

# Architecture of Hybrid Language's Systems

d8161105 Mirai Watanabe ・ Yutaka Watanobe

## 概要

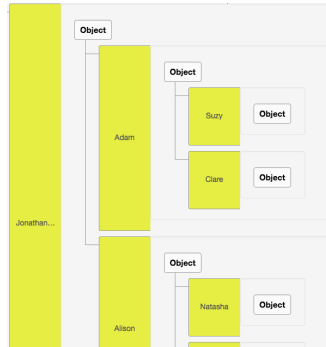


一般的に言語と言うと多くの場合は、文字によって構成された、テキスト言語を想像するかもしれません。しかし、視覚的な表現を使って構成されるビジュアル言語と呼ばれるものも多く存在します。我々は互いの長所を活かすために、テキストとビジュアルが一対一対応するようなハイブリッド言語・環境に注目しています。ハイブリッド言語はリアルタイムかつ相互に変更を反映する必要があるため、それを体現したシステムを構築するのは困難です。我々の研究では、ハイブリッド言語システムの構築をスムーズに行うためのアーキテクチャを提唱しました。それにより、ハイブリッド言語が多く生まれ発展していくことを目的としています。

## Hybrid言語

JSONによるツリー構造のハイブリッド言語

```
{
  "John Smith": {
    "Adam": {
      "Suzy": {},
      "Clare": {}
    },
    "Allison": {
      "Natasha": {}, "Zak": {}
    }
  }
}
```



### テキスト言語の利点

- 既存エディタの機能を使い迅速な編集が出来る。
- データの保存に適している。
- プログラムに直接埋め込める。
- データの解析が容易

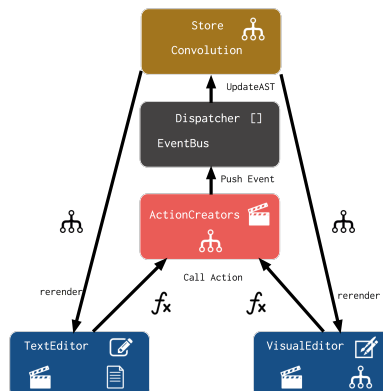
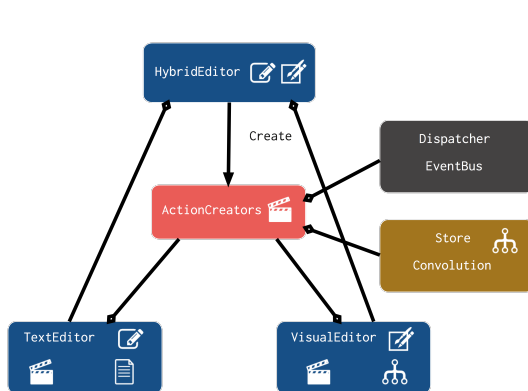
### ビジュアル言語の利点

- 視覚的にわかりやすい。
- 感覚的に編集することが出来る。
- 学習に適している。

## ハイブリッド言語の例

地図・数直線・グラフ・データ構造・UML・物理シミュレーション・APIドキュメント・建築設計図面・実験データ・譜面・ゲームデータ(棋譜等)

## Hybrid言語アーキテクチャ



- Reactフレームワークを採用している。
- Fluxアーキテクチャを採用している。
- VirtualDOMにより差分のみの更新で済むため、パフォーマンスに優れている。
- 各構成要素はコンポーネント化されており、再利用可能です。
- テキスト・ビジュアルどちらも同一の中間データ構造 (Abstract Syntax Tree) から生成することが出来る。
- 操作はActionCreatorsを通して行われる。
- Dispatcherにより操作の順序が守られる。
- Storeに保存されているASTが変更される。
- ASTの変更を通知し、ASTからテキスト・ビジュアル言語が生成される。
- 結果的に、リアルタイム性・言語の整合性が保たれる。

## 結論

テキストとビジュアルの両方の利点を兼ね備えているハイブリッド言語は、アイデア次第で様々な分野の現場で活躍出来るパフォーマンスを備えています。しかし、コンピュータ上でリアルタイム性・パフォーマンス・柔軟な表現力・ユーザビリティを確保しなくてはならないためシステムを構築するのは非常に困難です。そこで、我々の提唱したアーキテクチャを元にハイブリッド言語のためのシステムを構築することで、必要とされる項目を満たす事ができ、また多くの機能を共通化・再利用出来るため、生産性が向上します。生産性が向上することで、多くのハイブリッド言語が作り出され様々な分野の学習・発展に貢献出来ると考えています。また、アーキテクチャを採用したハイブリッド言語フレームワークを提供することで、更なる生産性の向上が今後の課題です。