software, which is not necessarily intuitive to all. Moreover, the audio channel is richer than haptics and mapping sounds to haptics is not straightforward, in particular for systems including multiple actuators. However, the transfer between audio and haptics is an obvious source of inspiration for designing haptics patterns; this is why in future work the possibility to transpose audio content to haptic patterns using TactiPEd will be investigated.

Touch Interaction. Given the advent of touchscreen interfaces, the possibility to program patterns directly by mapping touch properties to the vibration parameters has been explored. On the commercial side, the iPhone provides users with the possibility to record a custom-made vibration for contacts by simply tapping on the screen and recording the rhythm. Hong [19] investigated the mapping of user touch input to other vibration parameters to facilitate vibrotactile design. In his tool, duration of motion, touch pressure and the y-coordinate of the touch position are mapped to duration, amplitude and frequency respectively (see left of Fig. 3). This is an interactive and fun way to create patterns according to users; however, due to the low accuracy of touchscreens, gestures are hard to control and difficult for setting exact values. Furthermore, once recorded, a pattern is also difficult to edit as it requires re-recording of motion. This is why the future work proposes a hybrid method combining the 'Demonstration-Based' tool for a first recording and then a waveform editor for finer editing. Cuartielles et al. [20] proposed an editor where haptic patterns can be created

by touching iconographic representations of the body on a touchscreen interface (see right of Fig. 3). Their previous editors used timelines to control each actuator but testing showed that editing the timeline for each haptic stimulus was time-consuming, in particular when designing high resolution haptic patterns (i.e. with a high number of actuators). We share the same point of view and focus rather on the shape of the device to resemble the output device [20] and to be independent of the position of the device on the body (e.g. some wristbands are developed to be placed on any position on the arm, such as the VibroTac [2]).

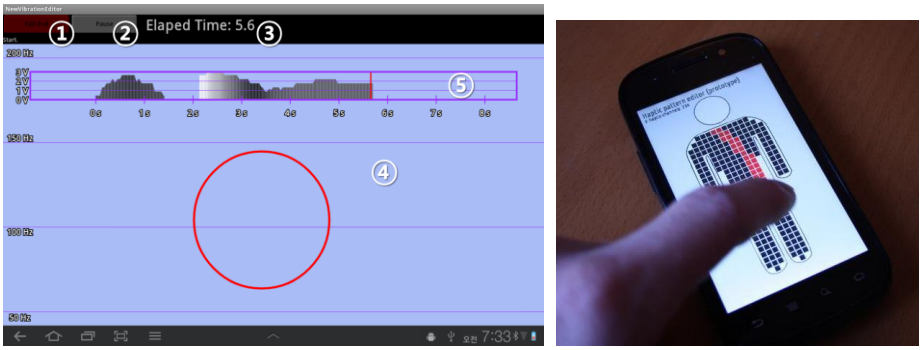

Fig. 3. On the left, tapping is mapped to frequency, amplitude and duration with the Demonstration-based editor [19]. Numbers 1 and 2 refer to editing buttons, 3 to elapsed time, 4 to the touch recording workspace and 5 to the display of the vibration signal graph. On the right, the Visual Editor for high resolution (i.e. high number of actuators) haptic patterns [20].

2.3 Summary

Waveform-based editors enable fine-tuning and precise editing and result. However, they require signal knowledge and are thus inaccessible to non-experts. This is why a number of approaches have investigated other representations, drawing inspiration from audio/music, touch or graphical interactions. However, these techniques usually restrict themselves to the time dimension of the vibration and omit the spatial distribution of the device. We believe this plays an important role in the design of patterns. For instance in navigation contexts, patterns often use the precise localisation of the actuators to form a pattern. A “right” message is often conveyed by a dynamic movement to the right or by simply activating the actuator located on the right of the device and the user. In other cases, some patterns are specific to a device layout and not easily transferrable to any device shape. For instance, for a diagonal pattern on a vest with eight actuators, its transfer to an 8-actuator bracelet around the wrist is not straightforward: what is a continuous diagonal in that case and is the meaning kept in the transfer? Moreover, few of these interfaces support both a variety of actuators and multiple actuators devices. A summary of the different metaphors and actuator support is depicted in Fig. 4.

