

博論構造メモ

Chapters

- 1.Introduction
- 2.Computer as a Liberal Machine
- 3.Infrastructure of Music in the age of Ubiquitous Computing
- 4.(Re-)Defining a design of Programming language for Sound and Music
- 5.Design of mimium
- 6.Discussion: Toward a Civil Engineering of Music
- 7.Conclusion

Structure

1.全体の概要

2.↓の説明に必要な背景知識

2.1.自由な装置としてのコンピュータ

2.2.音楽の情報インフラストラクチャ

3.音楽プログラミング言語の概念化と指針

3.1.音楽プログラミング言語（と、それを作る行為）とはなんなのかを概念化しよう

3.2.その上で、今どんな言語を作るべきか考えよう

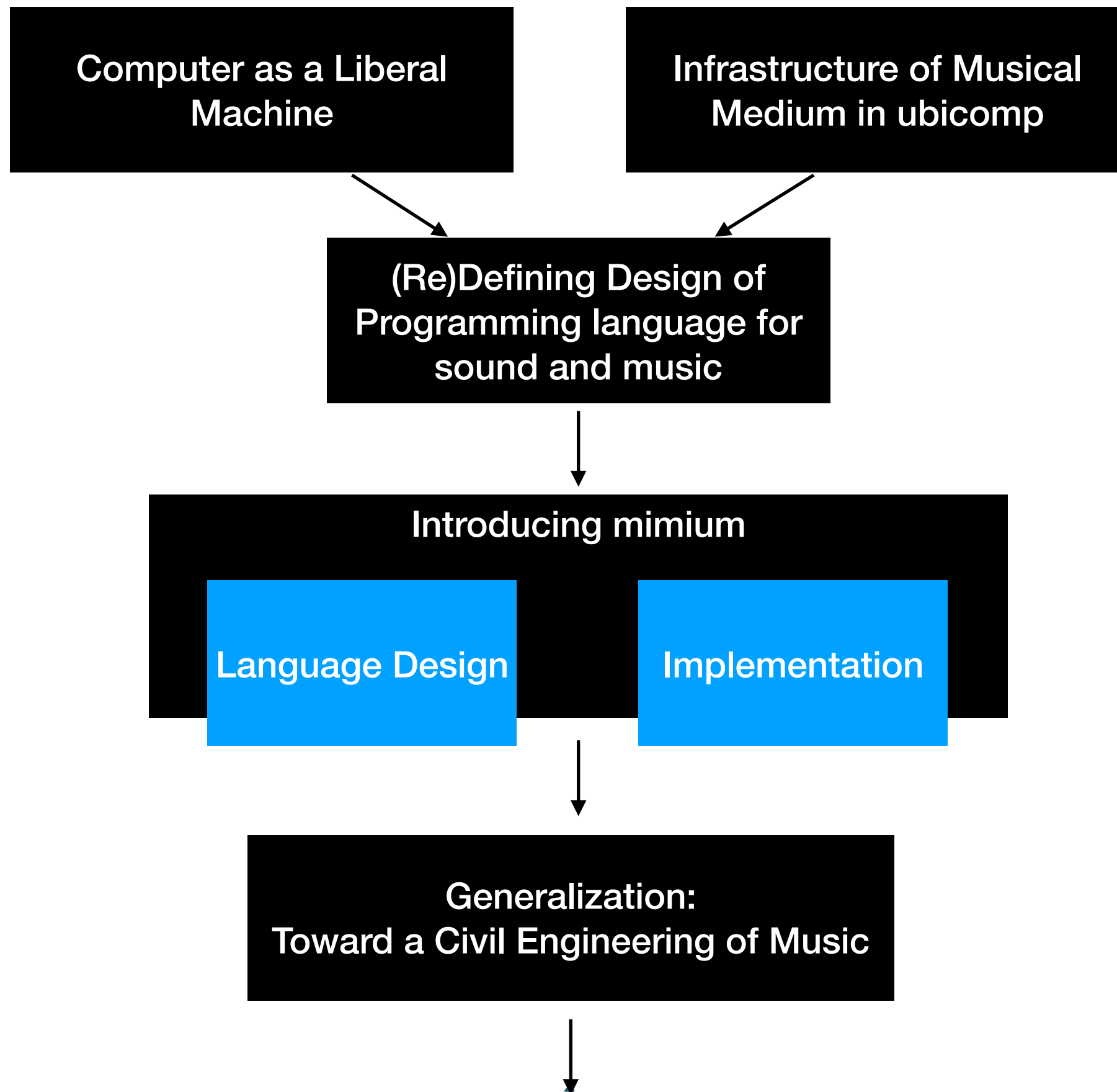
3.2.1.音楽xコンピュータのブラックボックスを無くそう

3.2.2.音楽プログラミングにおける「新しさ」をやめよう

3.2.3.音楽メディウム/インフラをプログラミング言語にしよう

4.言語の設計、実装

5.さらなる一般化：音楽土木工学をはじめよう



Computer as a Liberal Machine

- シャノンの通信モデル：あらゆる情報は信号＋ノイズである
 - サイバネティクス：モデルとシミュレーションによってあらゆる事象を説明できるようになる
- アランケイ “Programming own your computer”
 - Dynabookというユニバーサル・メディア・シミュレーター
 - メタメディアとしてのコンピュータの思想には、自分自身の機能を自分で書き換えられるフレキシブルなプログラミング言語の存在がセットとなっていた。
- マークワイザー “The World is not a Desktop”
 - コンピューターを可変性のある一つの箱ではなく、少ない機能でシンプルな道具として沢山使う、Ubicomp
 - ドナルド・ノーマンや渡邊恵太に繋がる”Invisible is better”の思想、カーム・テクノロジー
- 現在はどうか？ InvisibilityとInaccessibility批判
 - Emmerson “Reading Writing Interfaces”など、しかし、どうやらプロダクトデザインとかだけの問題でもないっぽい→インフラストラクチャの問題へ

Infrastructure of Music in the age of Ubiquitous Computing

- Sterne Audible Past
 - 原音忠実の思想
- Sound on Computer
 - Manovich Software takes command
 - マーカス・ポップ
 - ブライアン・イーノ
- Sterne MP3(Format Studies)
 - 閉じた/オープンな標準規格の成立過程

