Aalborg Universitet

States and Sound

Modelling User Interactions with Musical Interfaces

Larsen, Jeppe Veirum; Knoche, Hendrik

Published in:
Proceedings of the international conference on new interfaces for musical expression, Copenhagen, Denmark, 2017

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

States and Sound: Modelling User Interactions with Musical Interfaces

Jeppe Veirum Larsen
Department of Architecture, Design and Media Technology
Aalborg University
Rendsburggade 14
9000 Aalborg, Denmark
jvl@create.aau.dk

Hendrik Knoche
Department of Architecture, Design and Media Technology
Aalborg University
Rendsburggade 14
9000 Aalborg, Denmark
hk@create.aau.dk

ABSTRACT

Musical instruments and musical user interfaces provide rich input and feedback through mostly tangible interactions, resulting in complex behavior. However, publications of novel interfaces often lack the required detail due to the complexity or the focus on a specific part of the interfaces and absence of a specific template or structure to describe these interactions. Drawing on and synthesizing models from interaction design and music making we propose a way for modeling musical interfaces by providing a scheme and visual language to describe, design, analyze, and compare interfaces for music making. To illustrate its capabilities we apply the proposed model to a range of assistive musical instruments, which often draw on multi-modal in- and output, resulting in complex designs and descriptions thereof.

Author Keywords

Sound; Assistive Musical Instruments; ADSR; Three-State-Model; Modeling; Gestures.

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, I.6.5. [Simulation and Modeling] Modeling Methodologies

1. INTRODUCTION

The unambiguous and complete description of interactions with and feedback from musical interfaces is important for 1) designers to analyze and publish their designs, and 2) researchers intending to compare different designs and reproduce results from interactions with these interfaces. Frameworks and taxonomies for musical interface design have been proposed before [2, 7, 12, 14]. However, publications of novel interfaces still lack the required detail due to the complexity of the focus on a specific part of the interface and the absence of specific templates or structures to describe these interactions for easier comprehension and visual comparison. The interactions and feedback the musical interface provide can be difficult to describe with the existing vocabulary, which prompted Buxton to formulate his three-state model of interaction for input devices. However, musical instruments include a temporal course of sound that cannot be described by Buxton's model and its extensions alone; rather, it requires a temporal notion. We suggest a way of modeling musical instruments that draws on and synthesizes models from interaction design and music making. We apply the proposed model to a range of examples in the domain of assistive musical instruments, which often draws on multi-modal in- and output that complicates designs and their descriptions. All of these IMEs lack interaction/feedback details in their respective publications, which are representative of publications/descriptions of IMEs in general. The analysis of the resulting model allows for a visual comparison of the interfaces in terms of where and how input can manipulate the expressive parameters of sound and which feedback modalities the system employs when transitioning between states.

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

2. BACKGROUND

Buxton's Three-State Model of Graphical Input [3] introduced a vocabulary and modeling template to better describe interactive techniques vis-a-vis the technologies that implement a graphical user interface. His model draws on the notion of finite state machines consisting of labeled states (circles) and transitions (arrows) between them that describe how user input (labels on the transitions) from one input device changes the state of a system (see Figure 1). State 0 denotes an out-of-range state in which the user has not acquired the input device or control, state 1 allows for movement of a cursor (tracking), and state 2 allows the manipulation of objects (dragging). The transitions between states model discrete events, whereas the self-loop transitions model continuous input or non-input (in state 0).

Many instruments involve more than one input device or extremity. Hinckley et al. extended Buxton's model to

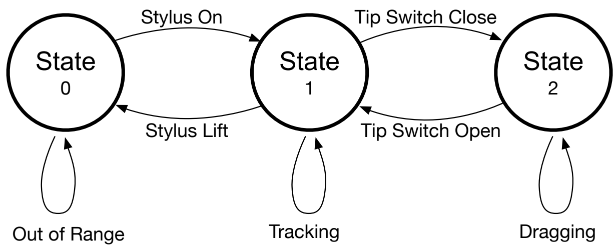

Figure 1: Buxton's Three-State Model with stylus and a tablet.

