

VR 環境における文字入力方法の 切り替えのためのソフトウェアアーキテクチャ

M2022SE005 清原隆一

指導教員：沢田篤史

1 はじめに

近年 VR (Virtual Reality) の普及と応用範囲の拡大が進んでおり、今後は様々な文字入力が増加すると考えられる。特に業務や教育などの分野では文字入力を行う場面が多く、VR アプリケーションを快適に利用するために文字入力をより使いやすくする必要がある。

VR デバイスの発展に伴い、VR 環境上での文字入力方法の研究が盛んに行われている。一方で、個々の入力方法をそれらが有効に働く場面ごとに切り替えて利用する方法は提供されておらず、利用者は 1 つの限定された入力方法を使用している。また、複数の入力方法を統一的に VR アプリケーションに組み込むための共通基盤も提供されていないので、複数の文字入力方法を VR アプリケーションに組み込んで、適切に切り替えて使用できるようにするためには手間がかかる。

本研究の目的は、VR 環境上で柔軟な文字入力方法の切り替えが可能なアプリケーションの構築支援である。ここでの柔軟性とは、個々の文字入力方法で対処できない状況に対応できるシステムの性質であるとする。構築支援を行うために、VR アプリケーション開発で利用可能な共通の基盤を提供する。

構築支援を行うにあたり、技術課題は次の二点である。

- 文字入力方法の柔軟な切り替えが可能なアーキテクチャを設計する
- アーキテクチャの妥当性を評価する

本研究では利用者の状態をコンテキストとしたコンテキスト指向と MVP モデルを組み合わせることで文字入力方法の柔軟な切り替えが可能なアプリケーションのアーキテクチャを設計する。文字入力の機能を、MVP モデルにおけるそれぞれのコンポーネント内にサブコンポーネントとして定義する。また、MVP の各コンポーネントに横断するメタレベルのコンポーネントとして、利用者の状態を表すコンテキストと、コンテキストに基づいて入力の切り替えを行うコンポーネントを配置する。アーキテクチャの妥当性を確認するために、複数の文字入力場面に対して文字入力方法を切り替えて使用できるアプリケーションを想定した具象アーキテクチャを設計する。具象アーキテクチャに基づいてプロトタイプを実装、使用することで評価する。

妥当性評価の結果、プロトタイプで文字入力方法の柔軟な切り替えが可能であることから、提案するアーキテクチャは VR アプリケーションにおける文字入力方法に対して柔軟性を担保しているといえる。

2 VR 環境上での文字入力における課題

2.1 背景

インタラクティブなシミュレーションが行えるという点から、VR を教育や業務で活用することが検討されている。現在 VR アプリケーションでの文字入力はパスワードの入力や検索などの短いタスクの場面でのみ用いられることが多い。これは VR 環境での文字入りにわずらわしさが伴うことに起因するが、長文の作成など様々な場面での文字入力がストレスなく行えるようになれば文字を扱う場面が多い教育や業務の VR アプリケーションをさらに拡大できる。

VR 環境上では入力デバイスが違ったり空間把握が難しいという問題から、現実世界で入力デバイスを用いる場合と同等の文字入力を行うことは難しい。多様な VR デバイスを活用した文字入力方法の提案がされているが、複数の入力方法を切り替えて使用する方法や、入力方法をアプリケーションに組み込むための共通基盤は提供されていない。したがって現状では各アプリケーションで実装されている 1 つの入力方法しか使うことができない。

利用者の好みや習熟度など、様々な要因によって適切な文字入力方法は変化する。VR 環境上で快適な文字入力を行えるようにするためには、様々な状況に応じて適切な文字入力方法を切り替えて提供できるようにする必要がある。

2.2 先行研究

Marco ら [1] は様々な VR デバイスを用いた 6 つの文字入力方法を実装して利用者へアンケートを行い、利用者の状況に適した入力方法を提案できる意思決定ツールを作成している。ここでの利用者の状況とはデバイスの有無や酔いやすさ、手の疲れやすさなどの VR 特有の問題である。一般的にはコントローラから発せられるポインタを用いる方法が最も速い文字入力方法だが、利用者の状況によっては最適な文字入力方法とならない場合がある。

