

卒業論文 2019 年度（令和 01 年度）

Hyper Launcher
ホットキー型アプリケーションランチャーの
利便性を高める研究

慶應義塾大学 環境情報学部

佐藤 佑哉

増井俊之研究会

2020 年 1 月

卒業論文 2019年度（令和01年度）

Hyper Launcher

ホットキー型アプリケーションランチャーの 利便性を高める研究

論文要旨

ホットキー型のアプリケーションランチャーはキーの入力のみで特定のアプリケーションを起動することができるとしても強力なユーティリティソフトウェアであるが、使いこなすのが難しく活用している人は少ない。このタイプのランチャーが広く活用されていないのは、その設定の煩雑さや設定を記憶しなければならないというハードルの高さにある。これらの問題は工夫次第でまだ改善できる余地があると思われる。

本研究では、既存のアプリケーションランチャーの種類や特徴について調査した上で、より強力で扱いやすいホットキー型のランチャーを実現するためのインターフェースを提案し、その発展性について考察する。

キーワード

ランチャー, インターフェース

慶應義塾大学 環境情報学部

佐藤 佑哉

目 次

第 1 章 序論	1
1.1 背景	2
1.1.1 アプリケーションランチャーの種類と比較	2
1.1.2 既存のアプリケーションの問題点	4
1.2 本研究の目的	5
1.3 本論文の構成	5
第 2 章 システムの提案	7
2.1 設計	8
2.2 基本操作	8
2.2.1 アプリケーションの登録と起動	8
2.2.2 アプリケーションの登録変更と解除	8
2.2.3 複数アプリケーションの登録と操作	9
第 3 章 実装	13
3.1 システム構成	14
3.2 クライアントアプリケーション	14
3.2.1 Electron	14
3.2.2 AppleScript	15
3.3 データストア	15
第 4 章 実運用及び評価	16
4.1 筆者の運用	17
4.2 第三者の意見	17
4.2.1 アプリケーションの登録について	17
4.2.2 ホットキーの覚えやすさについて	17
4.2.3 ホットキー自体の有用性について	17
4.2.4 問題点	18
第 5 章 議論	19
5.1 関連研究	20
5.1.1 ExposeHK	20
5.1.2 HotStrokes	20

5.2 展望	20
5.2.1 ネイティブアプリケーション	20
5.2.2 習熟のサポート	20
5.2.3 熟練者向けの機能	20
第 6 章 結論	22
謝辞	24
参考文献	25

図 目 次

1.1	Dock	2
1.2	Dock 内のメニュー	3
1.3	Spotlight	4
1.4	PMenu	5
1.5	Snap	6
2.1	Hyper Launcher	9
2.2	ドラッグアンドドロップによる登録	10
2.3	追加ボタンから登録	10
2.4	ドラッグアンドドロップによる追加	11
2.5	登録の変更	11
2.6	登録の解除	12
3.1	システムの構成	14

第1章　序論

本章では、本研究の背景と目的、及び本論文の構成について述べる。

1.1 背景

アプリケーションランチャーはオペレーティングシステムに標準で搭載されているだけでなく、サードパーティ製としても多くのソフトウェアが開発/公開されている。ホットキー型のランチャー以外にも異なるインターフェースを持ったランチャーが数多く存在しており、それぞれ異なった利点/欠点がある。

1.1.1 アプリケーションランチャーの種類と比較

アプリケーションランチャーには以下のような種類があり、その特徴に合わせて様々な方法でアプリケーションを起動することができる。

(1) パレット型

画面上の一部に固定されたパレット型のエリアにアプリケーションを登録し、マウスによるクリックで起動するタイプのランチャー。macOSにおけるDock（図1.1）がこれにあたる。デスクトップに常駐していることから簡単にアクセスでき、誰でも使いやすいものとなっている。しかしそのエリアは限られており、多くのアプリケーションを登録しようとすると、ボタン数を増やしたりそれぞれを小さく表示したりする必要がある。基本的にその数が増えれば増えるほど操作性が低下するため、数個から数十個の頻用するアプリケーションを登録して使用するのが推奨される。



図 1.1: Dock

(2) メニュー型

上述したDock等にメニューを設け、マウスを使用して登録したアプリケーションを起動できるようにするタイプのランチャー。ショートカットキーと組み合わせて使用されるものもある。複数のアプリケーションを一つにまとめることで場所を節約できるだけでなく、階層化によって自分の使いやすいように整理することもできる。しかし、項目や階層が増えれば目的のアプリケーションに辿り着くまでの操作ステップは増えることになる。