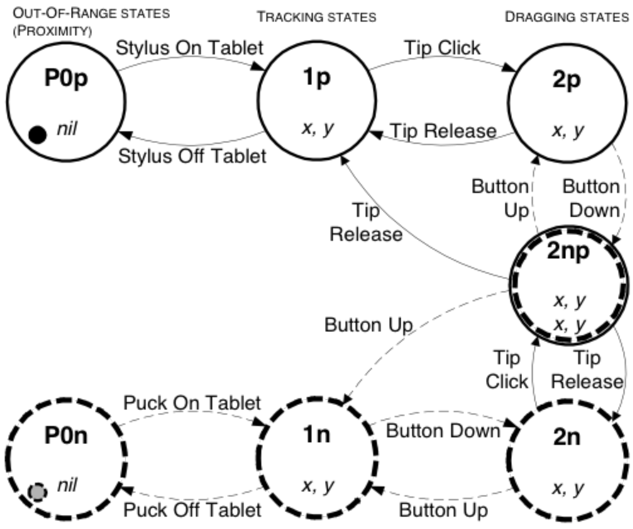

Figure 2: Hinckley et al.'s two-handed input example of stylus and o puck on a tablet.

address a wider range of design problems, multiple effectors through input devices, and interaction technologies [8] by drawing on Petri net representations [15] and including continuous properties. Hinckley's model uses tokens (represented as circles inside states in Figure 2) to express which state the system is in. The tokens can move along through the transitions to states that have the same outline (solid or dashed). Instead of Buxton's self-loops, (see arrows under each state in Figure 1), the model relies on the notion of sensing continuous input, like position, angle, force, or torque, within a state expressed through a named *italicized* property in the lower half of the state in Figure 2. Hinckley further added a prefix to state 0 to distinguish between the two out-of-reach states - touch (T0) and proximity (P0). Different formatting (dashed or solid lines Figure 2) of the states, tokens, and transitions indicate the devices. A state name postfix distinguishes between the respective effectors, i.e. input hands (p for the preferred and n for the non-preferred hand in Figure 2).

Apart from providing a language and notation for user interface interaction concepts, state modeling allows for visually inspecting the model and spotting asymmetries in the design in case certain states exhibit different behaviours [20]. Figure 1 is a case in point of symmetry, and Figure 2 illustrates that state 2np is special in terms of the larger number of transitions to and from it and its overlap.

Obrenovic and Starcevic created a modeling framework for specifying multimodal systems. Their model draws on the Unified Modeling Language (UML) and focuses on the inner workings and effectors of the system and its modalities [13]. Our work focuses on modeling the states the system can be in, how user input affects transitioning between states, and the feedback they receive from the system while doing so.

Birnbaum et al. suggest a dimension space for musical devices using a spider web representation [2]. The axes in the spider web have different representations, e.g. required expertise, musical control, feedback modalities. The axis values vary, e.g. high/low, none/extensive, few/many, etc., depending on what they are describing. Hattwick and Wanderlay further expand the dimension space for evaluating collaborative music [7]. Vertegaal and Ungvary investigate the relationship between body parts, transducers, and feedback modalities [21] in music controllers. Overholt presents the Musical Interface Technology Design Space (MITDS), which provides a theoretical conceptual framework and guidelines for describing, analysing, designing, and extending the interfaces, mappings, synthesis algorithms, and performance techniques for interactive musical instruments [14]. Morreale et al. also present a conceptual model called MINUET, which offers a way to understand the elements involved in musical interface design [12]. Most of these frameworks, however, do not provide a sufficient graphical representation of the musical interfaces and do not model different states and feedback to user interactions. MIDI, for example, does not concern itself with how users actuate sounds and what feedback the system provides apart from the generated sound.