Rafael ら [2] は VR アプリケーションを設計するためのソフトウェアアーキテクチャとして MVC (Model-View-Controller) モデルを提案している。View は画面の更新を行い、Controller は VR デバイスの操作を Model に通知し、Model はビジネスロジックや 3D オブジェクト、ライト、サウンドなどのデータを保持する。MVC モデルに基づいて分割したことにより、VR アプリケーションの開発労力の軽減やメンテナンスのしやすさの向上といった利点がある。

2.3 問題点

VR 環境上での文字入力において、解決されていない問題は次の 2 点である。

- 利用者の状況に合わせて複数の文字入力方法を切り替えることができない
- 文字入力方法をアプリケーションに組み込むための共通基盤がない

文字入力の切り替えは、利用者の様々な状況に対応して柔軟に行えるようにする必要がある。ほとんどの場合は入力速度が最も速い入力方法が最適であるが、使うためのデバイスがない場合や長時間使用した場合の疲労感、利用者の好みなど、状況によって最適な入力方法は変わることがある。音声入力も有効な方法だが、パスワード入力や声を出せない場面など、最適でない場合がある。

文字入力方法をアプリケーションに組み込むための共通基盤は、様々なアプリケーションにおいて状況に応じた文字入力方法の切り替えを実現するために必要である。多様なデバイスや仮想キーボードを共通基盤に実装できるようになることで、切り替え可能な入力方法を容易に追加できる。

3 目的と技術課題

3.1 目的

本研究の目的は、VR 環境上で文字入力方法の切り替えが可能なアプリケーションの構築支援である。そのためには、柔軟性を担保できるソフトウェアの構造を明らかにする必要がある。ここでの柔軟性とは、個々の文字入力方法で対処できない状況に対応できるシステムの性質であるとする。

柔軟性を担保するソフトウェアの構造を明らかにしたのち、VR アプリケーション開発で利用可能な共通基盤として提供する。これにより開発者は VR アプリケーション開発に柔軟な文字入力方法の切り替えを組み込むことができ、アプリケーションの構築支援が可能となる。

3.2 技術課題

アプリケーションの構築支援を行うにあたり、課題は次の 2 点である。

- 文字入力方法の柔軟な切り替えが可能なアプリケーションのアーキテクチャを設計する
- アーキテクチャの妥当性を評価する

本研究では、柔軟性という非機能特性の達成を、ソフトウェア構造の問題であるにとらえる。すなわち、状況に応じた文字入力方法の切り替えを行うソフトウェアアーキテクチャを定義することを技術課題として設定する。

設計したアーキテクチャの妥当性は、そのアーキテクチャにより入力方法の柔軟な切り替えが可能であるかという視点で行う必要がある。またアーキテクチャに基づいて実現・提供されるアプリケーション基盤によって、効率的に入力方法の切り替え機能を組み込めるか、あるいは入力方法や切り替え方針の変更が容易かという視点からの評価も必要である。

4 文字入力切り替えシステムのアーキテクチャ設計

4.1 課題解決のアプローチ

3.2 節で挙げた課題を解決するために、文字入力方法の柔軟な切り替えが可能な VR アプリケーションのアーキテクチャを設計する。本研究では、コンテキスト指向と MVP モデルを組み合わせたアーキテクチャを設計する。

アーキテクチャ設計では、まず利用者の状況に応じた文字入力方法の切り替えに関する共通の構造を概念アーキテクチャとして定義する。次いで、特定の利用状況や文字入力方法によって概念アーキテクチャを具体化した具象アーキテクチャを定義する。妥当性の評価は、複数の文字入力場面がある VR アプリケーションを想定し、状況に合わせた文字入力方法の切り替えができるかどうかを確かめることで行う。

4.2 コンテキスト指向

コンテキスト指向とは、文脈に依存した振る舞いをモジュール化するためのプログラミングの方法である [3]。時間や場所とともに変化する外部環境や内部状態をコンテキストとし、コンテキストに依存した振る舞いをモジュール化することで、コンテキストの変化によって振る舞いを柔軟に切り替えることができる。

コンテキスト指向を用いて、利用者の状態をコンテキストとし、複数の文字入力方法をモジュール化して振る舞いとして切り替えられるようにすることで、利用者の状態に応じた柔軟な文字入力の切り替えを実現できる。