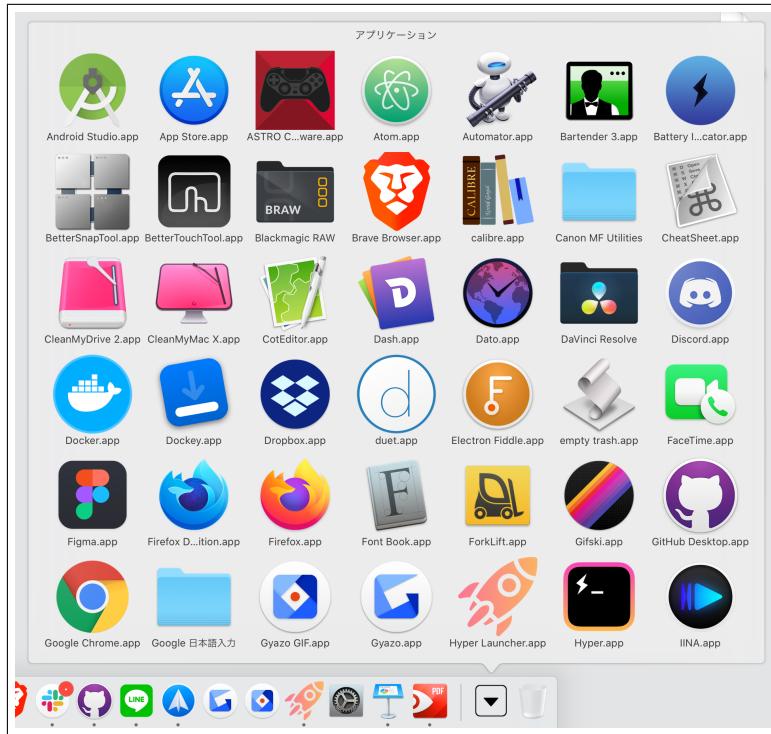


図 1.2: Dock 内のメニュー

(3) 検索型

アプリケーションの名前を入力することで対象のアプリケーションを検索し起動するタイプのランチャー。macOS における Spotlight がこれにあたる。検索メニューは使用するときのみ表示されるため使用していない時は場所をとらず、キーボードのみの操作で完結しているのも利点の一つである。もちろんマウスと組み合わせて使用することもできる。また大抵の場合インクリメンタルサーチが導入されているため、名前を全て覚えていなくても起動することができる。しかし汎用性が高い反面、毎回一定のキーボード入力が必要となってしまうというのが欠点である。

(4) ホットキー型

単一のキーもしくは複数のキーの組み合わせを入力するだけで、登録したアプリケーションを起動できるタイプのランチャー。今回着目しているのがこのである。他のとは違い、ホットキー型は標準で搭載されていないことがほとんどである。したがって自分に適したサードパーティ製ソフトウェアを探してくる必要がある。画面上に情報を表示する必要がなく場所を取らないことに加え、一発で特定のアプリケーションを起動できるため、とても強力なランチャーである。しかし設定が面倒であったり、どのキーにどのアプリケーションを登録したのか覚えておく必要があったりと、初心者には使いにくいタイプだとされているのも事実である。これこそが標準として機能が提供されていない理由の一つだと考えられる。

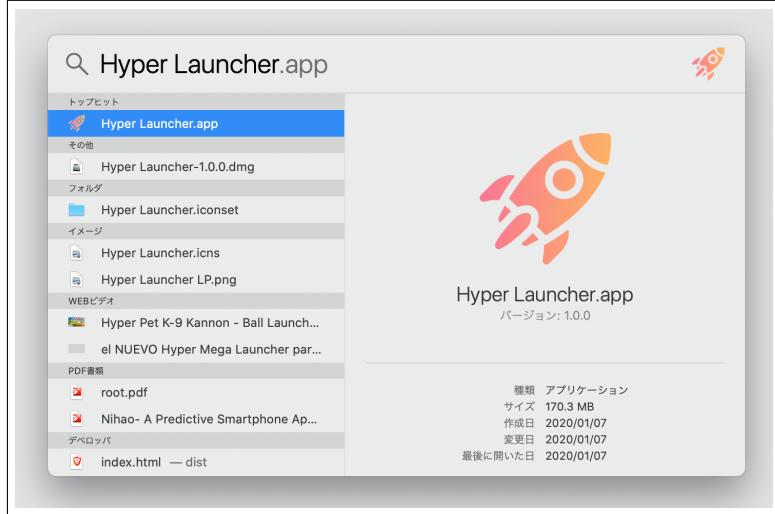


図 1.3: Spotlight

1.1.2 既存のアプリケーションの問題点

先述した通り、ホットキーを利用したアプリケーションランチャーは標準で搭載されておらず、使用者自ら使いやすいものを探す必要がある。しかしその種類は豊富とは言えず、突出した特徴があるわけでもない。例として筆者が日常的に使用してきた2つのアプリケーションランチャーを示す

