self-awareness in hypermusic

Arjun Chandra
Arve Voldsund
Kristian Nymoen
Jim Tørresen
Kyrre Glette

EPICS
Engineering Proprioception in Computing Systems
outline

hypermusic
self-awareness
hypermusic node
solojam
future work
hypermusic

Interacting with flexible musical compositions
Somewhere along an axis between an instrument and a passive listener, there is room for active music, i.e. music that is flexible enough to let a listener change musical parameters through participation.

**Active music**

**Hypermusic:** a particular kind of Active music

*Hypermusic is active music where listeners, i.e. users, interacts with flexible musical compositions*

**Keywords:**
Human in the loop
Context dependent
Flexible compositions
Personal music experience
Composer defined boundaries
Interaction with other users and music via device node e.g. media device
Both software and hardware can be nodes as long as they contribute to the active music context
Flexible Score

Interaction with structures

Interpret a structure

Detailed Score
self-awareness
as defined by the EPiCS project
A **self-aware** node may have:

- Private self-awareness
  - Information about internal state

- Public self-awareness
  - Knowledge about the environment to determine how it is perceived by other parts of the system
A node exhibits **self-expression** if it:

- Is able to assert its behaviour upon itself or other nodes
- And this behaviour is based upon the node's state, context, goals, values, objectives and constraints

Furthermore, a node is called **proprioceptive** if it exhibits both self-awareness and self-expression
hypermusic nodes
• A hypermusic node is a combination of a human user and the device or instrument (including hardware and software) enabling participation in a hypermusic score.

• One could also replace the human with an AI.

• This change could be dynamic, i.e. a human could start a node and then move on to another node, while the first continues autonomously from where the human left off.

• The human is not required to have a high level of musical proficiency, and the hypermusic node would ideally adapt to the skills of the human.
Proprioception feedback loop

Self-expression

Self-awareness

Human Node input

+ Sync +

Device Node input

feedback loop

touch

motion

face

pulse

neighbour

time of day

proximity

gait

orientation

temperature

accel. [x,y,z]

RFID

held?

direction

Music Engine Node

Node input

sym

face

accel. [x,y,z]

motion

orientation

temperature

Proprioception

feedback loop

Internal Selfexpr

Monitor / Controller

External Selfexpr

feedback loop
Possible hypermusic scenarios / topologies:

- Human only nodes
- AI only nodes
- Mixed human / AI nodes
- Small groups (trio, quartet, …)
- Medium groups (orchestra, smaller audience at a concert or exhibition)
- Large groups (festival or rave party participants)
- Local sound generation
- Centralised sound generation
- Run-time changes between the above configurations
solojam

A band with (artificial) solo artists who co-ordinate their respective solos via auctions
Main idea: Band with solo artists where:

• The artists send bids to the current solo player/leader, in order to become the next leader and play their respective solo.

• The current leader plays its respective solo until it can gain from handing over the lead to someone else.

• Imagine handing over the playing of solos from one artist to another over time.

• Related to a *smart camera* application in EPiCS
Node
• leader

Node

Node

10001000

00011100

10101010

Node

Node
Auction

- Auction broadcasted by leader at every time step
- Auction/Bidding scheme: second price auction
- Amounts paid used for book-keeping ("social welfare")
Utility function rationale

• A player should “understand” what the leader is doing musically, and propose a variation of this
• Musically neither too close nor too far away

• The current leader should not keep on playing the same solo/pattern for a very long time (fatigue, boredom, …)
Utility function

\[ u = \frac{c}{(1 + a \cdot D_l) \cdot (1 + b \cdot T_l)} \]

- \( D_l \) = hamming distance to leader’s solo (zero as a leader)
- \( T_l \) = duration of play as leader (zero as bidder)
- \( a \) = weighting for \( D_l = 1.0 \)
- \( b \) = weighting for \( T_l = 0.1 \)
- \( c \) = normalization constant = 2
Special cases:

- If a bidder bids to play a tune with $D_l = 0$, its utility is assigned to be negative (i.e. it is penalized for wanting to play exactly the same as the leader).

- If a leader has just handed over the playing of the solo to another node, it is not allowed to enter the immediate future auction with the same tune as before. This is presently done by assigning it a negative utility.

- These cases should be incorporated in a more general way in the future
Solo encoding and search:

- Bit string where each bit represents a fixed position in the beat pattern
- Bits are mapped to different drums based on position and player
- Searching for a new solo by mutations or random search
  - current solo is replaced if the new solo increases the utility in the bid
Relation to self-awareness definitions

- **Private self-awareness**: information about internal state in terms of knowing what the node wants to play and how long it has been played.

- **Public self-awareness**: capable of evaluating their internal state with respect to the environment by way of being able to assign a utility to their respective solos, which takes into account the leader’s solo (public knowledge) and the self-proposed solo (private knowledge).
Relation to self-awareness definitions

- **Self-expression** (proposing solos, playing sound) based upon
  - State and context (internal solo, leader/not)
  - Goals, values, constraints (musical variation, tempo, instruments)

- Network topology currently “all-to-all” (may change in future)

- Node interaction mechanism can be described as **self-interested** (“Can we make a mutually beneficial exchange?”)
Discussion

• Current utility function is such that it encourages the next leader to play *something like* the current leader, thus coming up with some kind of synchronisation in a decentralised fashion or at least a smooth transition from one solo to the next.

• Social welfare? Calculated as the sum across the instantaneous utilities of each player but not sure what it could mean for the band right now.
future work
solojam and hypermusic in general
Future work

- Ant colony-inspired multicasting approach
  - Build pheromone trails/graph based on handover dynamics
  - Use this for communicating auctions
  - Could be useful with higher number of nodes

- Utility function with more interesting dynamics

- Improved variation/similarity measures
  - Base on musical knowledge
  - Rule-based or data-based?
Future work

• Custom hardware controller

• Introduce user interaction
  • Rhythm input (motion capture or other sensing)
  • User proficiency measurement from machine learning
  • Adapt control level to user proficiency

• Extend to several layers of musical composition
  • Drums, bass, melody, …
  • Several auctions, nodes can bid for participation in different layers
Future work

• Increase the number of nodes, experiments with humans as well as AI nodes

• Apply other algorithms and investigate other kinds of dynamics

• Test on benchmarks

• Bring information back from the music application to the theoretical foundation of the EPiCS project and the other applications
Testing: selected requirements

- Response time (UI and musical)
  - Priority based processing in node
- User-friendliness (intuitive and responsive)
- Priority-based communication
- Sensor management / tradeoffs (monitor/controller)
thank you!

questions, comments and suggestions are welcome