2.1 Musical Control and Expression

A number of major components are used to describe musical control and expression. For music making, the *attack*, *decay*, *sustain*, and *release* (ADSR) envelope [16] describes the volume of a generated sound over time (c.f. Figure 3). We draw on Swink's visual depictions of these ADSR parts (the arrows) in our musical interface models. Goldstein [5] used a state transition diagram to model both sustaining and percussive instruments. The diagram describes how an instrument produces sounds and the different modes of control. Francoise et al. [4] also used ADSR to decompose gestures into four phases of sound control: *preparation* (P), *attack* (A), *sustain* (S), and *release* (R). Levetin et al. described musical control through more explicitly detailed steps called: *beginning*, *middle*, *ending*, and *terminus* [10]. These steps roughly map to the ADSR envelope. The *beginning* combines ADSR's *attack* and *decay*, the *middle* maps to *sustain*, and the *ending* paired with *terminus* makes up *release*. Levitin's beginning distinguishes how energy enters into the system through either *continuous excitation* (CE) or *impulsive excitation* (IE). Continuous excitation stems from continuously, e.g. bowing or blowing. Plucking a guitar string or pressing a key on a piano yields impulsive excitation. Levitin further classified instruments into two types of middle: the non-excited middle (NEM), e.g. a guitar, and the continuous excited middle (CEM), e.g. an organ. During *middle* (*sustain*) CEM instruments allow for gestures to control expressive parameters such as pitch, loudness, and timbre. NEM instruments usually do not support these manipulations during this step since the musician cannot manipulate the energy source. NEM and CEM instruments further differ in how a note can end. Musicians of NEM instruments, e.g. a guitar, can either let the impulse energy reach *terminus* (no sound) through *gradual decay* or actively *terminate* the note by muting the string. CEM musicians cannot employ *gradual decay* as the energy abruptly ends when the musician stops bowing or blowing.

2.2 Assistive Musical Instruments

Advances in technology have created opportunities for new assistive interfaces that make musical instruments accessible to people with impairments. Such assistive musical instruments have to overcome different challenges depending on the type of impairments [6, 9, 11]. Obrenovic classified constraints that hamper accessibility as user, device, environment, and social constraints. Users can be impaired in terms of their senses, perception, motor or linguistic skills, and cognition [13].

Designing musical interfaces for people with impairments requires design tools and a language to ensure the best solution and to convey the design in a clear and detailed fashion. We follow Buxton's lead and argue that state models are a

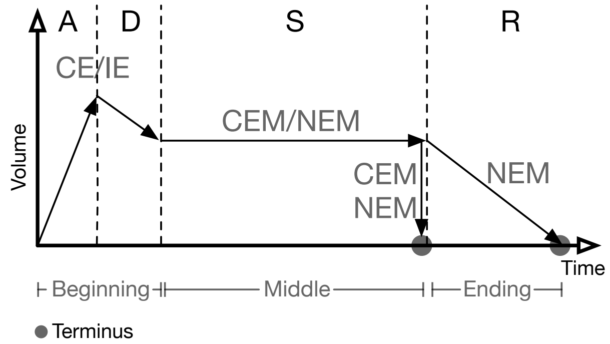

Figure 3: The ADSR Model (black text and arrows) overlaid with Levetin's stages (gray text labels) below and on the ADSR arrows.

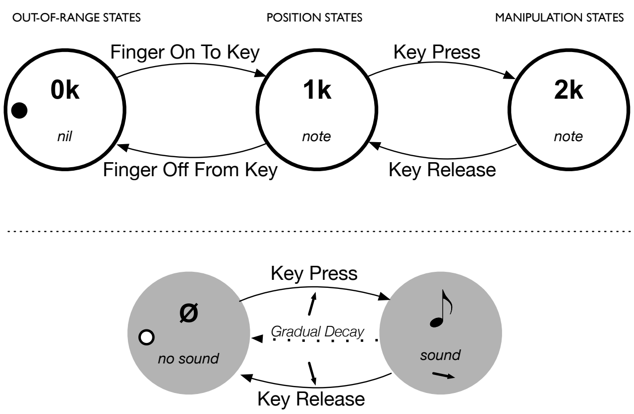

Figure 4: A piano key (CEM instrument) with sound states.

good basis to start from in this case. While our proposed model can be applied to any instrument, we believe that designing for impaired users, which requires attention to detail about which modalities are employed as input and feedback, can benefit especially from a tractable tool and the rigorous approach it promotes.

3. MODELING MUSICAL USER INTERFACES

The model in Figure 4 shows the physical properties of a piano key, but due to the temporal quality of music - specifically for NEM instruments - the model needs to be able to express time. If we imagine an organ instead of a piano we would have a continuous excited middle and thereby infinite sustain. When the organ key is pressed down we have the attack of the note and transition to *state 2k*. As long as the model is in state 2 we have sound, but on release, i.e. once we lift our finger off the key, there is no sound as we transition back to state 1 or state 0k. Just as with the organ, after the attack the piano is in state 2k but with a piano we have a non-excited middle. This results in a *gradually decaying* sustain, which will eventually fully decay while the key is still down. In that case, we would end up with no sound but still be in state 2k. Timed Petri nets [22] model time in the transitions, but this does not align with the temporal behaviors in music making such as preempting a current guitar chord with a new strum. Therefore, we propose to model the temporal course of a generated sound in states and transitions in a separate sub-model for sound. We will from hereon refer to the sub-models for sound and the interactions as the sound model and interaction model.