(1) PMenu

PMenus¹(図1.4)はホットキーとメニューを利用したランチャーで、アプリケーションだけでなく様々なファイルをコマンドで操作することができる。使い勝手はいたって普通で可もなく不可もなしと言ったところである。自分の好きなホットキーを割り当てられるのは便利であるが、キーを登録するためのステップが多くとても煩雑である。コンテキストメニューを利用する事が主な方法とされているため、ホットキーに関しては最低限の機能が実装されているだけと言える。

(2) Snap

Snap²(図1.5)はホットキーを利用したランチャーで、macOSのDockを生かした機能が備わっている。SnapではDockに登録されたアプリケーションに左から順に数字を割り当てていき、その数字と設定したモディファイアキーとの組み合わせでアプリケーションを起動できるようになっている。これにより面倒な設定をすることなくホットキーの恩恵を授かることが可能となる。ただしDockのサイズは有限であり画面上に場所もとるため、多くのアプリケーションを登録するというのはあまり現実的ではない。

¹<https://danadesign.ltd/>

²<https://apps.apple.com/jp/app/snap/id418073146>

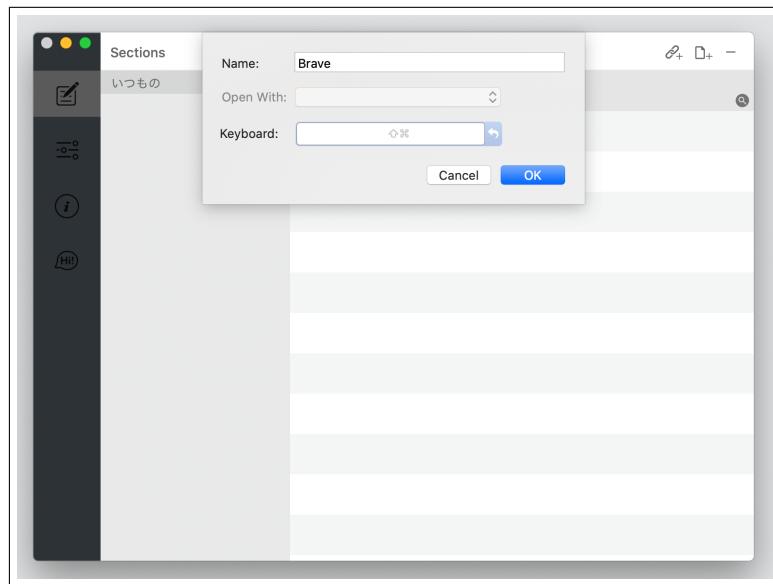


図 1.4: PMenu

1.2 本研究の目的

既存のホットキーを利用したアプリケーションランチャーにおける不便を解消し、より強力でより多くの人に使いやすいシステムを開発することが本研究の目的である。

1.3 本論文の構成

第1章では、本研究における背景と問題意識、目的について述べた。第2章では、第1章で述べた問題意識を踏まえ、新しいホットキー型アプリケーションランチャーシステムを提案する。第3章ではシステムの実装に関して述べ、第4章では関連研究について述べる。第5章ではシステムの考察を行い、第6で本研究を総括する。