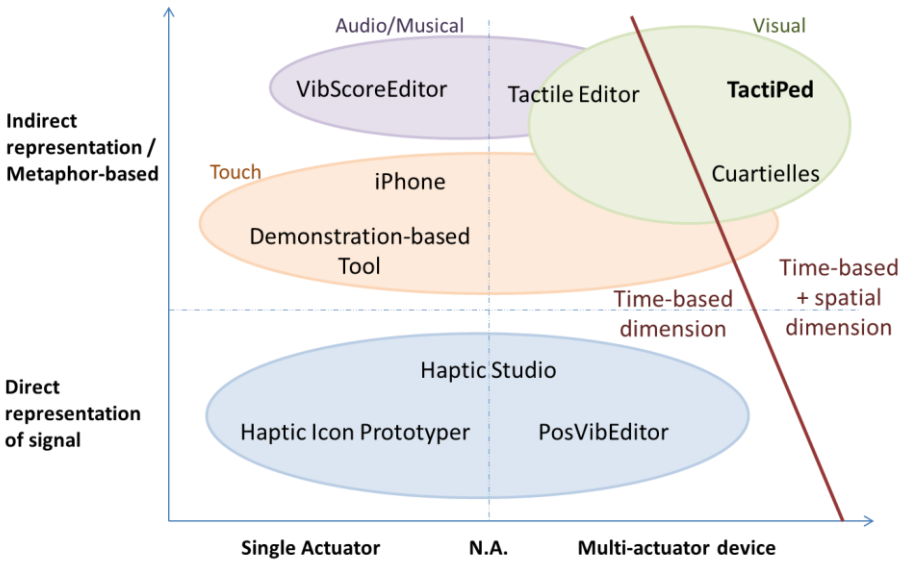

Fig. 4. Summary of the prototyping interfaces by their type of representation and the support for multiple actuators. N.A. refers to the case where the information is not available; e.g. for Haptic Studio the support for a multi-actuator device is unclear.

Therefore, we propose a tactile editor that uses the spatial layout of the device to set actuators activation along with other intuitive parameters such as duration, frequency and amplitude and that supports several types of actuators and multi-actuator devices.

3 TactiPEd: Tactile Pattern Editor

3.1 Overview

TactiPEd aims to provide a high-level interface accessible to non-technical experts (e.g. ergonomists and users) that facilitates the rapid prototyping and testing of tactile patterns. For that purpose, the editor is based on a graphical metaphor using the spatial layout of the device to tune its parameters. It also supports different types of actuators as well as multi-actuator devices so that pattern metaphors can be tested onto several devices. It has been implemented in Python and has been used on the Windows platform. The editor consists of four modules:

- *Graphical user interface*: for creating, editing, opening device templates; creating and editing patterns; and providing access to the other modules.
- *File handler*: for opening and saving device templates and pattern files via XML.
- *Communication module*: for interfacing the hardware, by mapping pattern values from the GUI to the hardware format, which is abstracted in the GUI through a defined device template.

- *Player/Recorder*: patterns can be tested on the fly and also recorded on the device if the hardware allows it.

Each module and its features are further detailed in the following subsections.

3.2 Graphical User Interface

The GUI provides the graphical view to create and edit the pattern and provides the user with access to the rest of the modules through a top bar menu (see Fig. 5).

Fig. 5. Top bar menu of the graphical user interface

The pattern creation and editing relies on using graphical items that can be modified (see Fig. 6 and Fig. 7) using mouse and keyboard interactions. The pattern is created by first opening a device template file (set by hardware constraints), which defines the number of sequences, the shape of the device and the parameters and their values (amplitude and frequency) and is the equivalent of a new file. The maximum duration of a sequence is currently set to 1.6 seconds as it was the maximum value for the devices tested; however it can easily be added as a parameter in the template. For each sequence, the parameters that can be modified are actuator activation, amplitude for each actuator or for all, duration and frequency.

The duration (see region 5 of Fig. 6) can be set by moving a slider, which gives a temporal indication through the length of the associated horizontal bars, similarly to a music equalizer. The actual value is displayed within the horizontal bar.

The amplitude can be either modified for each actuator if these are independent (see region 4 of Fig. 6) or for one sequence with the same value for all actuators (see region 2 of Fig. 7). In the first case, the amplitude values are mapped to a color gradient and set by either clicking on the corresponding actuator on the device shape until the desired value or for faster editing by right clicking the actuator (selection shown with a visual feedback) and setting directly the value via the keyboard. In the latter case, the amplitude is similar to a volume widget with a color gradient and the same mouse/keyboard interactions are used (see region 2 of Fig. 7). When the amplitude is defined per sequence, the activation of actuators is selected the same way as the amplitude per actuator, except with only two values: on and off.

The frequency is displayed within the duration slider to prevent a crowd of widgets using a bitmap with colored bars and white gaps (see region 1 of Fig. 7). The higher the frequency, the smaller the gap between the bars is, and inversely. The same interactions are used for setting the values: touch for successive increase and right click with keyboard value for direct setting.