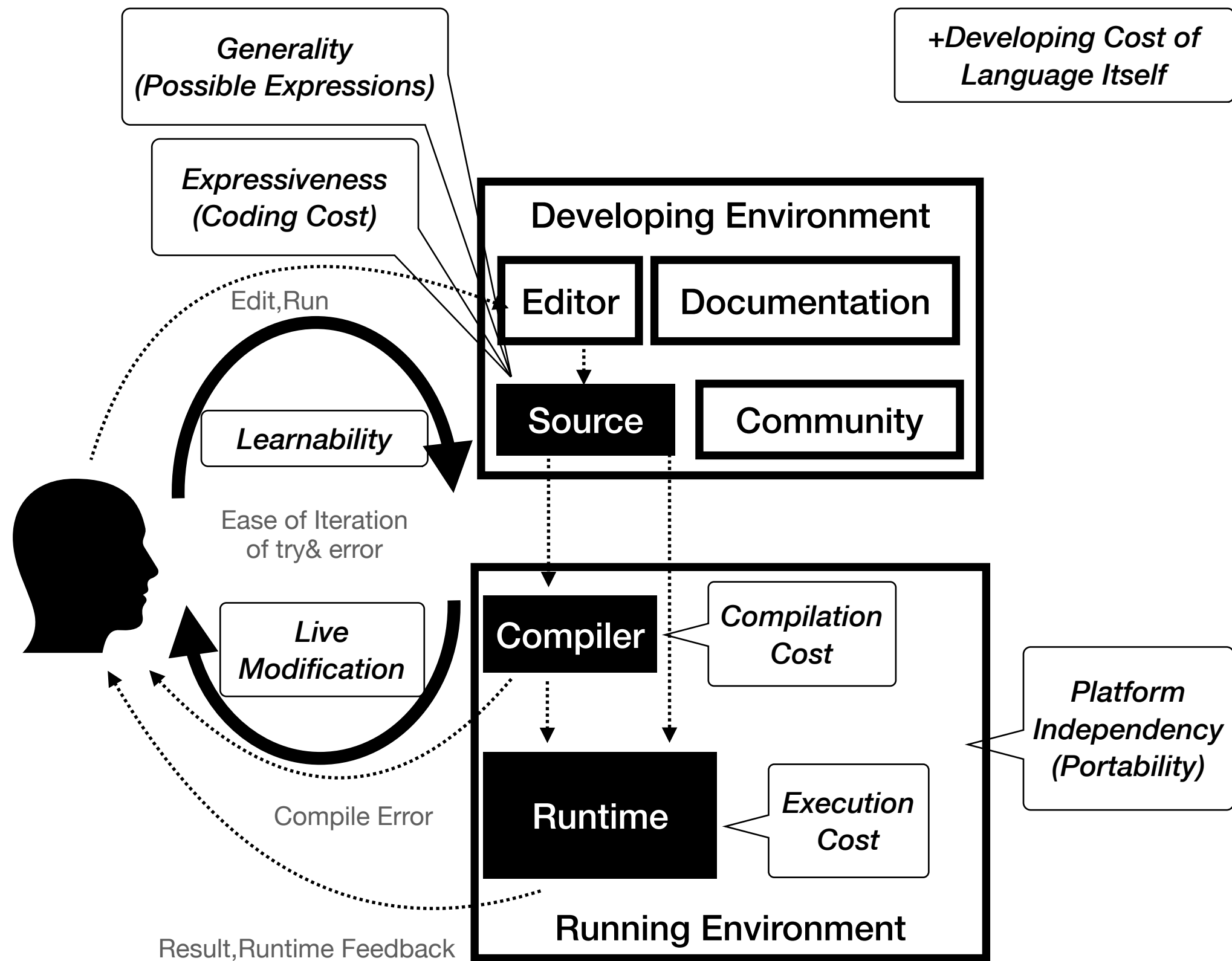
Infrastructure of Music in the age of Ubiquitous Computing

- Infrastructure Studies
 - Star & Bowker “Steps Toward an Ecology of Infrastructure”
 - Infrastructure Inversion (Becker “Art World”)
 - Format Matters(Reader)
- 2021年現在の音楽情報インフラ
 - 作る側：DAW+α / MIDI2.0
 - 配信：ほとんどストリーミング+たまにbandcamp
 - 再生：オブジェクトベース始まる（原音忠実の終わり）

Programming language for sound and music

1.音楽プログラミング言語（と、それを作る行為）とはなんなのかを言語化

- 簡単にいうと、音楽制作ソフトウェアと、プログラミング言語双方の歴史が絡み合っているもの
 - なので、ざっくり「どういう構造で作られるものか」と「どういう経緯で作られてきたのか」の説明が必要
- その上で以下を一般化した形で言語化する
 - 音楽プログラミング言語を「使う」とはどのような行為なのか
 - 音楽プログラミング言語を「作る」とはどのような行為なのか
 - どんなものが「良い」音楽プログラミング言語と呼べるのか



Programming language for sound and music

2.1音楽プログラミングにおける「新しさ」をやめよう

- 言語自体の改善で新しい表現が開拓されることはある（microsoundとか）
- しかし音楽プログラミング言語を作る上で、それによって生まれ得る新しい表現を開拓することだけが有意義なわけではない。
 - むしろ、状況はすでに結構クソだ。まず直すことを考える
- STSにおけるTechnology-in-use(イノベーションについてばかり考察することをやめる)、Infrastructure Studiesの”退屈な研究をはじめようと思う”（誰かがやらないと困るが、誰も率先してやりたくはない）
- 新しさを追求すること自体がイデオロジカルなこともある（グロイス：“On the Newness”）