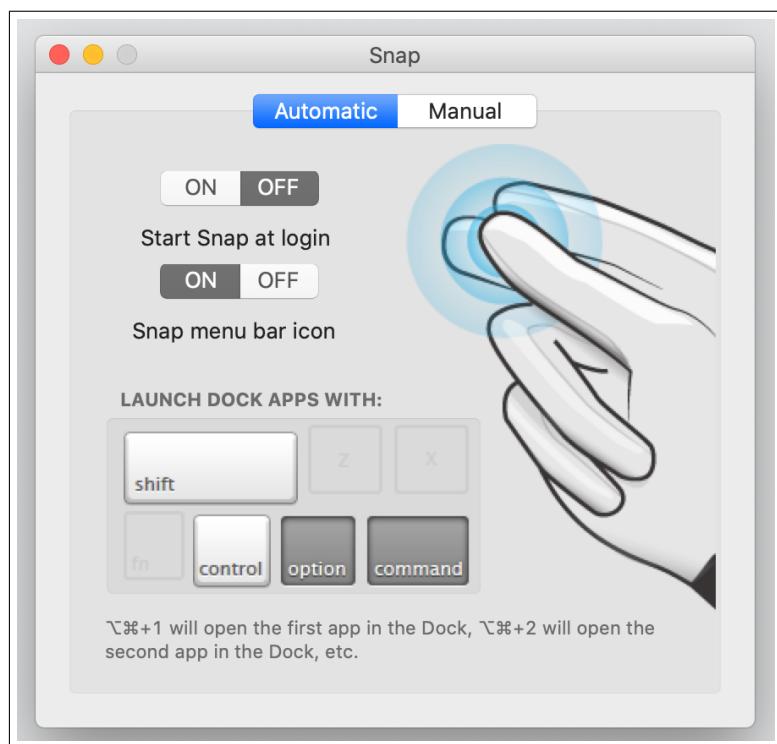


図 1.5: Snap

第2章 システムの提案

本章では、アプリケーションランチャーについての背景を踏まえ、ホットキーを利用した新しいランチャーシステム「Hyper Launcher」を提案する。

2.1 設計

既存のホットキー型アプリケーションランチャーに見られる不便を解消するため、

1. アプリケーションの登録及びその変更が容易にできる
2. 設定が覚えやすい
3. より強力に操作できる

という3つの目標を実現するランチャーシステム「Hyper Launcher」を設計した。具体的には

1. アプリケーションの登録や変更をドラッグアンドドロップによって簡単に行えるようにする
2. 使用するホットキーの組み合わせを9つに制限し、そのキーに対してアプリケーションを登録することで、可能な限り認知不可を下げられるようにする
3. 単一のホットキーの組み合わせに対して複数のアプリケーションを登録できるようにする

というアプリケーションを実装した。

2.2 基本操作

自身の開発環境を考慮し、アプリケーションはmacOS上でデスクトップアプリとして利用できることとする。起動画面を図2.1に示す。画面は9つのセクションに分けられており、予めそれぞれにホットキーが指定されている。

2.2.1 アプリケーションの登録と起動

任意のアプリケーションをそれぞれのセクションにドラッグアンドドロップすることで簡単に登録することができる。また、タイトル横のプラスボタンを押すことでアプリケーション選択画面が出現し、そこからも登録することができる。後は指定されたホットキーを入力するだけでアプリケーションを起動することができる。(図2.4, 2.3)

2.2.2 アプリケーションの登録変更と解除

アプリケーションに紐付けるホットキーを変更したい時は、対象のアプリケーションを目的のセクションへドラッグアンドドロップするだけで操作が完了する。またアプリケーションにマウスカーソルを乗せた際に右側に表れる削除ボタンを押すことで、登録を解除することができる。(図2.5, 2.6)

2.2.3 複数アプリケーションの登録と操作

既にアプリケーションが登録されているセクションに、同じように新しいアプリケーションを追加するだけで複数のアプリケーションを登録することができる。アプリケーションは表示されている順番によって優先順位が定められており、キーを入力する度にその優先順位に基づき順番に起動するようになっている。その優先順についてもドラッグアンドドロップするだけで変更することができる。