4.3 MVP モデル

MVP (Model-View-Presenter) モデル [4] は MVC モデルの派生パターンであり、Model と View が独立しており、両者の通信を Presenter を介して行う構造である。MVP モデルはモバイル向けソーシャルゲームアプリケーションの設計と相性がよく、利用者が GUI (Graphical User Interface) を操作するという考え方で設計を行うことができる。

VR 文字入力方法における重要な機能は利用者の手を表す VR デバイスとキーボードの 2 つであり、これらを View に位置し、文字決定のロジックや予測変換を Model に位置することで、利用者が GUI であるキーボードに VR デバイスである手で触れることで操作するという考え方で MVP モデルに基づいた設計を行うことができる。

4.4 概念アーキテクチャ

MVP モデルを基盤として、文字入力システムを 3 つの主要な層に分割し、メタレベルのコンテキストがそれぞれの層を切り替えるアーキテクチャを設計する。図 1 に提案するシステムの概念アーキテクチャを示す。メタレベルに位置するコンテキストは、利用者の状態を反映して変化する。アーキテクチャは、このコンテキストに依存して文字入力方法を切り替える構造となっており、状態に合わせた柔軟な文字入力方法の切り替えが可能となっている。

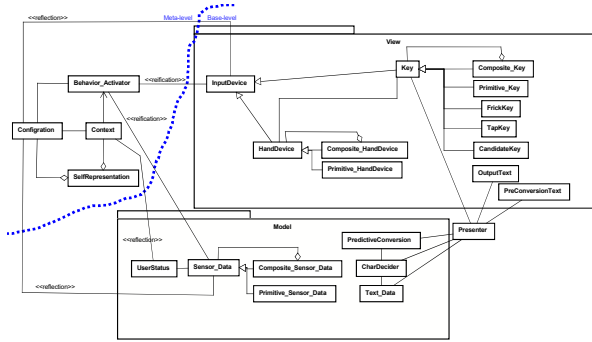


図 1 概念アーキテクチャ

メタレベルには SelfRepresentation が位置しており、システムの自己記述を行う。Context は利用者の状態、Configuration はシステムの構成を表している。利用者の状態が変化すると自身を変更し、Behavior_Activator が決められた文字入力方法を有効化する。

文字入力方法は Rafael ら [2] の MVC モデルから文字入力方法に必要な機能を抽出し、MVP モデルを基に分割する。それぞれの層の役割をまとめたものを次に示す。

- View: Hand_Device と Key, 出力テキストの表示を行い、指先に触れた Key の文字を Presenter に通知する。
- Model: Text_Data と Sensor_Data, User_Status を保持しており、入力された文字の情報から文章を更新し、Presenter に通知する。
- Presenter: View から送られた通知を受け取って Model の文字決定メソッドを起動する。また、Model から送られた通知を受け取って View のキーボードや文章を変更するメソッドを起動する。

4.5 具象アーキテクチャ

図 1 のアーキテクチャの妥当性、すなわち状況に合わせた柔軟な文字入力方法の切り替えが可能であるか評価するために、複数の文字入力場面が存在する VR アプリケーションを想定した具象アーキテクチャを設計する。利用者の状態であるコンテキストを明らかにし、先行研究の文字入力方法を再現するためにデバイスやキーボード具体化する。

利用者の状態は様々なものがあるが、本研究では次の 3 点をコンテキストとする。

- 文字入力タスク
- 利用者の好み
- デバイスと HMD との距離

以上のコンテキストによって、文字入力方法を切り替える。図 2 に提案するシステムの概要を示す。次にコンテキストと振る舞いの切替について説明する。

文字入力タスクとは、文字入力の場面で想定される文字入力の量である。これはテキストボックスの大きさから入力可能な文字数を読み取ることで、例えば簡単なメッセージ返信と長文の記述とで使う入力デバイスを方法を変えることが VR 環境上でも可能となる。

利用者の好みは VR 文字入力において最も優先されるコンテキストであると考えられる。文字入力方法の好みにはばらつきがあり [5]、習熟度や疲れにくさなども好みの要因に関係するが、それらを全て定量化することは難しいことや、複数の文字入力方法を使用できるという前提があることから、アプリケーションの利用開始時に利用者がタスクに合わせて使いたい文字入力方法を決定できるようにする。文字入力が始まるとタスクに合った文字入力方法が起動し、好みの入力方法を利用できる。