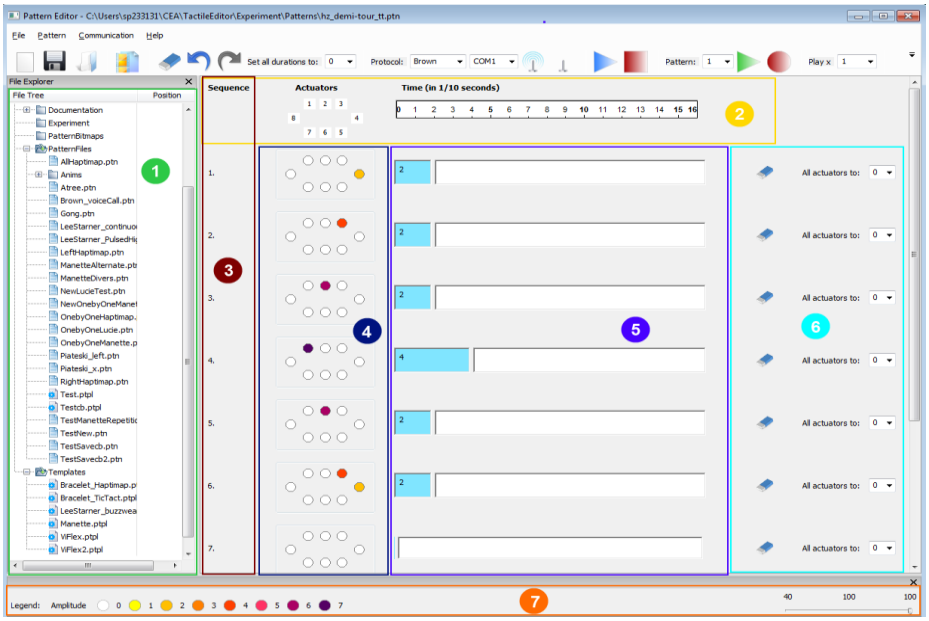

Fig. 6. Screenshot of the interface: Region 1 is a file explorer listing device templates and pattern files. Region 2 is a static ribbon explaining what each column is. The first column (region 3) is the sequence number. The second (region 4) represents the shape of the device where each actuator-circle is a clickable item that changes color according to its value. The next column (region 5) holds the duration bars while the last (region 6) holds widgets for editing (e.g. the eraser resets the sequence). Region 7 contains the legend and zoom.

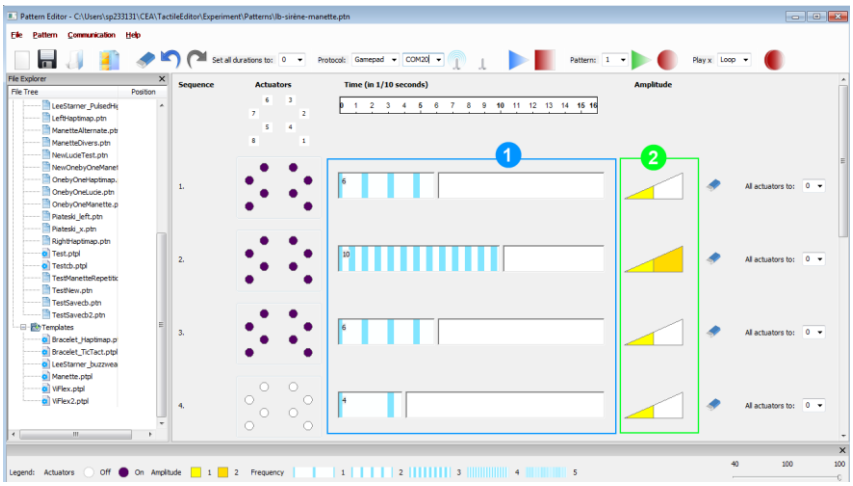

Fig. 7. Screenshot for a pattern tested on a custom built gamepad. Region 1 shows the duration bars displaying the frequency and region 2 shows the widget to set the amplitude per sequence.

The frequency and amplitude can also be copied and pasted by dragging and dropping the values from one sequence to another using the mouse. Finally, a legend is displayed at the bottom to show the matching between colors and values.

3.3 File Handler

The file handling module is based on the XML format for easy management of device templates and pattern files, including saving and opening them. Given the wide use of XML, the files could easily be used in other applications or extended to contain additional information. Small adjustments can also be made directly into the file, if preferred. The pattern file is divided into the definition of the device template (which is the same content as the template file) with the shape, the order of actuators and type of parameters, as described above, and the pattern parameters (activation, amplitude, duration, frequency and repetitions) for each sequence.

3.4 Communication Module and Internal Player/Recorder

The communication module establishes the connection to the device via a defined communication protocol and sends the data appropriately to the hardware. In the GUI (see Fig. 5), in order to test a pattern with the internal player, the user first needs to select a protocol and the port where the device is connected before attempting a connection. Each protocol in the list corresponds to a protocol file. Each protocol file follows a protocol template that implements a defined set of functions used by the GUI, namely *init* (for initialization and opening the connection), *closeSocket* (for closing the connection), *play* (for testing the displayed pattern), *stop*, *playRecorded-Pattern* (for playing a pattern recorded on the device) and *recordPattern* (to record a pattern at a certain memory position on the device). These functions are called upon pressing the corresponding button of the internal player (see Fig. 5).

The protocol list can also be updated 'on the fly' by adding a new protocol. The new protocol simply needs to be written in python by a device expert, implementing the functions cited above, and then added via the GUI. The complexity of the protocol is hidden to the novice end-user who simply uses a device template file to create and edit the pattern, as well as the connection and internal player buttons to test the patterns. The device template file reflects the data required by the protocol template and should also be created by someone knowledgeable. It provides information about the number of maximum sequences that can be sent to the hardware, the different parameters available (amplitude, frequency) and the number of values, and the shape of the device with the order of each actuator for data transmission.

4 Evaluation

The evaluation focused on assessing two main aspects: evaluating the usability of the interface with three user profiles (ergonomists, engineers and non-expert users), and gaining initial insights about the creation of patterns on different devices.