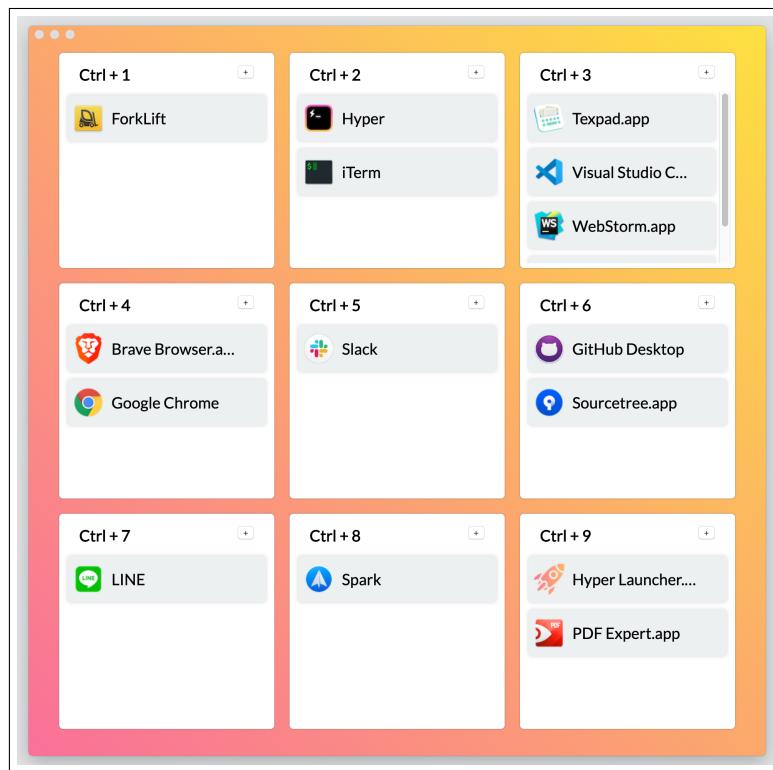


図 2.1: Hyper Launcher

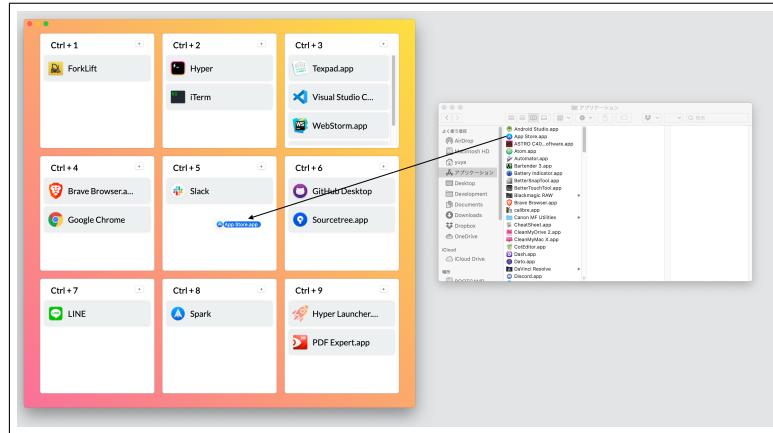


図 2.2: ドラッグアンドドロップによる登録

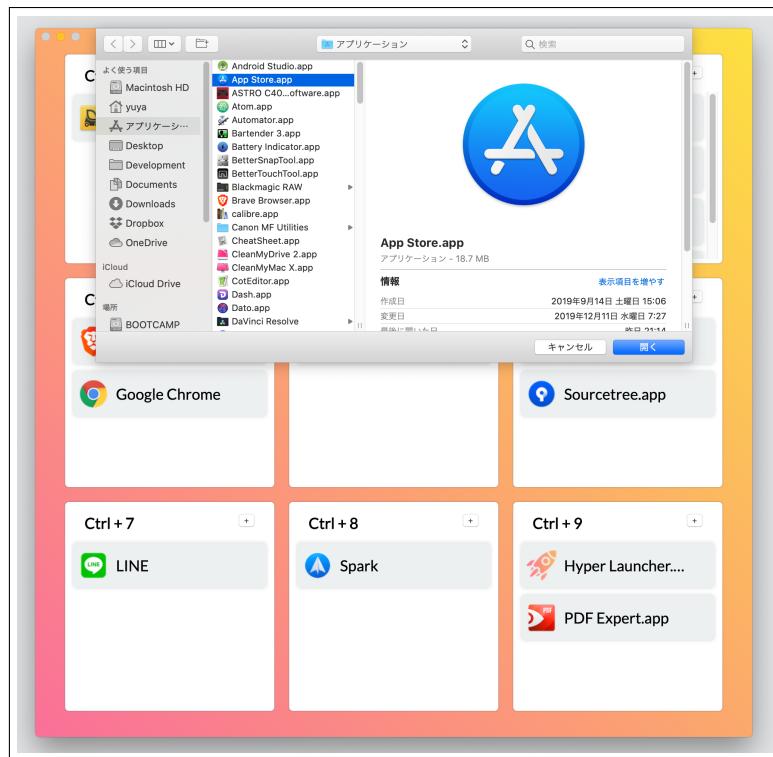


図 2.3: 追加ボタンから登録

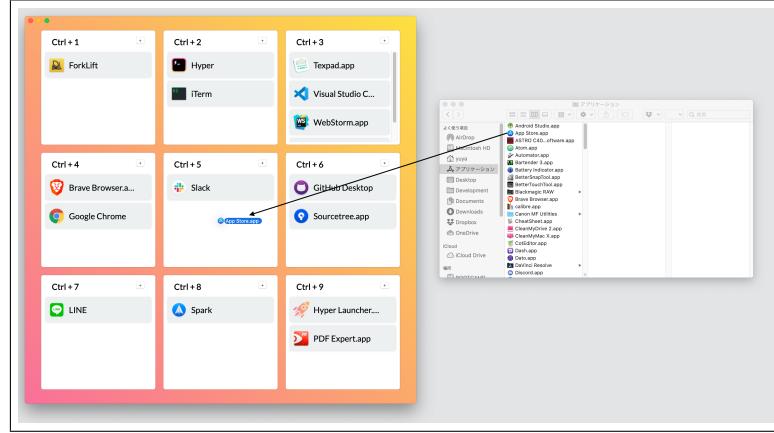


図 2.4: ドラッグアンドドロップによる追加

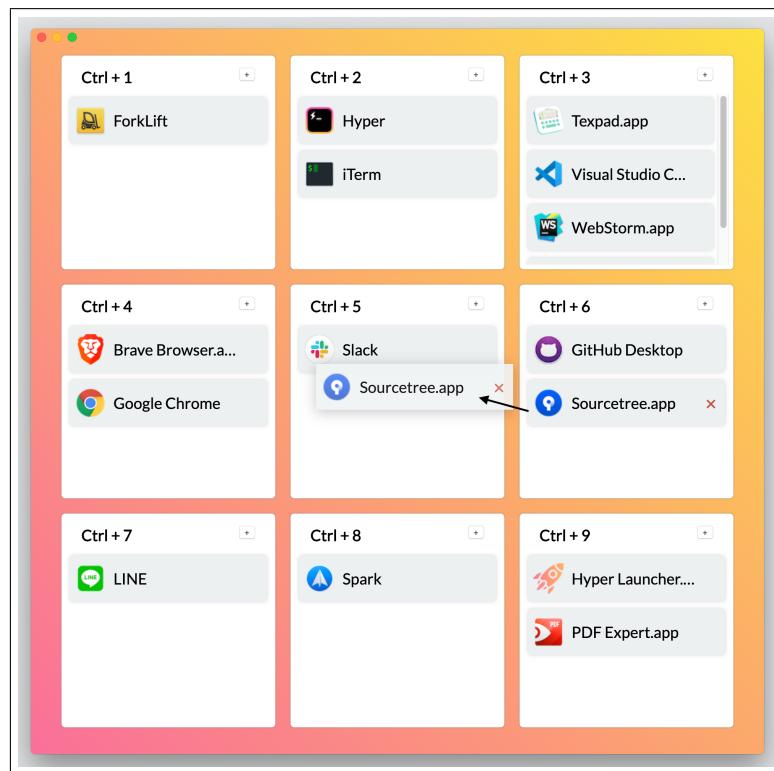


図 2.5: 登録の変更



図 2.6: 登録の解除