デバイスの距離は利用者が装着している HMD と、文字入力に使用する VR デバイスとの距離である。VR は歩き回って使用することもあり、文字入力に使いたいデバイスや物理キーボードが離れている場合がある。タスクに合った文字入力方法のデバイスが離れている場合は、デバイスの位置を示し、代替の入力方法を起動する。

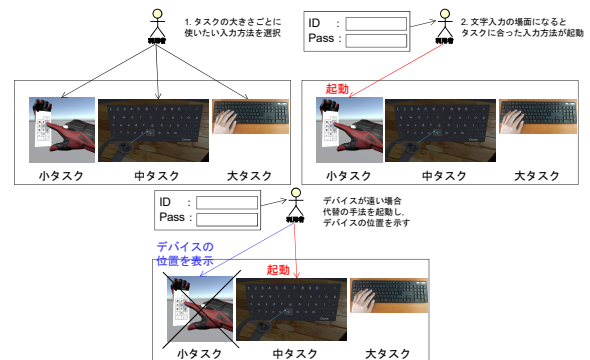


図 2 文字入力切り替えシステムの概要

具象アーキテクチャを図 3 に示す。利用者が多い QWERTY キーボードと日本での利用者が多いフリック入力キーボード、入力速度が速い傾向のあるデバイスを実装できるようにする。キーボードである仮想・物理 QWERTY と仮想・物理フリック、デバイスおよびセンサであるハンドコントローラと Leap Motion を具体化する。

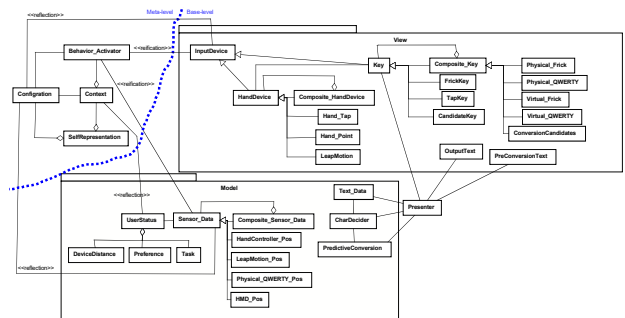


図 3 具象アーキテクチャ

Hand_Device は現在使用している VR デバイスの指先座標を保持しており、仮想キーボードと接触を確認する。物理キーボードを使用する場合は VR 環境上へのキーボードの表示は行わが手デバイスの指先座標との接触は検知せず、文字決定が入力情報を受け取る。

手デバイスとキーボードの組み合わせは自由なので、物理フリック入力 [5]、仮想キーボードを用いたフリック入

力, 物理 QWERTY キーボードを用いた入力, Marco ら [1] が提案する QWERTY キーボードに対するポインタ入力, タッピング入力, フリーハンド入力を実装できる.

5 評価

プロトタイプの実装については, 図 3 の具象アーキテクチャに基づいて行うことができた. メタレベルとベースレベルの分割によりコンテキストの追加, MVP モデルにより文字入力方法の追加を容易に行うことができた.

プロトタイプを使用することで, 具象アーキテクチャに基づいて実装したアプリケーションが利用者の状況に対応して文字入力方法を柔軟に切り替えられるかを確かめる. 評価では, タスクが変わった時に文字入力方法が切り替わるか, また, デバイスが遠い時にデバイスの位置を表示できるか, 位置を表示しているデバイスに近づいたら有効化されるかを確かめる. 評価の結果, それぞれのシナリオで期待した通りの文字入力方法の動的切り替えを行うことができたが, 長時間の文字入力ほどの文字入力方法を用いても強い疲労感を感じた.

6 考察

具象アーキテクチャに基づいて文字入力方法の切り替えシステムを実装することができ, コンテキストの変更や文字入力方法の追加, 変更を容易に行うことができた. また, プロトタイプを使用することで利用者の状況の変化に伴って文字入力方法が動的に切り替わることを示すことができた. このことから, アーキテクチャの妥当性については想定していた使用シナリオの中で柔軟な文字入力方法の切り替えを実現できたと考える.