Programming language for sound and music

2.2音楽xコンピュータのブラックボックスを無くそう

- 音楽を使ったり作ったりコンピュータにつきまとうブラックボックス
 - 音声信号処理の理論の高度化と複雑化、高速化
 - ブラックボックスモデルによる言語化不可能性
 - プラットフォームの不可視性
- これらはなぜ発生するのか、細かくみるとどう違うのか、なぜやめたほうが良いのか
 - ユビキタスコンピューティングにつきまとうinvisibleの思想
 - サイバネティクスにおけるブラックボックスの思想

Programming language for sound and music

2.2音楽xコンピュータのブラックボックスを無くそう

- InvisibilityとBlackboxingの悪魔合体：コンピュータの理論的下支えでもあるサイバネティクスの、Blackboxing(中身を知らずとも入出力だけ知っていれば)を言い訳にして、社会的に非対称な構造を作ってきていないだろうか？
- 翻訳不可能性という意味での、サイバネティクス本来のブラックボックス性は今後も消すことはできない（徳井：作るためのAIなど学習モデル解釈問題）
- でも、開けられるもんは開けところ（UGen、アーキテクチャなどなど）
- とはいえコンパイラが本質に持っている暗号化的な性質が無意識的に後押ししてしまってる側面もあるかもしれない（なので、JITコンパイルは最適化以外にも意義があると言えるのでは）

Programming language for sound and music

2.2音楽xコンピュータのブラックボックスを無くそう

- DAWのプロジェクトファイル
- Unit Generator（特に注目）
 - C++を頑張って書けば拡張可能だが、をどう捉えるか
 - これはChuckKで明確に”White-box Generator”という語用がある

Programming language for sound and music

2.3音楽プログラミング言語のソースコードをメディアウムにしよう

- （流通する）音楽を再生する上でコンピューターをほぼ絶対に経由する状況。
- オブジェクトベースに代表される現代の音楽のフォーマット/インフラストラクチャ事情。
- Software as a minimal musical medium
 - 言語であることの必要性-メディアアートの輪廻転生における徳井作品
 - すでにある事例:SC140、ゲームミュージック
 - もしくはハードウェア 8bit Mixtape, Te Stem Player

(Re-)Defining a design of Programming language for Sound and Music

- Software as a minimal musical medium
 - 言語であることの必要性-メディアアートの輪廻転生における徳井作品
 - すでにある事例:SC140、ゲームミュージック
 - もしくはハードウェア 8bit Mixtape, Te Stem Player
- MIDI:HCIデザイナーにはわりと高評価だが(ノーマンとか)、インペリアルズムになってないか？ OpenSoundControl, CV/Gateとの比較
- 新たな帝国にならないためには？

