**Real-Time Control Improvisation for the SmartJukebox**

Ilge Akkaya, Alexandre Donzé, Sanjit Seshia, David Wessel, Edward A. Lee
University of California, Berkeley

**Background**

**Introduction**
Real-time machine improvisation is a task that requires satisfaction of real-time latency constraints and music specifications. We present an interactive demo featuring real-time improvisation based on user preferences.

**Goals**
- Low-latency, seamless aggregation of human and computer generated music sequences
- Real-time score visualization of generated improvisations
- Interactive music specification via face recognition and RF beacon message inputs
- Tune specification
- "Creativity" level
- Improvisational solo trading between human performer and Ptolemy-Max improvisation
- Integration of AvB protocol for time-synchronized audio streaming

**Learning and Improvisation**

**Factor Oracle Based Learning**
- A factor oracle is a data structure that represents at least all the suffixes of a given input sequence. We use factor oracles to build online automata that produce computer-generated melodies upon execution.
- An input solo is partitioned into "licks": short phrases of note sequences terminated by rests or long notes. Time stamped rhythm and pitch sequences of a training solo are used in creation of separate factor oracles.

**Learning/Improvisation Timeline**

- User preferences received
- Solo playback starts
- Begin OSC send
- End OSC receive
- Solo playback ends
- Imrovisation playback starts
- Factor oracles generated
- Begin OSC send
- Modal model execution
- End OSC receive
- Oracle time

**Workflow**

**Real-Time Learning of Probabilistic Automata in Ptolemy II**

- Ptolemy II is a framework for modeling, simulation, and design of concurrent, real-time embedded systems.
- A DE model in Ptolemy is used to
  - Receive time-stamped OSC messages and produce DE events for note and chord specifications of the training sequence.
  - Utilize a ChordFollower to accept produced note duration values from the DurationOracle and provide current chord information to the PitchOracle for fulfillment of the music specifications in the improvised melody.
  - Update MobileModel actors by providing the generated factor oracle MoML and generate modal models during model execution whenever new training data is received.

**Pitch and Rhythm Specifications**

- The pitch oracle is supervised to impose music specifications on the improvised melody.
- The top-level state machine for pitch generation is supervised via a parameter called startLick that indicates when a new lick may begin.
- validatePitch(pitch, chord) is a function that evaluates to true if the proposed pitch is "acceptable" with the current chord value. The harmony decision is taken upon considering the chord tones and a scale specification.

**References**