第3章 実装

本章では、第2章で述べたシステムの設計を受け、Hyper Launcher の実装について述べる。

3.1 システム構成

Hyper Launcher はユーザーが実際に操作するためのクライアントアプリケーションと、登録したデータを保存しておくためのデータストアから構成される。構成図を図 3.1 に示す。

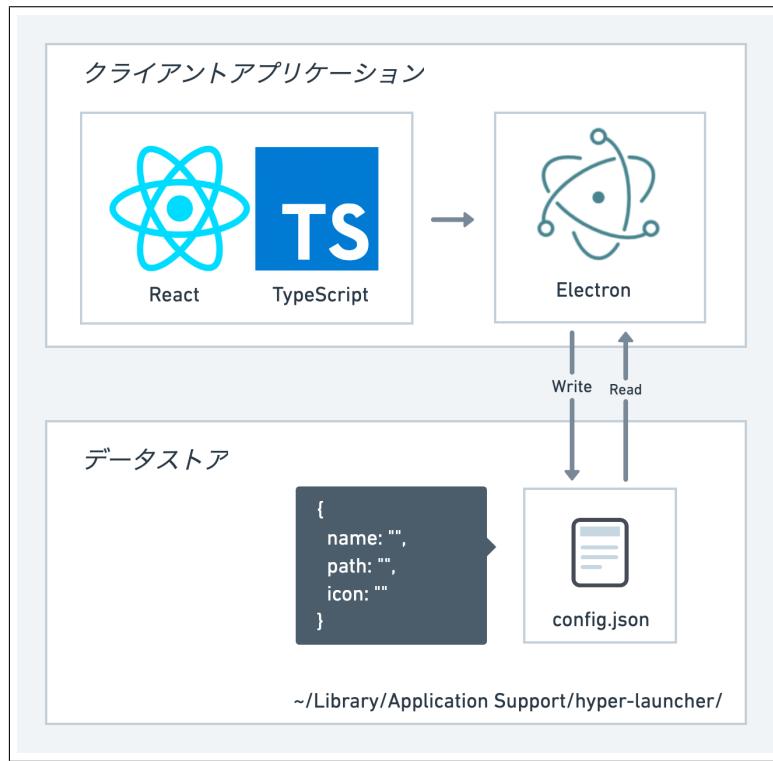


図 3.1: システムの構成

3.2 クライアントアプリケーション

クライアントアプリケーションは TypeScript 及び React などの Web 技術によって実装されており、macOS のデスクトップアプリケーションとして動作する。

