SpeedDial: Rapid and On-The-Fly Mapping of Mobile Phone Instruments

Georg Essl
Deutsche Telekom Laboratories
TU-Berlin
Berlin, Germany
georg.essl@telekom.de

Abstract

When creating new musical instruments on a mobile phone platform one has to map sensory input to synthesis algorithms. We propose that the very task of this mapping belongs in the creative process and to this end we develop a way to rapidly and on-the-fly edit the mapping of mobile phone instruments. The result is that the meaning of the instruments can continuously be changed during a live performance.

Keywords: NIME, Mobile Phone Instruments, On-the-fly, mapping problem

1. Introduction

Mobile devices have rapidly become a viable platform for musical performance. Mobile phone ensembles have been formed [9], there has been an array of efforts to understand and appropriate sensor technologies for mobile music performance [4] and parametric synthesis engines have emerged [3]. There also have been an array of commercial and free software efforts to allow people to make various types of music with mobile devices (Smule's Ocarina, RjDj, ZooZBeat to mention a few that developed out of academic contexts at Stanford, UPF Barcelona and GeorgiaTech). Often currently available mobile music instruments have a specialized scope. Either they implement a specific model instrument, as is the case with Ocarina, or prescribe a specific style of music-making such as pre-scripted music in the case of RjDj, or sequenced music as is the case with ZooZBeat.

Our goal is to maintain the mobile platform as a generic music making device along the lines of a PC or laptop being a generic music platform [4]. This is very much related to efforts of turning the laptop itself into a musical instruments while still allowing much of its flexible power to be used on the fly [1, 8]. Hence the goal is to make the sound generation capabilities accessible to the performer, as well as allow the range of input options to be used. The main

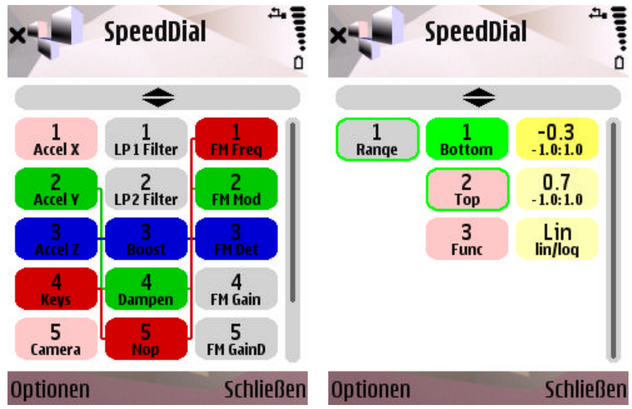

Figure 1. A typical SpeedDial mapping (left). Editing manipulator parameters (right).

task then is to map the input of the performer to the sounding results. Because of the very large space of possibilities of such mappings and the difficulty to associate them with a measure of quality and goodness, this is known as the "mapping problem" [6].

We propose that a way to tackle the mapping problem is to in fact make the mapping part of the creative process. Just as mapping scored music to interpreted performance, mapping individual notes and timbres to musical pieces and so forth are creative processes, we see the mapping of input and gestures to sounds equally as a creative process that is well placed with human beings who are good as dealing with complex relations and come with inherent measures of quality to evaluate the result.

The purpose of this paper is to describe efforts to propose one possible solution to a user interface that is meant to make the mapping of input to sound explicit to the performer and allows this on-the-fly during an ongoing performance. This suggests a certain simplicity of the interface and interactions processes. But we also want to use an approach that are deterministic and reliable and can be executed under time-critical conditions such as life musical performance.

The proposed solution is modeled for the prevailing 12-key dial keyboard plus support keys and was implemented on the Symbian OS 6.0 3rd gen N95 mobile smart phone. The basic input and output capabilities relevant for our purposes are shown in Figure 2.

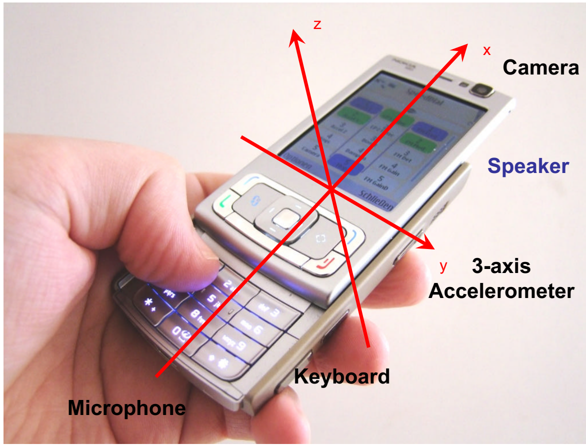

Figure 2. Basic input (black labels) and sound output (blue label) capabilities of the Nokia N95. Tilt axis of the 3-axis accelerometer are depicted in red.