4.1 Methodology

Participants. Nine participants (5f-4m), aged 25 to 60 (*M*=37.67), were recruited to evaluate the tool. Their background ranged from ergonomists (PhD Student to research engineers), engineers in electronics, mechanical design and computer science, and a retired commercial director. Most of the participants use devices with tactile feedback on a regular basis for phone or sports watch notification, for improving immersion in games and in the context of developing and testing haptic devices. Based on their background and subjective level of expertise rating, the participants were divided into three groups: three non-experts with little or no use/knowledge of tactile devices, three haptic engineers and three haptic ergonomists.

Tutorial. After filling a background questionnaire, the participants were instructed about how to use the interface through a tutorial read by the experimenter. This step-by-step tutorial introduced the notion of a haptic pattern, before describing interactively how to create a new pattern using a device template, set its parameters, test it and open/save pattern files. It was available to the user during the experiment along with a one-page visual summary. Two devices were used for the evaluation: a haptic bracelet and a gamepad presented in Fig. 8 and further described below. At the end of the tutorial, the participants were asked to create from scratch a pattern corresponding to the "no" nodding metaphor on both devices to assess whether the participant understood the tutorial and could proceed to the tasks. Metaphors were chosen to describe the patterns to create as they are useful for the design of intuitive patterns [21, 11].

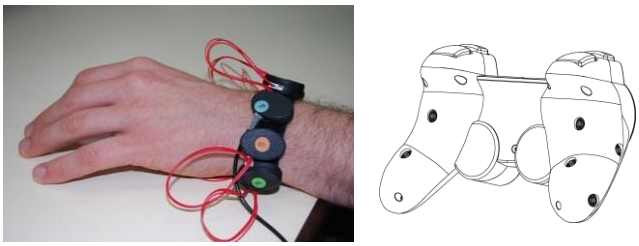

Fig. 8. Left: custom-made bracelet with vibration motors. Right: Gamepad from [7] with 8 electromagnetic actuators (black dots) integrated in the back of an off-the-shelf gamepad.

Devices. The editor was tested with two different devices: a custom-built bracelet [11] and a gamepad developed by Changeon et al. [7]. The custom vibrotactile bracelet (Fig. 8) provides stimulation through eight independent actuators mounted on a velcro band wrapped around the wrist. Each actuator is composed of a commercially available coin motor (Precision Microdrives 310-113), a microcontroller and a power circuit to control the amplitude of the vibration (7 levels). The actuators are linked in series to a supervisor microcontroller that synchronizes the actuation level, the spatial distribution and the timing of the tactile stimuli. The gamepad [7] is composed of eight actuators, with independent frequency and amplitude, and a microcontroller to drive the actuators. Each actuator contains a coil, in a ferromagnetic core, that interacts

with a magnet. The magnetic interaction produced generates a controlled attraction/repulsion force and thus produces a controlled vibration. The main parameters for designing patterns are amplitude per sequence (2 levels), frequency (5 levels) and duration (16 values between 0 and 1.6s).

Tasks. At the end of the do-it-yourself example, the participants were instructed to create patterns from scratch for each of the device based on three different metaphors, an audio metaphor "Siren", a visual metaphor "Turn back" and a tactile metaphor "Stroking". Each metaphor was described with a sentence and an image. They were each chosen as a representative metaphor for each modality commonly used in the design of patterns: audio, visual and tactile. Participants were given the choice about which device and which metaphor to start with. The metaphors were completed in the following order for most participants: siren, turn back and stroking. At the end, participants were asked to fill out satisfaction and cognitive dimensions questionnaires.

Data. Time for task completion, notes about comments and general use, and answers to questionnaires were collected. The experiment took two hours on average.

4.2 Results

Effectiveness. Participants completed all the tasks without difficulty, leading to patterns they felt satisfied with, no matter the device used and the arrangement of actuators. Participants took longer on average to program on the gamepad but a t-test on the log-transformed data (for the assumption of normal distribution) revealed that the difference was not significant between the bracelet (M=245s, SE=45.2) and the gamepad (M=285.4, SE=54.24) with $t(8)=-1.73$, $p > .05$ (see Fig. 9). There are also no significant differences between the user groups (see Fig. 9). Interestingly, some participants commented that it was more difficult to program patterns on the gamepad as the arrangement of actuators was less intuitive than a bracelet.