In the sound model, based on and-states [20], we reuse the transition labels from the interaction model to make explicit how user input changes sound. To visually distinguish between the two sub-models we include a dotted line between the models and a gray fill for the sound model states. To specify the temporal properties of sound we incorporate the ADSR stages on the transitions together with the labels. We use attack (↗) for the onset of the note, sustain (→) for the length of the note, and release (↘) for the end of the note. These are not used on the interaction model as that would be redundant information that would bloat the model. Sustain is a special case when trying to model a musical instrument. The white token indicates what state the sound model is in. On *key press* and (↗) we transition from *no sound* to *sound*. When in *sound* there are two ways to transition to *no sound* depending on Levitin's type of middle. The first is to release a key, dampening a string or stop bowing. This would cause a (↘) and stop the sound moving us back to *no sound*. The second possibility is through *gradual decay* when a key or a string is pressed down until the sound decays fully, see Figure 4. To describe what type of middle or sustain the musical interface or modelled instrument has, a horizontal arrow is used for CEM interfaces and a slightly tilted arrow is used for NEM instruments. Preempting chords and gestures are shown with a curled arrow returning to the same state just as the loops in Buxton's original model. The loops use the same type of line and color as the effector to which they belong.

3.1 Logical and Relational Conditions

To better explain and control the flow and transitions in our model we draw on logical and relational conditions to express exceptions and special cases throughout the interaction with a given musical interface. We also add a new effector so that our model consists of a single piano key, a sustain pedal, and the sound states. When playing a piano releasing the key dampens the sound, but a sustain key can avoid stopping the sound when releasing the key. To be able to model this we have used *if* and *NOT* to express when we get a release and transition to *no sound*.

3.2 Feedback

In Figure 5 we have added information about the type of auditory, haptic, and visual feedback the system provides in states (at the bottom) and during transitions (on the inside of the arc). Assistive devices can often benefit by improving or adding additional feedback to better signal when certain interactions occur that otherwise would be missed or cause the user to doubt. We focus on the auditory (illustrated through an ear), haptic (hand), and visual feedback (eye) shown as icons.

4. ANALYSIS OF ASSISTIVE MUSICAL DEVICES

In this section we apply our suggested model to analyze five assistive musical interfaces to illustrate its expressive capabilities. We highlight both the value in the design stages of a musical interface and in the analytical or comparative stages to ensure a complete description of a MUI.

4.1 Soundbeam

Soundbeam (SB) is a commercially available assistive musical (NEM) instrument using an ultra-sonic range finder

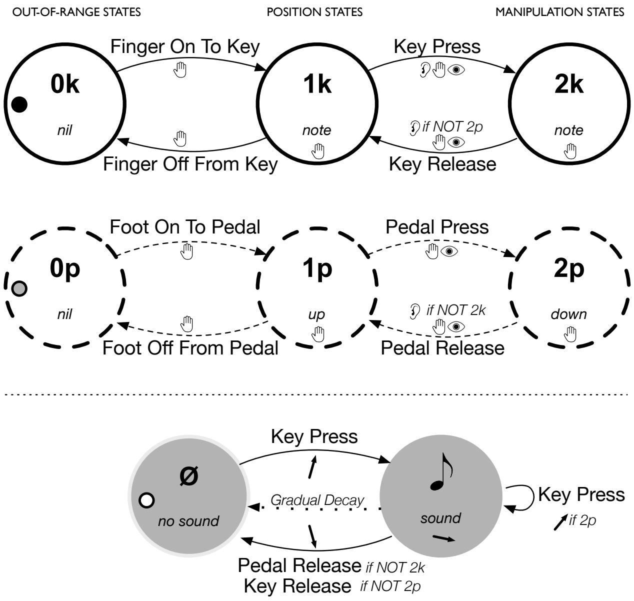

Figure 5: A model of a piano key and a sustain pedal with the use of logical, relational conditions and icons for type of feedback.