2. Designing Rapid Mapping

In order to guide our design we first explored goals that are desirable for on-the-fly mapping of input sensors to synthesis algorithms given the constraints of the device. These include

- Does not disrupt audio playback: Editing needs to be possible while current sound processing is on-going. Hence editing needs to be concurrent.
- Allows very rapid mapping: Meaningful mappings should be possible with very few performer interventions.
- Use reliable sensing when accuracy and control is important: Live performance is stressful and error prone. The interface needs to be robust to ensure that it can be operated on with confidence.
- Fast recovery of slips: Errors happen in live performance. A main concern here is that errors can be quickly reversed when identified.

Given these constraints we made the following basic design decisions:

- Use keys for critical interactions. These provide fairly fast and safe discrete interactions.
- Limit the number of key sequences needed to finish any task.
- Immediate undo at any discrete step.
- Limit the complexity of possible mappings.

Further constraint on the design is the size of the screen. In the current design we only consider Nokia N95 smart phones which have a 240x320 pixels on 4cm by 5.3cm. This is a comparatively large screen historically, however smart phones rapidly evolve to increase screen real-estate as can be seen with the iPhone and other emerging smart phones.

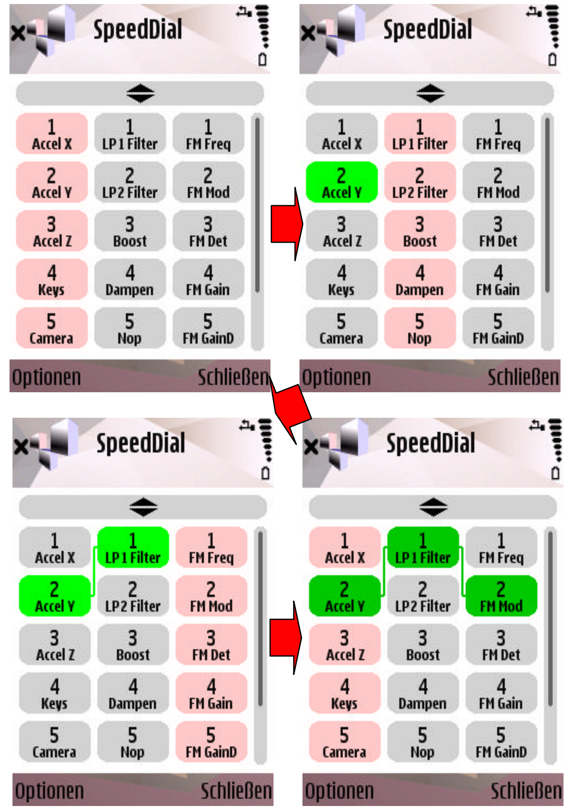

Figure 3. Basic mapping steps of a mapping in SpeedDial. The highlighted column is the next to be set. Currently created but incomplete mappings are bright green. Other mappings use various colors of reduced brightness (see Figure 1 (left)).

2.1. Sensor to Synthesis Mapping

The task of creating a musical instrument is to map sensing capability to sound output. Hence it was clear that there needed to be a way to connect those two. Our design chooses to place one intermediary stage, which can serve to filter, manipulate, or give semantic meaning to sensory input and hence condition it in various abstracted ways for a synthesis algorithm. The primary mapping mode hence consists of three parts, the input dimensions (*source*), which are displayed in the left column, the manipulation dimensions (*manipulators*), which are found in the center column, and the synthesis dimensions (*sinks*), which occupy the right column (see Figure 3). A sensor, a manipulation algorithm and a synthesis algorithm itself can be comprised of multiple dimensions. For example an 3-axis accelerometer can provide acceleration along the x, y and z axis. We generally split these multi-dimensional components into 1-dimensional part, however this is not required and there are some meaningful exceptions that simplify mappings. A manipulation algo-