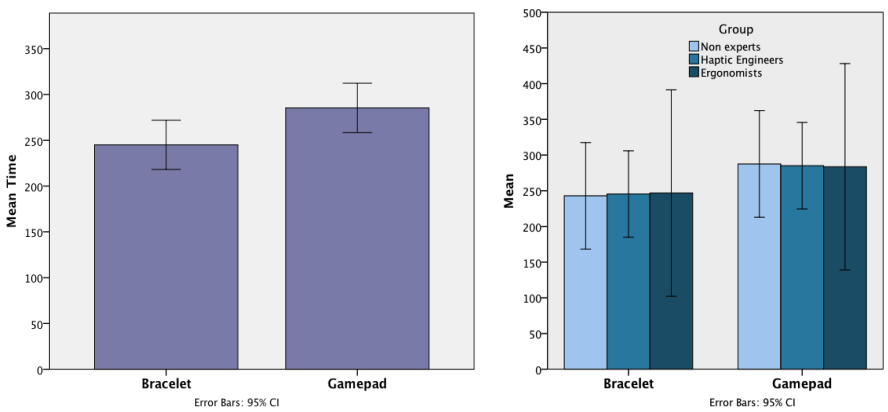

Fig. 9. Results for time completion (in seconds) for each device

Efficiency. The time taken to program a pattern was 265s on average [237s-291s]: patterns were prototyped in less than 5 minutes with the editor. The time taken to create patterns was compared according to the metaphors. We hypothesized that the audio metaphor (Siren) would be the quickest to prototype while the tactile one (Stroking) would be the longest. Indeed, audio can be easily mapped to tactile properties through rhythm and pitch/intensity, whereas a tactile sensation is very subjective and much richer than what current vibrotactile devices provide. The results showed a slight increase in time between the metaphors as we hypothesized (see Fig. 10). We performed a repeated one-way ANOVA on the log-transformed data (for a normal distribution). Mauchly's test indicated that the assumption of sphericity was not violated $\chi^2(2)=3.98$, $p > 0.05$. Though the results are in line with the hypothesis, there are no significant differences between the Siren (M=259.78, SE=42.2), the Turn back (M=261.67, SE=46.5) and the Stroking metaphors (M=274.17, SE=85.5), $F(2,16)=0.36$, $p> 0.05$. There was also no significant effect of expertise, $F(2,6) < 1$ ns. These results show that the editor was used successfully (prototyping < 5min) with no significant differences in time to complete the different tasks for each metaphor.

During the experiment, we noted that participants had more trouble associating the tactile metaphor (Stroking) to a pattern. They thought the shape of the devices was not appropriate and/or the levels of parameters were not fine-grained enough. Interestingly, only the haptic engineers followed our hypothesis and took longer with the tactile metaphor. The non-experts and ergonomists were getting faster in prototyping patterns. This can be explained by the fact that participants were getting more familiar with the interface and with the process of associating a pattern to a metaphor but also in some cases by the lack of ideas for the stroking pattern.

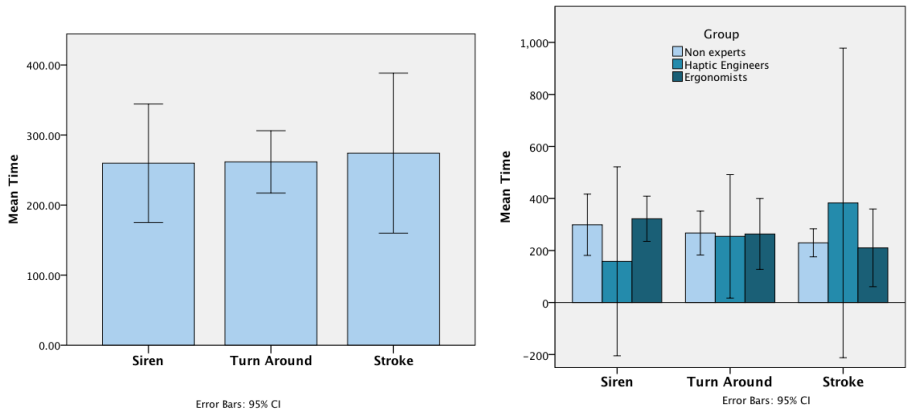

Fig. 10. Results for time completion (in seconds) for each metaphor

To evaluate efficiency along cognitive parameters, we used the cognitive dimensions framework [22], which can be used as an evaluation technique for visual programming environments. It characterizes the system along a set of system properties or cognitive dimensions to illustrate its strengths and weaknesses and trade-offs to real-

ize for effective use. An adapted version of the questionnaire (available online[1]) was provided to the participants with continuous scales assessing each dimension and further questions to elicit feedback and possible improvements. The different dimensions that we used include: visibility (VIJU) for the visibility of the different parts of the system, viscosity (VISC) for the ease of making changes, diffuseness (DIFF) for the verbosity of language, hard mental operations (HMOS) for the demand on cognitive resources, error-proneness (ERRP), closeness of mapping (CLOS) for the closeness of representation to domain, role expressiveness (ROLE) to qualify whether the purpose of a component is readily inferred, progressive evaluation (PROG) for the ability of checking work-to-date at any time, provisionality (PROV) for the ability to play around with ideas, premature commitment (PREM) for constraints on the order of doing things, consistency (CONS) for the expression of similar semantics in similar syntactic forms and secondary notation (SECN) for the possibility to add extra information.

Ratings for each of the used dimensions are depicted in Fig. 11 both on average and for each group of users. Overall the ratings are positive: error proneness (3.2) and hard mental operations (3.6) received low scores while the other dimensions are rated highly, in particular viscosity (8.6), visibility (8), provisionality (8.6), progressive evaluation (8.85) and closeness (8). These are the key features the tool is aiming at, i.e. a clear notation close to the end result where changes can be made easily and tested at any time, letting the user freely play around with ideas. The tool was rated poorly on the secondary notation dimension, justified by the fact that no feature was available to annotate the patterns. This will be added in future versions. It is also worth noting, as expected, that ergonomists were more critical of the system with higher expectations while non-experts were more easily satisfied.