設計：FARMとかでやってきたやつ

- Ocamlなどの汎用言語に最低限の音楽用機能だけを追加
 - @オペレータ
 - Self、状態付き関数としてのUGen
- 機能比較
- サンプルコード

Generalization

一般化：音楽土木工学を始めよう

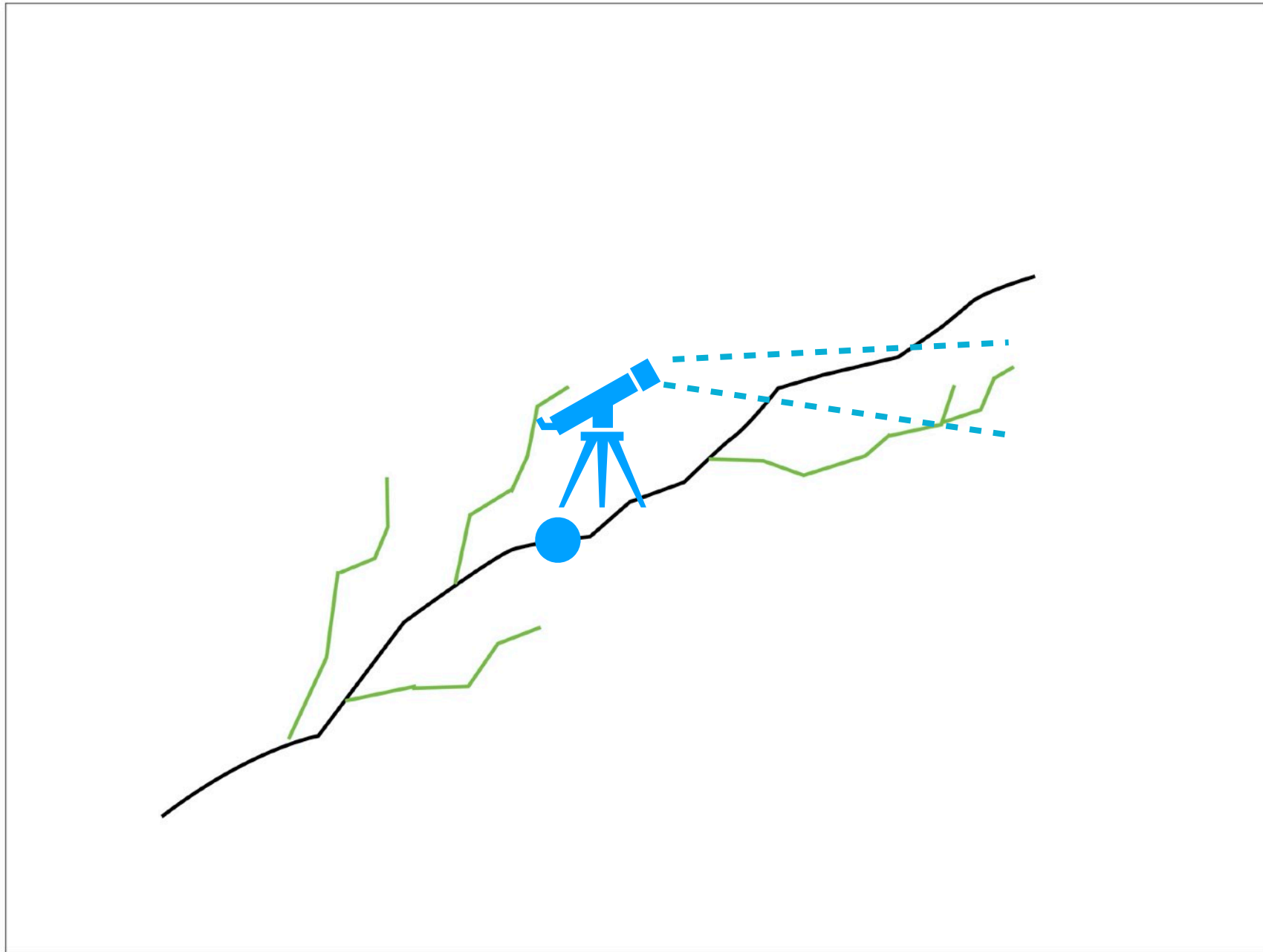
- この論文・研究自体のメタ構造での把握と一般化
 - フーコー的歴史観とメディア考古学、スペキュラティブデザイン
 - Design in progress:いつまでも終わらないデザイン
 - “この状況下で大規模な情報インフラストラクチャを開発しようとするのは、めちゃくちゃ競争率の高いボートレースで、ゴールラインが後ろへどんどん移動し続ける中、自分の乗るボートを組み立てながら、ナビゲーションシステムの設計をしてるようなものだ”(Star&Ruhleder 1999)
 - 少しオートエスノグラフィ的にしてもよいかもしれない、gitのログを辿るとか、言語仕様の変遷をみるとか
- Related approaches
 - 100rabbits,ucnv,???
- 音楽土木工学の確立に向けたFuture works



Improvement? and Sideways Invention- Alternative Technology Narratives
Leah Buechley(2011)