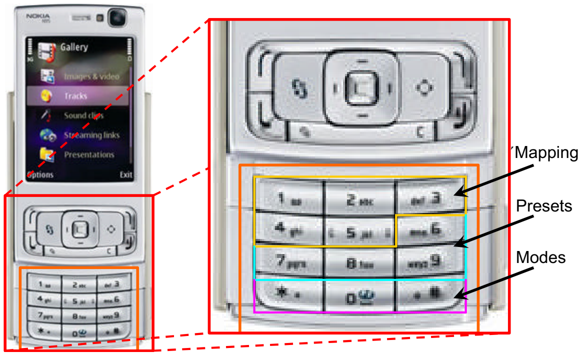

Figure 4. Basic mapping steps of two mappings in SpeedDial. The highlighted column is the next to be set. Currently created but incomplete mappings are bright green. Other mappings use various colors of reduced brightness.

rithm may be a lowpass filter whose controlling dimensions are input stream and cut-off frequency. The performer can choose to map each individually. The same holds for synthesis algorithms. For example an FM algorithm can expose base frequency and modulation index as separate dimensions and each can be mapped to individually.

The basic layout allows for 5 rows to be displayed per dimension, if this many are available in the current configuration. If more dimensions than 5 are available within a column, scrolling is possible.

2.2. Key Mapping Design

Hence the primary input device becomes the 12 key standard mobile phone block depicted in Figure 4.

We considered two basic rapid typing layouts: One uses a number block layout corresponding to the visual display of the mapping. Early evaluations show that this always forces finger motion, larger overall finger motion and relatively high subjective fatigue. Hence we settled on an ordered paradigm where we used the first 5 keys in sequence, and reused them for each mapping stage. Overall movement was reduced, and key hit repetition was possible for simple mappings that happen to align, making this choice less strenuous subjectively. This is also justified theoretically by Fitt's law. Using just 5 keys of the keyboard reduces the overall area to cover and the average moving distance between key-presses. Pure dialing speed is very rarely studied, while the area of text input using 12 key arrangement has a fair body of literature [5]. A further effect of this use of keys to cell selection is that we have spare keys to map to additional capabilities.

3. Offline and Online Editing

On of the main goals is to limit the amount of interactions. Hence also the amount of complexity that can be directly manipulated is limited by the choices that can sensibly be presented. That is, in order to still give detailed editing possibilities we therefore separate two parts of the process. One is an offline editing stage in which the performer can prepare aspects of the setup before a performance. Primarily this is concerned with two things. The first is to allow the choice of sensors, manipulators and synthesis parameters which are intended to be actually used and hence define performance presets. The second provides ways to specify parameters that are not intended to be manipulated on the fly. For example a performer may intend to manipulate the base frequency of an FM algorithm but leave the modulation index constant. This constant can be edited offline with high accuracy (see Figure 1 (right), parameters with a highlight border are used in the current preset for online mapping).

The primary editing mode is however the 3-stage mapping display already described above. However the offline editing can directly contribute to the online performance through the presets. Presets can be accessed through keys 7 through 9, and multiple pages of presets can be iterated through via the 6 key. Hence if 5 presets are available, initially the first 3 are mapped to keys 7 through 9 and upon pressing 6 the remaining 2 are mapped to the same keys. This allows the rapid changing of complex mappings in one key-stroke. The 0 key appends the current configuration to the preset list allowing fast recall of mapped presets that were generated on the fly.

The online mode itself allows manipulation and creation of mappings on the fly, that is mappings that are exploratory and evolving in nature.

Finally whole configurations, which consists of current presets and current online mappings can be saved to the phone's file system and loaded back from there from the file menu, which also contains options to toggle audio playback and exit the program.

4. Keypad Performance

The keypad plays a double role in SpeedDial. It is the primary means to manipulate mappings, but it also can be used as input dimensions for a performance mapping. In order to support this, we use the * key to toggle between mapping and performance mode with respect to the keypad. If in performance mode, keypads no longer affect the mapping UI but are directly used within the currently active mapping patch for performance.

5. Sources, Manipulators and Sinks

We call the individual blocks that can be linked together units. There are three types of units: sources, manipulators and sinks. Sources are sensors and input capabilities. Manipulators are algorithms which modify the data arriving from the linked sources or use it to modify a parameter of the manipulator Sinks are are parametric components of a synthesis algorithm. Typical examples of sources are: each axis of an accelerometer, keys of a keyboard, microphone signal [7], camera signals, and bluetooth protocol signals. Sources can be direct signals or signals modified to yield a single data stream. For example multiple keys can be modified to form a range of responses of multiple keyboard and serve as one source. Camera images can be converted to overall brightness and be used as single stream sources in this fashion. In our first prototype we primarily focused on accelerometers and keys as input modalities to illustrate the architecture. Sinks are typically parameters of synthesis algorithms. Here we use FM synthesis, additive synthesis and sample-playback to illustrate the principle, but any synthesis algorithm is thinkable as a collection of sinks. The most interesting aspect are the manipulators. These serve a range of functions to give meaning to the sensing data. The trivial manipulator is called "nop" and performs no actions on the signal and just passes it through. But manipulators can act on the gain (dampen and boost) act on the signal range (ranger) or filter the signal in various ways (LPFilter, HPFilter, BPFilter). It can also act in more semantic ways, for example through signal gates which pass signals only in a certain range, or half-wave rectification (rect), thresholders (thresh), or onset-detectors (onset) can be used to detect certain aspects of the control signal and modify it to control synthesis.