(range between 23 cm and 6 meters with a conical shape of diameter of 90 cm and height of 4 meters) to expand an area in front with virtual notes [18, 17]. By interrupting the invisible beam, e.g. with a body part, the user triggers a note, the discrete pitch of which depends on the distance to the range finder. We have modeled SoundBeam in one of its nine settings called multi, and Figure 6 illustrates the absence of a state 1.

As a touch-less device using an ultra sonic range finder it plays one note when breaking the beam at a given distance to the sensor and a different note at discrete distances as it moves closer to or farther away from the sensor. There is no intermediate or positioning state like the piano key and sustain pedal in Figure 5 and therefore no state 1. Soundbeam gives only auditory feedback at the attack/onset of a note; it uses no other modalities or feedback. The sound states in Soundbeam are different from other instruments. It provides no gesture for stopping a sound, so the only way to stop a sound is to let the note decay fully. It has no resting (position) state but is either on or out of range.

4.2 Movement-To-Music

The movement-to-music (MTM) prototype [19] - a CEM instrument - combines exercising and music making. It uses computer vision to capture body movement, renders the user's body on a screen, and superimposes a number of colored shapes around the user. When the rendered body intersects with the shapes, they change transparency and trigger a musical note.

The Movement-To-Music Instrument is like the Sound-Beam - a touch-less device without any haptic feedback. MTM can track more than one limb, but every limb would have an identical but independent model. So we only model one effector here as an example. Unlike Soundbeam, MTM has a state 1, which tracks the coordinates of the user, see Figure 11. In state 1 the instrument gives visual feedback when tracking the user and when entering and exiting the predefined trigger regions. The sound state illustrates a continuous instrument where the attack and onset of the sound start when entering the region and stop when exiting.

4.3 The Actuated Guitar

The Actuated Guitar (AG) [9] allows people with hemiplegia to play a real electric guitar. The fretting hand takes regular chords, and a foot pedal, when pressed down, triggers an actuator to strum the strings. The actuated guitar has multiple effectors that interact with one another. The instrument consists of seven effectors, six strings, and a pedal-controlled strum actuator. However, Figure 8 only consists of two effectors, one exemplary string, and the foot pedal because the strings are independent and identical in behaviour. To distinguish between the two effectors we color the string blue and the pedal red.

When looking at the string effector the feedback is primarily tactile. When sound is present in the system we have auditory feedback when sliding or bending the string in state 2 or transitioning to state 1, which dampens the string and stops the sound. The pedal effector offers primarily haptic feedback except at Pedal Down, which also gives visual and auditory feedback. When the pedal is en-

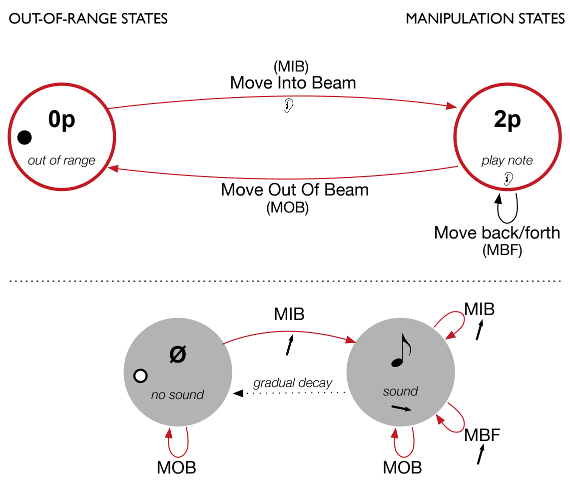

Figure 6: In Soundbeam (NEM) there is no state 1 and therefore no resting position.