Marco ら [1] が作成した VR 文字入力方法に存在する特徴の違いについて, 本研究では疲れやすさの具合を利用者の好みであるととらえ, 利用者が使いたい文字入力方法をタスクの大きさに合わせて登録できるようにした. これによって疲れやすさという定量化が難しい状況をコンテキストに含めることができる. 文字入力方法の切り替えは期待通りの挙動を確認できたが, VR の文字入力の快適化については文字入力方法の改善とコンテキストの設定の両方から考える必要がある. また, アーキテクチャのさらなる妥当性の評価を行うために, 他の状態を考慮したコンテキストの追加や文字入力システムを VR アプリケーションに組み込んだ評価を行う必要がある.

ベースレベルの MVP モデルの実装では Presenter が Model と View の通信を全て管理していることから, MVP モデルは単体テストが容易に行えるという点や各モジュールの変更が容易である点に加えてコードの可読性の向上という利点がある. また, 文字入力方法を View である手デバイスとキーボードに分離し, Model に文字入力のロジックを配置することで, 先行研究で提案されている文字入力方法を実装できる共通基盤を提供することができた. 文字入力方法の追加は, HandDevice の場合は指先座標の位置を指定し, Composite.Key は所有する仮想キーの配置を決めれば実装できる. その際他の文字入力方法や Model, メタレベルのコードには影響がなく, Presenter に Composite.Key の入力を受け取るように登録をすれば

使用できる. さらに, 複数の文字入力方法を同時に使用することもできる. このことから, 文字入力方法は容易に追加, 変更ができる. 以上の理由から, VR 文字入力方法の実装に MVP モデルを用いることで保守性の向上が期待できる.

7 おわりに

VR の発展に伴い, 文字入力場面が多様化していく中で, 様々な文字入力方法を利用者の状況に合わせて切り替えて利用できると, 文字入力方法をアプリケーションに組み込むための共通基盤が必要である.

利用者の状況に柔軟に対応して文字入力を切り替えられるようにするために, VR 環境上で文字入力方法の切り替えが可能なアプリケーションの構築支援を目指した.

構築支援を行うにあたり, 利用者の状況に応じた文字入力方法の柔軟な切り替えが可能なアプリケーションのアーキテクチャを設計することと, アーキテクチャの妥当性を評価することが必要がある.

本研究では, 利用者の状態をコンテキストとしたコンテキスト指向と MVP モデルを組み合わせたアーキテクチャを設計した. アーキテクチャの妥当性を評価するために, タスクに合わせて好みの入力方法を使用できるアプリケーションを想定し, 具象アーキテクチャを設計した. 具象アーキテクチャを基にプロトタイプを実装し, 使用することで評価した.

評価の結果, 文字入力方法の柔軟な切り替えが可能な VR アプリケーションを実装することができた. また, プロトタイプを使用したことで文字入力方法を柔軟に切り替えることを確認できた. このことから, 提案するアーキテクチャで柔軟性を担保することができた.

今後はより多様な文字入力場面で適切な文字入力を提供できるように, コンテキストの追加と文字入力方法の改善の両面からアプローチを行うことで VR アプリケーションの構築支援を目指す.

参考文献

- [1] Marco Speicher, Anna Maria Feit, Pascal Ziegler, Antonio Krüger, “Selection-based Text Entry in Virtual Reality”, *2018 CHI Conference on Human Factors in Computing Systems*, No. 647, pp. 1-13, 2018.
- [2] Rafael Capilla, Margarita Martínez, “Software Architectures for Designing Virtual Reality Applications”, *EWSA 2004: Software Architecture*, pp. 135-147, 2004.
- [3] 紙名 哲生, “文脈指向プログラミングの要素技術と展望”, *コンピュータソフトウェア*, Vol. 31, No. 1, pp. 3-13, 2014.
- [4] Mike Potel, “MVP:Model-View-Presenter the Taligent Programming Model for C++ and Java”, Taligent, Inc, 1996.
- [5] 清原 隆一, 沢田 篤史, 野呂 昌満, “VR 環境における文字入力支援システム”, 第 29 回ソフトウェア工学の基礎ワークショップ (FOSE2022), pp. 61-66, 2022.