6. Limitations of the System

On of the primary limitations of the system is floating point operation performance. Most implemented and openly available synthesis algorithms on smart phones use floating point operations extensively [3]. This does limit the amount of concurrent renderings that are possible without seriously degrading performance of the system. Primarily for this reason we have artificially limited the number of active concurrent synthesis algorithms to one. The setup itself is able to handle an arbitrary number of these and many more can be mapped in principle. To relate multiply mapped synthesis algorithms in our interface we have implemented the policy that the latest mapped synthesis algorithm is active. Hence one can rapidly transition between synthesis algorithms by either mapping a new one or deleting the current one.

7. Conclusion

In this paper we explore the design philosophy that the *mapping problem*, i.e. the task of finding a good relation of sensor capability to synthesis algorithms should be interactively exposed to performers for use in a realtime setting. We believe that mapping can inherently be part of creative performance. To this end we proposed an interface that is designed to allow rapid mapping in an interactive setting, for smart phones with the standard 12 key layout.

Future work include rapid editing for alternative mobile phone input technologies. In particular single- and multi-touch screen interactions become increasingly popular. In some cases they come hybrid with hardware keyboards of various sizes and configuration (whether 12 key layouts (N96) or larger keyboard layouts to support text typing (Blackberries)). Some platforms exclusively focus on touchscreen interactions (such as the iPhone). While one could translate the method proposed in this paper quite literally to all of these setups, via mapping to larger keyboards, or via virtual key areas on the touch screen, we believe that interaction techniques should be closely related to the physicality of the primary input technology [2]. Hence we suggest that for example multi-touch based phones require an alternative paradigm and we are currently working on a design for it. Furthermore we have so far focused on the user interaction of the system. The integration of synthesis and language paradigms is not yet fully developed. For example one may want to plug in synthesis methods, maybe via VST plugins. Or one my define blocks via PD patches or ChucK scripts. How to best specify such capabilities also is future work.

8. Acknowledgments

The author would like to thank Michael Rohs for early discussions on rapid key mappings. Some of the synthesis algorithms used in the prototype were developed by Ge Wang and the author for the Stanford Mobile Phone Orchestra.

References

[1] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*, 8(3):321–330, 2003.

[2] G. Essl and S. O'Modhrain. An Enactive Approach to the Design of New Tangible Musical Instruments. *Organised Sound*, 11(03):285–296, December 2006.

[3] G. Essl and M. Rohs. Mobile STK for Symbian OS. In *Proceedings of the International Computer Music Conference*, pages 278–281, New Orleans, Nov. 2006.

[4] G. Essl, G. Wang, and M. Rohs. Developments and Challenges turning Mobile Phones into Generic Music Performance Platforms. In *Proceedings of the Mobile Music Workshop (MMW-08)*, Vienna, Austria, 2008.

[5] T. Klockar, D. Carr, A. Hedman, T. Johansson, and F. Bengtsson. Usability of mobile phones. In *Proceedings of the Int. Symposium on Human Factors in Telecommunications*, pages 197–204, 1-4 December 2003.

[6] E. R. Miranda and M. M. Wanderley. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard.* AR Editions, Middleton, Wisconsin, 2006.

[7] A. Misra, G. Essl, and M. Rohs. Microphone as Sensor in Mobile Phone Performance. In *Proceedings of the International Conference for New Interfaces for Musical Expression (NIME-08)*, Genova, Italy, 2008.

[8] G. Wang and P. R. Cook. On-the-fly programming: using code as an expressive musical instrument. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, pages 138–143, Hamamatsu, Japan, 2004.

[9] G. Wang, G. Essl, and H. Penttinen. Do Mobile Phones Dream of Electric Orchestras? In *Proceedings of the International Computer Music Conference (ICMC-08)*, Belfast, UK, 2008. ICMA.