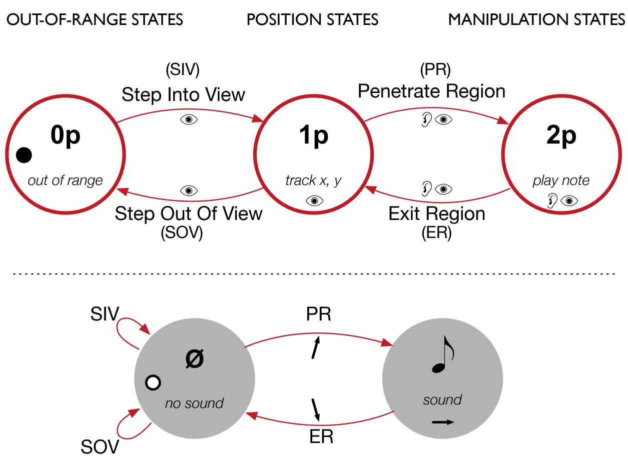

Figure 7: The Movement-To-Music instrument uses computer vision to capture the movement of the user.

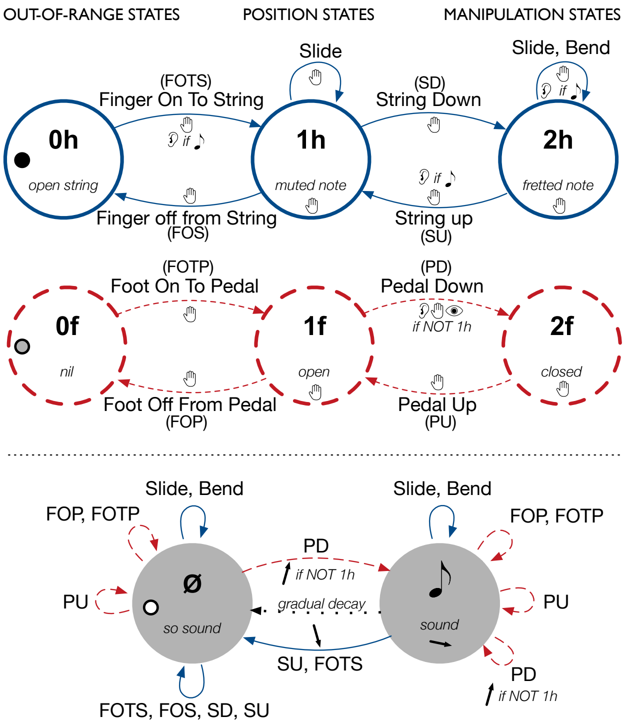

Figure 8: Model of the Actuated Guitar with a single string and a foot pedal.

gaged the actuator drags the pick across the string/s, giving clear visual and auditory feedback. The sound states illustrate an impulse (NEM) instrument in which the attack occurs when pressing the pedal down (PD). Gestures such as sliding or bending can manipulate the sound but the exact mappings of gestures to sound modification are outside the scope of our interaction model. The sound can either decay fully over time, be actively stopped by String Up, or be renewed when the string is strummed again (PD).

4.4 TouchTone

TouchTone (TT) lets children with cerebral palsy engage in musical composition [1]. The instrument consists of 10 pressure sensitive pads, in two rows of five, with associated LED indicators for the unaffected hand and a momentary switch for the affected hand. The pads allow for playing a note while the switch modulates the pitch frequency by one octave up when pressed. Figure 9 shows the model of a multi-effector interface with a single pad and a switch. What is noticeable right away is the shared state **2ps** - the combined solid and a dashed circle. A shared state shows that when both effectors are in state 2 they manipulate the same note. This results in a note playback raised by an octave. This shared state gives some more explicit connections between the states instead of using conditions like *if* and *if NOT*. The pad effector has haptic and auditory feedback. The switch effector is purely haptic unless we move into the shared state when sound is present. The sound states illustrate the creation of sound by putting pressure on the pad (PP). The instrument has continuous (CEM) non-decaying sound as long as the pad is pressed, requiring a release of the pad (RP) to stop the sound.

5. COMPARISON & DISCUSSION

A visual comparison shows a big difference between the modeled instruments from simple (SoundBeam) to more