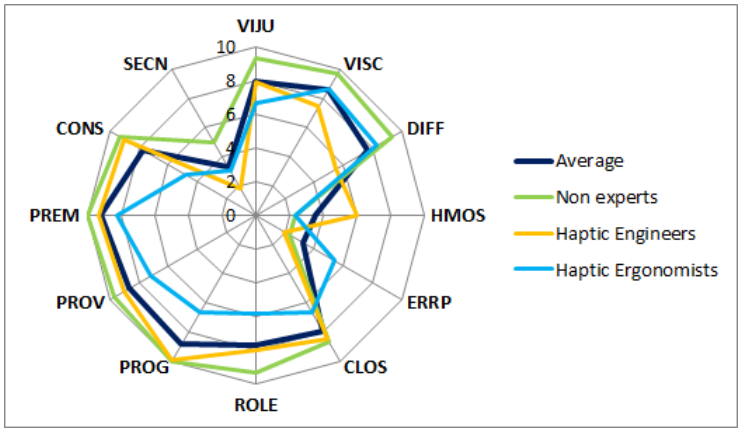

Fig. 11. Averaged ratings for each of the cognitive dimensions scales on a scale from 0 to 10, both overall and for each user group. 0 is generally a bad score except for error proneness (ERRP) and for hard mental operations (HMOS).

[1] http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/

Overall participants successfully used the system in little time and thought it was easy to use and intuitive, but there was still room for improvement with:

- **General layout**: with suggestions on the placement and look of icons. For instance, users requested to change the play button color and position (1 participant), to move it along with stop and repetitions to the pattern window (1) or just repetitions (1), to keep a single icon for connect/disconnect (1), to make the duration bar shorter (1) and to place the "reset all" icon above the reset sequence icons (1).
- **New functionalities**: such as the possibility to add comments (5 participants), the ability to open several patterns for comparisons (4), the ability to convert a pattern from one protocol to another in the possible extent (1), right click on a file directly from the library to play it (1), a visual dynamic representation of the pattern (1) (this feature is already available by generating an animated gif), and copy/paste a pattern from one window to another (1).
- **Interactions**: covering the improvements of existing widget interactions, e.g. the "All actuators to" a chosen value should apply to the ones selected (4 participants) and the "all duration to" to active sequences (1), copy/paste drag and drop of sequences (2 participants), of time (2) or from one actuator to another (1), the ability to move sequences (2), the possibility to click directly in the slider to change time (2) or using the roller of the mouse (1) or the ability to set the duration via the keyboard (1), separating the time from the frequency (1) and changing the interactions and widgets so that there are no differences between amplitude per sequence and per actuator (2).

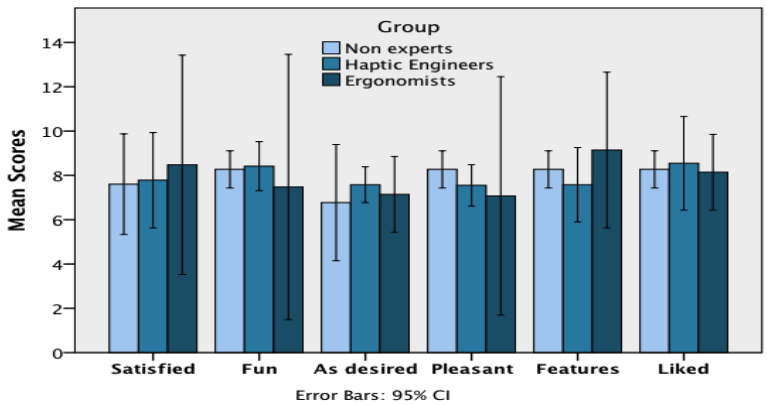

Fig. 12. Satisfaction for each question per user group (out of 10)

Satisfaction. General satisfaction about the interface was evaluated through six statements on a continuous scale from 0 to 10 spanning the general satisfaction of the interface, whether it is fun to use, functioning as desired, pleasant to use and the satisfaction with the features and use. The satisfaction was rated quite highly with an average score of 7.9 out of 10 (see Fig. 12). There is no particular trend according to the user group. Apart from one ergonomist that was critical in her ratings to give room for

improvement (see the high variations in the error bars); most participants were happy with the features and enjoyed using the interface.

Pattern Analysis. During the usability study, results about the properties for designing patterns on each device and for each metaphor were also collected. There were various patterns designed but common characteristics emerged for each metaphor (see Table 1 for a detailed list and Fig. 13 for representative patterns).

Table 1. Description of the main characteristics of the user-created patterns for each metaphor. The number in brackets corresponds to the number of patterns with such features.