3.2.1 Electron

Electron¹は HTML、JavaScript、CSS を用いてクロスプラットフォームなデスクトップアプリケーションを作成することができるソフトウェアフレームワークである。Electron 自体は Chromium と Node.js を使用しており Web 技術のみで開発が完結するのが利点である。Hyper Launcher では Electron の globalShortcut という API を利用し、Hyper Launcher が非アクティブ状態でもホットキーの入力を特定できるようにしている。アプリケーションの起動やアクティブ化なども標準の機能で実装できるため、ランチャーに適したフレームワークと言える。

¹<https://electronjs.org>

3.2.2 AppleScript

Web 技術のみでは実装が難しい部分については AppleScript を使用した。macOS には osascript と呼ばれるシェルスクリプトが存在し、これを利用することで Node.js から AppleScript を実行することができる。AppleScript を使用することで、起動中のアプリケーションや特定のアプリケーションがアクティブかどうかといった情報を取得することができ、これによってデスクトップアプリケーションとして十分な機能を実装することが可能となった。ただし実行する毎に子プロセスを立ち上げるため、何も考えずに使用してしまうとラグが発生してしまうこととなる。またこれは macOS のみで使えるものであるため、Electron の利点であるクロスプラットフォーム対応はできなくなってしまった。

3.3 データストア

ユーザーが登録した情報は、Electron が定義するユーザーデータ領域 (`/Library/Application Support/hyper-launcher/`) に JSON ファイルとして保存されている。データのやり取りが全てローカルで完結するため、オフラインでも使用できようになっている。データはホットキーの数字をキーにしアプリケーション情報の配列を値として持ったオブジェクトになっており、アプリケーション情報にはアプリケーションの名前、パス、そして base64 エンコードされたアイコンの 3 つが含まれている。これにより単一のキーに対して複数のアプリケーションを登録することが可能となっている。以下にその例を示す。ただし icon の base64 文字列と中間部分は長くなるため省略した。

ソースコード 3.1: config.json

```
1 {
2   "shortcut": [
3     "1": [
4       {
5         "name": "ForkLift",
6         "path": "/Applications/ForkLift.app",
7         "icon": "*****"
8       }
9     ],
10    "2": [
11      {
12        "name": "Hyper",
13        "path": "/Applications/Hyper.app",
14        "icon": "*****"
15      }
16    ],
17    ...
18    "9": [
19      {
20        "name": "Hyper_Launcher",
21        "path": "/Applications/Hyper_Launcher.app",
22        "icon": "*****"
23      }
24    ],
25  ]
26}
```

第4章 実運用及び評価

本章では、Hyper Launcher を実際に運用した結果および評価について述べる。

4.1 筆者の運用

筆者は本システムを開発段階を含め半年以上利用した。日常的にホットキー型のランチャーを使用しているため覚えやすさについての評価は出来ないが、その他については大きな利点を実感することができた。特に登録の作業は格段に便利になった。今までではアプリケーションを選択した後に割り当てるキーを選択する必要があったが、そのフローが一つにまとめられたことにより、余計なことを考える必要がなくなった。また一つのホットキーに対して複数のアプリケーションを登録できるため、これまで以上に柔軟な操作が可能となった。19に対してカテゴリーごとに分けてアプリケーションを管理し状況に応じてそれらを使い分けたり、よく使うアプリケーションの組み合わせを一つのホットキーにまとめたりと活用の幅が広がった。ただし、複数登録の操作についてはまだまだ改善の余地があると考えられる。

4.2 第三者の意見

研究室のメンバーのうち、ホットキー型のランチャーをあまり使ったことのない2名に実際に利用してもらった。

4.2.1 アプリケーションの登録について

アプリケーションの登録にドラッグアンドドロップを採用した点については好意的な意見が得られた。普通は面倒な作業でも Hyper Launcher では楽しく設定できるようになっていると評価できる。また事前にホットキーがしていされているということも、設定の敷居を下げている要因の一つになっている。ランチャーに限らずホットキーを登録するインターフェースは煩雑なものが多いため、これは大きなアドバンテージだと考えられる。

4.2.2 ホットキーの覚えやすさについて

設定自体は簡単になっていたが、覚えやすさという点ではまだまだ難しいという意見が得られた。事前にホットキーが固定されてはいるものの、そのホットキーとアプリケーションの対応を覚えなければならぬことには変わらず、ホットキーの使用に慣れていない人にとっては依然として敷居が高いということがわかった。