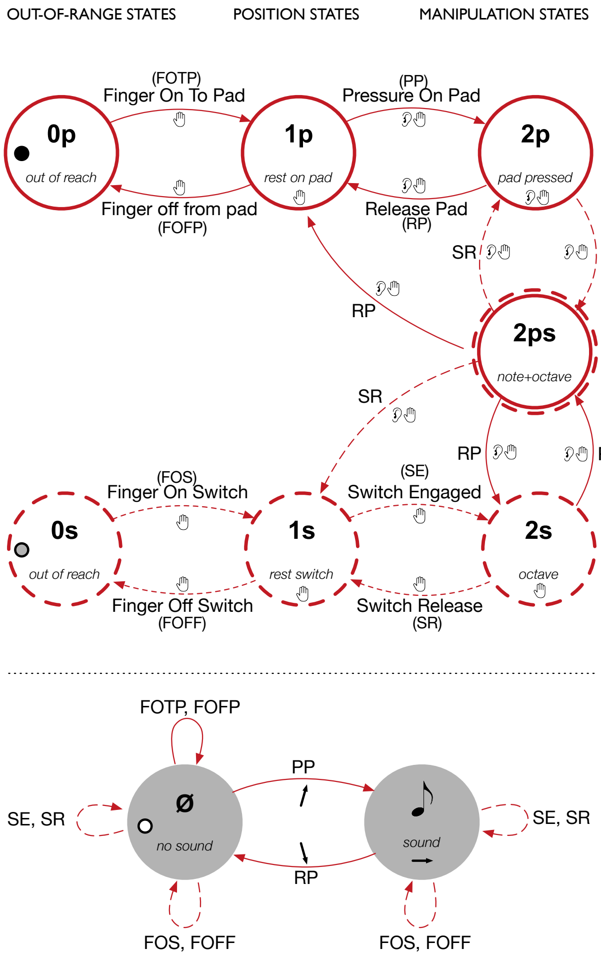

Figure 9: Touch Tone has a shared state shown as the two states overlap.

complex (Actuated Guitar). A visual comparison is much faster than reading through and comprehending large amounts of text. It is quick and easy to compare the different models and it unveils differences and similarities.

But all models describe all meaningful interactions, presence of sound, and feedback. The actuated guitar is the only system that is a hybrid of assistive and existing instruments evident from the different colors of the model effectors. Simple asymmetries, in TT's sound model in Figure 9, allow for verification that the actions FOTP and FOFP can only occur when there is no sound in the system. This is due to the fact that the user releases the pad (RP) prior to taking their finger off the pad (FOFP) and RP stops the sound. A designer pondering whether to add some auditory feedback in response to Finger onto Pad (FOTP) can thereby verify that this does not conflict with sound from the system. The sound model allows for checking completeness as all transitions should either be represented and emanate from each state or not be possible. Take Figure 8 as an example. The only transitions from the interaction model missing from the sound state are FOS and SD. Both transitions originate from state 1h (muted note), which, by definition, does not allow for sound. The absence of self-loops in the sound state for AR and MTM illustrate that these MUIs do not allow for further manipulation or pre-empting of sounds through gestures unlike SB, TT, and AG. Manipulation using gestures would be particularly interesting for assistive musical interfaces that need to be tailored to perceptual, cognitive, and motor abilities of their users. Preconception would also be interesting to add to the model as remote sensing technologies completely remove the tactile feedback channel; however, proprioception are indirectly shown in our model by the absence of tactile feedback (the hand symbol).

Designers can harness the model as a design tool to help identify requirements, incorporate desired gestures, and manipulate expressive parameters (pitch, timbre, and loudness) of the musical interface. It can be used for documenting, discussing, and publishing new musical interfaces and gives a birds eye view of the current design and facilitate a much more efficient and less error prone design process as discussed. Further benefits are easier checks for completion, e.g. by checking that in each state we have all eligible actions represented in transitions. We can use visual asymmetries to verify and potentially re-think design choices. For the merits of state diagrams for modeling we refer the reader to e.g. Thimbleby's work [20]. Using conditional logic helps control the flow of the model, but it could be further extended to include a weighting factor to state transitions to quantify cognitive and motor costs of actions as suggested by Hinckley et al. [8]. Researchers can more easily establish an overview and compare a range of instruments allowing for a faithful reproduction of research results. In further research we would like to explore if the model might require further extensions to incorporate more gestures to capture the expressiveness of musical devices from velocity, vibration, tempo, etc. In the current state of the model we cannot tell if certain interactions in *sound state* actually manipulate sound and, if so, how. Such information could be included by further enhancing the self-loops of the sound states with additional modifiers. Including these could help when comparing the different devices and evaluating whether the chosen interaction comes with an expressiveness cost.