Siren	Turn back (U-turn)	Stroking
Activation & Movement: • All (7) – Continuously (2) and with amplitude variation (4) – Intermittent (1) • Circular (1) • Alternating up/down (1) **Amplitude**: mostly constant high (4), ↗ (2) and ↗↘ (2), varied (1)	**Activation & Movement:** • Circular (7) – Alternating half circle (6) – Full circle (1) • All intermittently (1) • Alternating up/down (1) **Amplitude:** mostly medium to high (5), ↗↘ (2), ↗ (1) or low (1)	**Activation & Movement:** • Circular (7) – Half circles (4) – Full circle (1) • Wave around both sides (2) • Slight long vibration (2) **Amplitude:** low (7) to medium (2)
Activation & Movement: • Alternating / Intermittent (7) – Intermittent all (3) – Alternating (3) – Both (1) • ↗↘ the perceived intensity (2) (with number of actuators and/or the frequency and amplitude) **Amplitude & Frequency:** • Both ↗↘ (5) • High amplitude and medium to high frequency (3) • Both low (1)	**Activation & Movement:** • "wave", i.e. actuators playing one after the other (6) – Half circle (3) – Laterally or vertically (3) • All-intermittent (1), • Alternating left to right (1) • Down/side/up motion (1) **Amplitude & Frequency:** • Low to medium frequency (5) • ↗↘ (2) and ↗ (1) frequency • High frequency (1)	**Activation & Movement:** • "Wave" (6) – In one direction (4) – Alternating wave (1) – Both (1) • Alternating sides (1) • All-intermittent (1) **Amplitude & Frequency:** • Both low (8) • High frequency (1)

- **Siren:** Overall, the patterns reflected the sound effect of sirens, either with an increasing/decreasing intense wailing or an intermittent sound similar to alarm clocks.
- **Turn back (U-turn):** Overall, the resulting patterns were continuous movements either mimicking the "visual" representation of a U-turn or the actual physical movement of going back.
- **Stroking:** Similarly to the "turn back" metaphor, the patterns were based on continuous movements. However, they were combined with low amplitude and frequency to imitate the soft continuous movement of a stroking gesture.

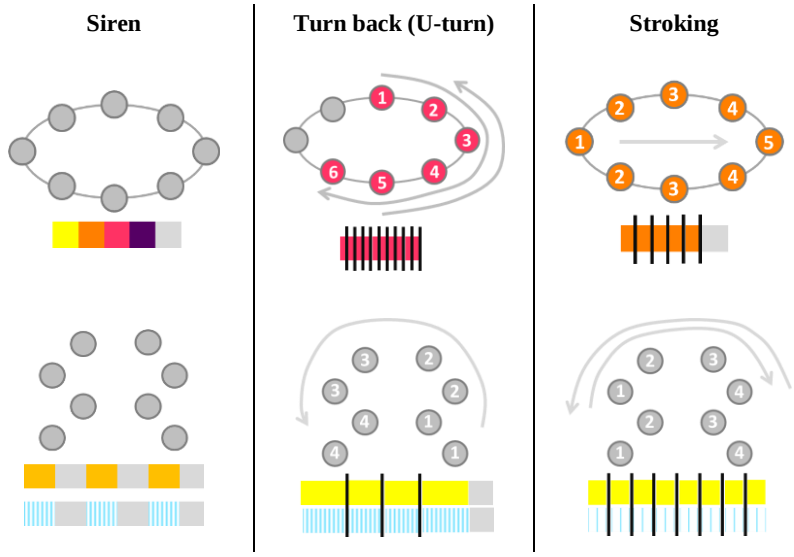

Fig. 13. Examples of the most representative patterns for each metaphor and each device. The grey blocks indicate a pause whereas the black lines indicate a change of actuator with the same amplitude. The colours correspond to the actual values in the interface.

The resulting patterns are, as we anticipated, heavily influenced by the common representations of metaphors, leading to a consensus on the representation, no matter the type of device, as long as they permit it. Given an appropriate set of metaphors, the patterns could potentially be transferrable from one device to another after identifying the dominant parameter(s) related to a metaphor. This would enable the creation of a library of metaphor-patterns that would be suitable to any device. This also supports the hypothesis that original and intuitive patterns can be created from well-chosen metaphors [11]. Further testing with various devices and configurations, various metaphors including ones that can involve localization on the device is needed to investigate further this hypothesis.

5 Conclusion

This paper presented the initial work undertaken to help facilitate the development of vibrotactile patterns and make it accessible to non-experts. It resulted in the development of a visual interface based on the spatial layout of the device, as we believe this plays a role in the pattern design. The interface enables the creation, editing and testing of new patterns on multi-actuator devices and supports various types of vibrotactile actuators. It has successfully been used with a number of custom-developed devices, including a prototype bracelet that consists of six electromagnetic actuators (presented in [23]), and for reproducing patterns described in the literature [4, 3] by adapting the protocol files to match the characteristics of the corresponding devices. The interface can potentially be extended to support any device by providing the corresponding communication protocol and device template. It was also successfully used by nine participants during a usability evaluation that resulted in positive feedback about satisfaction and ease of use, all the while highlighting several areas for improvements (e.g. enabling comparison of patterns, adding annotations, etc.). The resulting patterns provided positive initial insights about the possibility of designing patterns that are transferrable from one device to another.

The future work will not only focus on improving the interface with the collected suggestions but also on investigating other more direct and intuitive approaches to prototype patterns, especially whilst being mobile. A larger study on designing metaphor-based patterns on different devices will also be conducted to validate the initial insights and possibly provide guidelines about a "universal metaphor-based design" of tactile patterns, which takes into account the cultural differences for metaphors.

