

音楽土木工学を設計するー音楽のためのプログラミング言語mimumの開発を通じて

松浦, 知也

<https://hdl.handle.net/2324/4784631>

出版情報 : Kyushu University, 2021, 博士 (芸術工学), 課程博士
バージョン :
権利関係 :

氏 名 : 松浦 知也

論 文 名 : 音楽土木工学を設計する—
音楽のためのプログラミング言語 *mimum* の開発を通じて

区 分 : 甲

論 文 内 容 の 要 旨

本論文では、音楽のためのプログラミング言語 (Programming Language for Music: PLfM) *mimum* の開発の経験を通して、音楽土木工学という、音楽に関わる既存の工学的実践とは異なる学術研究領域を提示する。

第1章では音楽土木工学の概念と論文全体の構造を示す。音楽土木工学とは、プログラミング言語や OS、計算機アーキテクチャのように、音楽のためだけに作られたわけではないが、音楽の技術環境に大きく影響を与えるものを音楽の視点から再考する研究アプローチである。今日、理論的にはあらゆることが可能な電子計算機を介さずに、音楽を作ったり聴くことはもはや難しい。しかしそうした環境にもかかわらず、表現者自らプログラムを書き道具を作るという、パーソナルコンピューティングの当初の思想、メタメディア (Kay and Goldberg 1977) としての電子計算機の利用は主流から程遠く、音楽の技術環境は常に技術インフラストラクチャに依存している。本研究は、そうした環境の変化を促す行動である音楽土木工学という学問を PLfM の設計と開発という実践と思索の中から描き出す。

第2章では、デザインリサーチの研究プログラムの歴史的発展を追いかけることで、本研究が取る、道具を実際に作る中で問題を明らかにするプロセスの学術的方法論としての立ち位置を明らかにする。1960年代以降、特にコンピューターを用いたシステムのデザインは定量的で反証可能な科学のプロセスを取り入れてきた。しかし1980年代以降、解決すべき問題の複雑化や、科学技術社会論や現代美術など隣接する領域の影響を受け、役に立つものを作るのでなく定量的に効果も測定できない、人工物の創出を通じて社会に問いを投げかける目的としたデザインの意義が提示されてきた。本研究はこうしたデザイン研究史の延長線上にある、メディア考古学と批判的デザインの接続 (城 2016) や、批判的奇譚づくり (Rosner 2018) を参照し、作ることを通じたメディア技術の異なる歴史認識の提示という方法論を取る。

第3章では、PLfM を異なる歴史の視点から捉え直すための背景として、表現のための道具としてのコンピューターの歴史を振り返る。現代のコンピューティング環境は、メタメディアの理想が不完全な形で実現されユーザーは元来コンピューターの利用と不可分であったプログラミングから遠ざけられてきた。コンピューターを用いる音楽実践には、そもそもその音楽を想像できなければ作ることもできない原理的限界がある。しかし既存の音楽様式が産業発展と結びつくことにより、標準規格などのインフラストラクチャに埋め込まれた結果、もはや想像できても実現できない状況を作り出す。テクノロジーを積極的に用いる音楽実践として2000年代までは機能していた、グリッチのような技術の中身を知らないままの意図的な誤用は、今日のブラックボックス化された音楽技術文化の中ではもはや機能しなくなっている。

第4章では、より PLfM に焦点を当てその歴史的系譜を整理する。PLfM の形式で長く所与のものとされている Unit Generator (UGen) の考え方は、元々様々な形がありうる抽象化方法の中で暫定的に選ばれた1つでしかなかった。しかし、コンピューターの産業化とリアルタイム性の要求に伴い UGen という抽象化方法をそのままハードウェア化した環境が研究され、それは結果的にソフトウェア中心の今日の PLfM の言語仕様にも根強く結びついた。2000年以後は UGen に限らない多様な抽象化の試みが行われてはきたが、PLfM を抽象化のレイヤー毎に個別の言語を開発/利用する多言語パラダイムという状況を生み、音楽様式に依らない抽象的な計算モデルの追求は未だ不十分である。

第5章では、前章での通時的な整理に対し共時的比較として、PLfMを用いて生成される表現とは別に議論可能な、かつ現状不足している設計や実装に関する理論をまとめる。汎用言語における議論を参照しつつ、PLfMを使用、開発するプロセスをHuman-in-the-Loopモデルとして提示し、言語の中間表現を大きくすれば動的変更に強くなる、小さくすれば表現の汎用性が高まる、動的変更と汎用性を両立すると設計が複雑化し実装のコストが嵩むといったトレードオフの存在を指摘する。

第6章では筆者が実際に設計したPLfM、mimiumの詳細を記述する。mimiumは汎用の関数型言語の設計に、音楽のための仕様を最小限追加する方針を取る。1つは関数の実行のスケジューリングを行う@演算子、もう1つは内部状態を伴う信号処理を純粋な関数と同じ記法で書ける、状態付き関数である。mimiumはこの2種類の記法を持つことで、ブラックボックスとして与えられていた基本的処理の単位をライブラリとして実装可能であることを、既存の言語との比較を交えて示す。

第7章では、mimium開発を通じて浮かび上がってきた音楽のための道具作りの異なるあり方について議論する。mimiumは最低限の時間操作を備えた汎用言語という方針により、メタファーに頼らない音声の記述を可能にはした。しかしブラックボックスの少ない言語を作る方針を突き詰めると、ホスト言語自体の実装とその上でのライブラリ構築という2種類の作業への分岐が起き、結果として、音楽家とプログラマという分業が残ってしまう。それこそが、PLfM開発よりもメタ的な行動理念、例えばプログラミング言語を作る技術者に音楽の視点を提供したり、表現者に自らの表現と技術インフラが無関係でないことを伝える枠組み、すなわち音楽土木工学を要請する。