6. CONCLUSION

We have described a novel way of modeling musical interfaces and provided a visual vocabulary and method for systematically describing and analyzing existing musical interfaces in terms of their actions and feedback, as well as how they manipulate sound. The modelling framework provides a quick overview that allows for easier collaboration when designing or analyzing musical interfaces. The model has been shown to work on CEM and NEM instruments in general and, more specifically, on a wide range of different assistive musical instruments. It allows for a complete description of the individual instrument's interaction possibilities with respect to sound.

7. REFERENCES

[1] S. Bhat. TouchTone: An electronic musical instrument for children with hemiplegic cerebral palsy. In *Proc of TEI'10*, pages 305–306. ACM, 2010.

[2] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley. Towards a dimension space for musical devices. In *Proc. NIME'05*, pages 192–195. National University of Singapore, 2005.

[3] W. Buxton. A three-state model of graphical input. In *Proc of INTERACT'90*, volume 90, pages 449–456, 1990.

[4] J. Françoise, B. Caramiaux, and F. Bevilacqua. A hierarchical approach for the design of gesture-to-sound mappings. In *9th Sound and Music Computing Conference*, pages 233–240, 2012.

[5] M. Goldstein. Gestural coherence and musical interaction design. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1076–1079. IEEE, 1998.

[6] M. Grierson and C. Kiefer. Noisebear: a wireless malleable multiparametric controller for use in assistive technology contexts. In *CHI'13 EA*, pages 2923–2926. ACM, 2013.

[7] I. Hattwick and M. M. Wanderley. A dimension space for evaluating collaborative musical performance systems. In *Proc. NIME'12*, volume 12, pages 21–23, 2012.

[8] K. Hinckley, M. Czerwinski, and M. Sinclair. Interaction and modeling techniques for desktop two-handed input. In *Proc of UIST'98*, pages 49–58. ACM, 1998.

[9] J. V. Larsen, D. Overholt, and T. B. Moeslund. The Actuated Guitar: Implementation and User Test on Children with Hemiplegia. In *Proc. NIME'14*, pages 60 – 65, 2014.

[10] D. J. Levitin, S. McAdams, and R. L. Adams. Control parameters for musical instruments: A foundation for new mappings of gesture to sound. *Organised Sound*, 7(02), Aug. 2002.

[11] B. McCloskey, B. Bridges, and F. Lyons. Accessibility and dimensionality: enhanced real time creative independence for digital musicians with quadriplegic cerebral palsy. In *Proc. NIME'15*, pages 24–27, 2015.

[12] F. Morreale, A. De Angeli, and S. O'Modhrain. Musical interface design: An experience-oriented framework. In *Proc. NIME'14*, pages 467–472, 2014.

[13] Z. Obrenovic, J. Abascal, and D. Starcevic. Universal accessibility as a multimodal design issue. *Communications of the ACM*, 50(5):83–88, 2007.

[14] D. Overholt. The musical interface technology design space. *Organised Sound*, 14(2):217, 2009.

[15] J. L. Peterson. Petri nets. *ACM Computing Surveys (CSUR)*, 9(3):223–252, 1977.

[16] C. Roads. *The Computer Music Tutorial*. MIT press, 1996.

[17] T. Swingler. *" That Was Me!": Applications of the Soundbeam MIDI Controller as a Key to Creative Communication, Learning, Independence and Joy.* ERIC, 1998.

[18] T. Swingler. The invisible keyboard in the air: An overview of the educational, therapeutic and creative applications of the EMS Soundbeam. In *2nd European Conference for Disability, Virtual Reality & Associated Technology*, 1998.

[19] C. Tam, H. Schwellnus, C. Eaton, Y. Hamdani, A. Lamont, and T. Chau. Movement-to-music computer technology: A developmental play experience for children with severe physical disabilities. *Occupational Therapy International*, 14(2):99–112, June 2007.

[20] H. Thimbleby. *Press On*. MIT Press, 2007.

[21] R. Vertegaal, T. Ungvary, and M. Kieslinger. Towards a musician's cockpit: Transducers, feedback and musical function. In *Proc. International Computer Music Conference*, pages 308–311. International Computer Music Association, 1996.

[22] W. M. Zuberek. Timed Petri nets definitions, properties, and applications. *Microelectronics Reliability*, 31(4):627–644, 1991.