4.2.3 ホットキー自体の有用性について

今回はホットキー型のランチャーをあまり使ったことのない人に試してもらうことができたが、しばらく使っていく中でその便利さを実感してもらうことができた。特にホットキーの入力一発で素早く起動できるという点は好評だった。しかし今後も使用を続けるかとなるとまだ障壁があり、初心者でも段階的に使いやすくなっていくような仕組みが必要なのではないかと感じた。

4.2.4 問題点

その他、細かい問題点についても意見が得られた。以下にそれを示す。

- 複数登録している場合の操作がわかりづらい。直感的でない。
- 削除時の確認はいらない。不可逆の変化ではないので簡単に消せたほうが良い。
- ドラッグアンドドロップの移動にバグがある。
- MacBook の場合小指で入力する必要があるため、Control キーの入力が難しい。

これ以外にも痒い所に手が届かないという指摘はいくつかあり、そういった部分については個別にカスタマイズできるような仕組みが必要であると感じた。

第5章 議論

本章では、Hyper Launcherについての議論を行う。

5.1 関連研究

5.1.1 ExposeHK

ExposeHK[2] はホットキーの使用を助けるシステムで、モディファイアキーを長押しすることで対象アプリケーションに割り当てられているホットキーをツールバー上にプレビューできるというのだ。これにより非習熟者であっても段階的にホットキーを覚えていくことが可能となる。このようなプレビュー機能については実際に使ってもらった中でも欲しいという意見があった。Electron では長押しの実装ができず断念したが、この機能の重要性は高いと言える。プレビューの表示方法についても様々なバリエーションが考えられ、全てを羅列するだけでなく隣接するホットキーの情報を表示するなど Hyper Launcher に最適な方法を模索したい。

5.1.2 HotStrokes

HotStrokes[1] はホットキーの有用性は認めつつより習得しやすく且つ GUI メニューよりも効果的な手法として、トラックパッドを利用したジェスチャによるコマンド入力を提案している。確かに視覚的に記憶しやすいジェスチャをコマンドに対応させるという試みは興味深いが、そもそもトラックパッドの使える環境は限られており、ホットキーに比べ汎用性に劣ると考えられる。インターフェースの大きな変革がない限り、ExposeHK のようなホットキーの習得をサポートするような手法を取り入れるほうが効果的だと言える。

5.2 展望

5.2.1 ネイティブアプリケーション

今回は筆者のスキルの都合上 Electron を使用した開発になったが、やはりネイティブアプリケーションと比べると実装に限界を感じる部分があった。今後様々なインターフェースを試していくためには、Swift 等を利用したより高度が開発が求められる。

5.2.2 習熟のサポート

実際に使用してもらい一番感じたのは、やはりソフトウェアが使用者に寄り添っていく必要があるということだ。Hyper Launcher ではあまり革新的な手法を提案できなかつたが、関連研究はたくさんありそれらを組み合わせて実用レベルで色々な機能を試していきたい。

5.2.3 熟練者向けの機能

Hyper Launcher で実装した機能のうち、その操作性を向上させるものについては一定の効果がみられた。これらは熟練者にも有用なものであり、こういった機能もより拡充させていきたい。

特に慣れていくほど細かい好みが表れてくるものなので、モディファイアキーの変更や複数アプリケーションの操作方法、新たなコマンドの登録等、日々活用している人たちにさらなる利便性を提供できるようにしていきたい。

第6章 結論

本章では、本研究で得られた成果について述べる。

既存のホットキーを利用したアプリケーションランチャーの問題を明らかにし、新しいシステム Hyper Launcher を設計/開発した。その中で新たな機能を取り入れ実際に運用することで様々な利点/欠点を得ることができた。特に

- ドラッグアンドドロップを利用した簡単な設定。
- 単一のホットキーに複数のアプリケーションを登録できる機能。

等については、既存の不便を解消しより柔軟で強力に操作できるということが明らかになった。しかし未習熟者のホットキー利用をサポートする機能についてはまだまだ改善の余地があり、今後は既存の手法や新たなアイデアを組み合わせさらなる利便性の拡充を目指していきたい。

謝辞

本研究を進めるにあたり、学部2年次よりご指導いただきました増井俊之先生に深く感謝いたします。また、研究会や日々の活動を通して貴重な意見やアドバイスをくださった方々にこの場を借りて感謝を表します。

参考文献

- [1] Wenzhe Cui, Jingjie Zheng, Blaine Lewis, Daniel Vogel, and Xiaojun Bi. Hotstrokes: Word-gesture shortcuts on a trackpad. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [2] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. Promoting hotkey use through rehearsal with exposehk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, p. 573–582, New York, NY, USA, 2013. Association for Computing Machinery.