Speculative Design

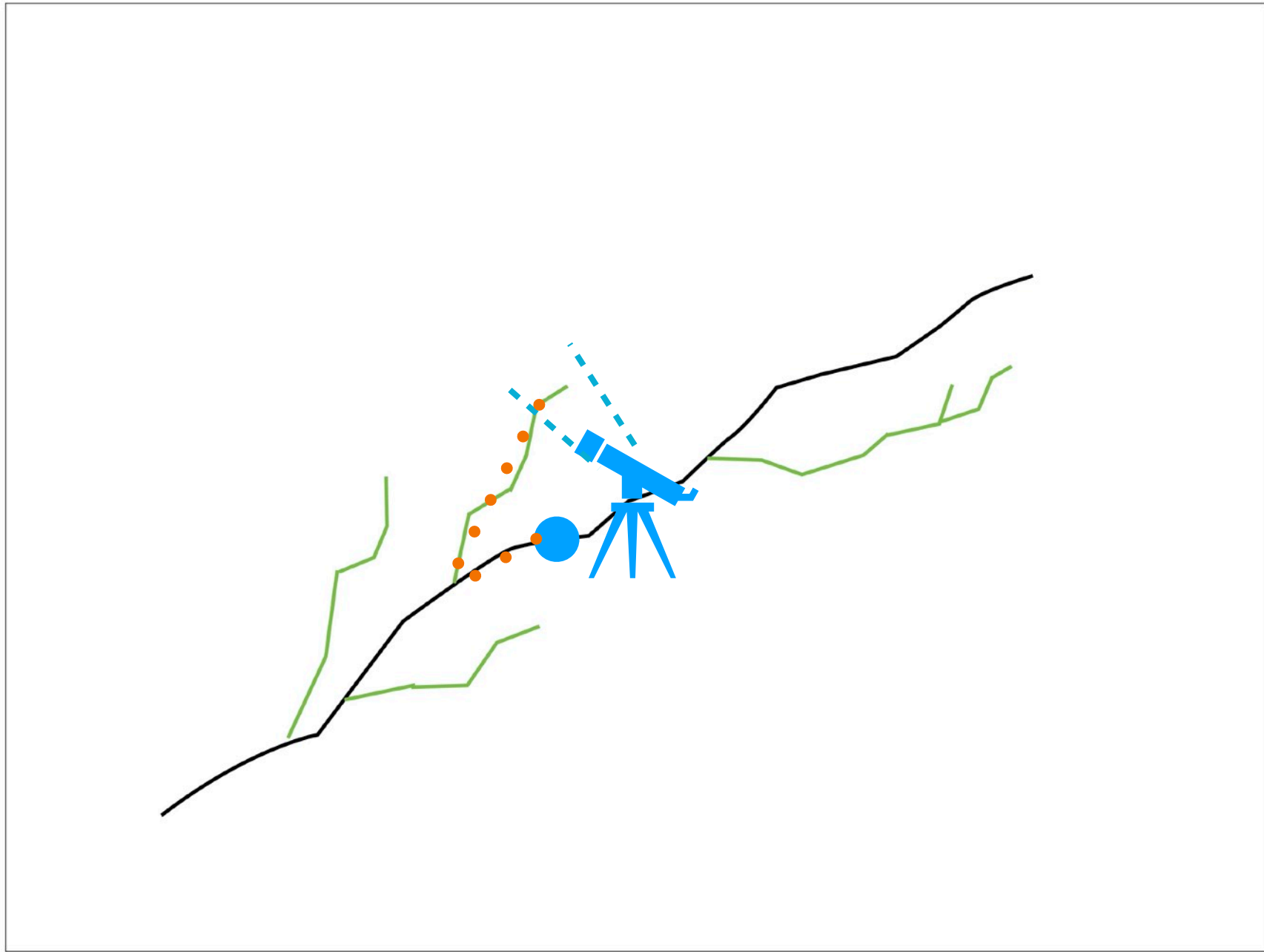
“Possible/Preferable Future”



Improvement? and Sideways Invention- Alternative Technology Narratives
Leah Buechley(2011)