References

1. Pasquero, J., Stobbe, S.J., Stonehouse, N.: A Haptic Wristwatch for Eyes-Free Interactions. In: SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM (2011) 3257–3266
2. Weber, B., Schätzle, S., Hulin, T., Preusche, C., Deml, B.: Evaluation of a vibrotactile feedback device for spatial guidance. In: WorldHaptics, IEEE Press (2011) 349–354
3. Lee, S.C., Starner, T.: BuzzWear: Alert Perception in Wearable Tactile Displays on the Wrist. In: SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM (2010) 433–442
4. Jones, L.A., Lockyer, B., Piateski, E.: Tactile display and vibrotactile pattern recognition on the torso. Advanced Robotics **20**(12) (2006) 1359–1374
5. Heuten, W., Henze, N., Boll, S., Pielot, M.: Tactile Wayfinder: A Non-Visual Support System for Wayfinding. In: Nordic conference on Human-computer interaction: building bridges (NordiCHI), ACM (2008) 172–181
6. Pielot, M., Poppinga, B., Heuten, W., Boll, S.: Pocketnavigator: studying tactile navigation systems in-situ. In: SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM (2012) 3131–3140
7. Changeon, G., Graeff, D., Anastassova, M., Lozada, J.: Tactile Emotions: A Vibrotactile Tactile Gamepad for Transmitting Emotional Messages to Children with Autism. In: Eurohaptics. LNCS, vol. 7282, Springer Heidelberg (2012) 79–90

8. Ryu, J., Choi, S.: posVibEditor: Graphical Authoring Tool of Vibrotactile Patterns. In: Workshop on Haptic Audio Visual Environments and Games (HAVE), IEEE Press (2008) 120–125
9. Forlizzi, J., Ford, S.: The building blocks of experience: an early framework for interaction designers. In: Designing interactive systems: processes, practices, methods, and techniques (DIS), ACM (2000) 419–423
10. Sanders, E. In: From User-Centered to Participatory Design Approaches. Taylor & Francis Books Limited (2002)
11. Brunet, L., Megard, C., Panëels, S., Changeon, G., Lozada, J.: "Invitation To The Voyage": The Design of Haptic Metaphors to Fulfill Occasional Travelers' Needs in Transportation Networks. In: WorldHaptics, 5th Joint Eurohaptics Conference and IEEE Haptics Symposium (WHC). IEEE Press (2013)
12. Enriquez, M.J., MacLean, K.E.: The Hapticon Editor: A Tool in Support of Haptic Communication Research. In: Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAPTICS), IEEE Press (2003) 356–362
13. Swindells, C., Maksakov, E., MacLean, K.E., Chung, V.: The Role of Prototyping Tools for Haptic Behavior Design. In: Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAPTICS), IEEE Press (2006) 161–168
14. Immersion Corporation: Haptic Studio. http://www2.immersion.com/developers/-index.php?option=com_content&view=article&id=507&Itemid=835
15. Lee, J., Choi, S.: Evaluation of Vibrotactile Pattern Design Using Vibrotactile Score. In: Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAPTICS), IEEE Press (2012) 231–238
16. Lee, J., Ryu, J., Choi, S.: Vibrotactile Score: A Score Metaphor for Designing Vibrotactile Patterns. In: WorldHaptics, Third Joint Eurohaptics Conference and IEEE Haptics Symposium (WHC), IEEE Press (2009) 302–307
17. Jonas, M.: Tactile Editor: A Prototyping Tool to Design and Test Vibrotactile Patterns. Master's thesis, Computer Science Department, RWTH Aachen University (2008)
18. Minamizawa, K., Kakehi, Y., Nakatani, M., Mihara, S., Tachi, S.: TECHTILE toolkit: a prototyping tool for design and education of haptic media. In: Virtual Reality International Conference (VRIC), ACM (2012) 26:1–26:2
19. Hong, K.: Demonstration-Based Vibrotactile Pattern Authoring. Master's thesis, Division of Electrical and Computer Engineering, Pohang University of Science and Technology, Korea (2012)
20. Cuartielles, D., Göransson, A., Olsson, T., Stenslie, S.: Developing Visual Editors for High-Resolution Haptic Patterns. In: Workshop on Haptic and Audio Interface Design (HAID) – Posters and demos. (2012) 42–44 Online at: http://www.english.certec.lth.se/haptics/-HAID12/proceedings.pdf.
21. Pirhonen, A., Brewster, S., Holguin, C.: Gestural and audio metaphors as a means of control for mobile devices. In: SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM (2002) 291–298
22. Green, T.R.G., Petre, M.: Usability Analysis of Visual Programming Environments: a 'cognitive dimensions' framework. Visual Languages and Computing **7** (1996) 131–174
23. Panëels, S., Anastassova, M., Strachan, S., Van, S.P., Sivacoumarane, S., Bolzmacher, C.: What's Around Me? Multi-Actuator Haptic Feedback on the Wrist. In: WorldHaptics, 5th Joint Eurohaptics Conference and IEEE Haptics Symposium (WHC). IEEE Press (2013)