Media Archaeological Practices

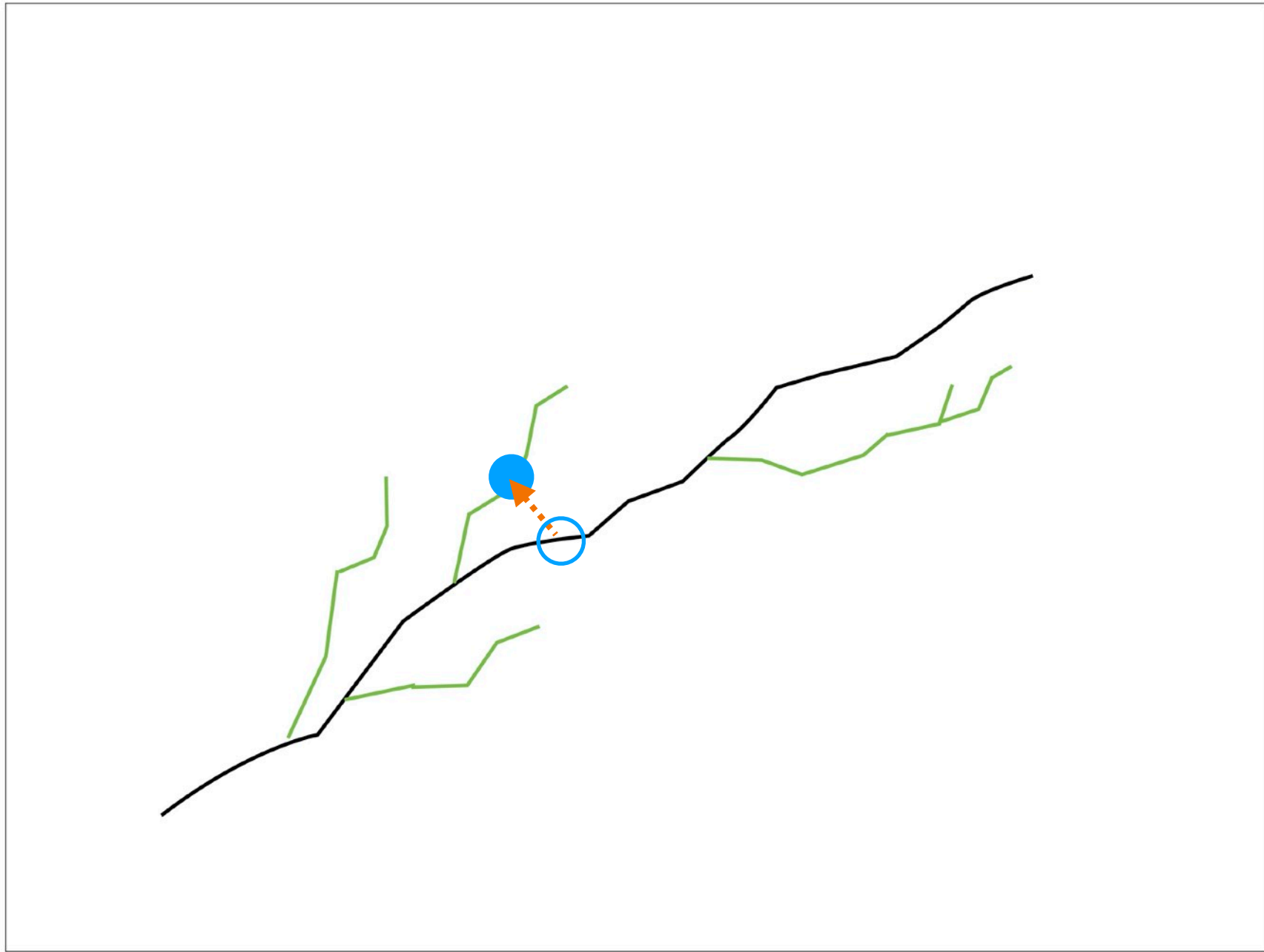
“Alternative Present through the Past”



Improvement? and Sideways Invention- Alternative Technology Narratives
Leah Buechley(2011)

Our Approach

Walk into another present through the wild path



Improvement? and Sideways Invention- Alternative Technology Narratives
Leah Buechley(2011)