

知っておきたいキーワード「VRM」

まえがき

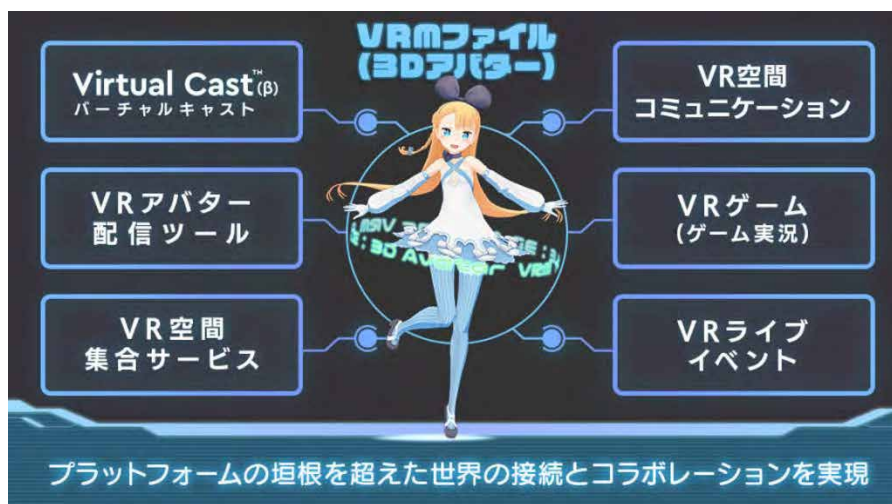
近年、バーチャルリアリティ技術、そしてメタバースが注目されています。メタバースとは meta（超越した）、universe（宇宙）を組み合わせた造語で、コンピュータネットワークの中に構築された 3次元の仮想空間とそのサービスを指します。メタバースの中では利用者は世界中から思い思いのアバターと呼ばれる自分の分身で参加することになりますが、このアバターを定義するためのファイルフォーマットとして標準化が進められているのが、3D アバターファイルフォーマット「VRM」¹です。

そもそもなぜアバターのためにファイルフォーマットが必要なのか？VRMが存在することでどのようなことが可能になるのか？そこには技術的な理由だけでなく、将来のメタバース社会に向けた思想的なバックグラウンドも絡んできます。本稿では、この「VRM」について解説します。

VRM とは

「VRM」のポイントを一言で言うと、仮想空間上の「人型のキャラクターやアバター」において細かいモデルデータの差違を吸収・統一し、アプリケーション側の取り扱いを簡単にするプラットフォーム非依存・横断型の 3D アバターファイルフォーマット、となります。

「プラットフォーム非依存・横断型」というところがポイントで、VRM はプラットフォームや処理系に依存しない汎用のデータフォーマットとして定義されており、かつ、ランタイムに（実行時に）読み込み処理できるよう設計されています。



そのため、たとえば 3D キャラクター作成ツールで作った自分のアバターを、さまざまな

¹ VRM – VR 向け 3D アバターファイルフォーマット <https://vrm.dev/>

会社の、さまざまな VRM 対応アプリケーションで読み込み使用することができるわけです。今まではゲームごと・アプリケーションごとにキャラクターを作ったり選んだりしてきましたが、VRM 対応のアプリケーションであれば同一のモデルデータを「自分のアバター」として持ち込むことができます。そして、各アプリケーションが VRM に対応する際、開発側は特定のプラットフォームや特定の開発環境などに縛られることがないのです。

VR とアバターがもたらす身体感覚

ではなぜこのようにフォーマットを策定までして「自分のアバターを各アプリケーションに持ち込む」ことにこだわるのでしょうか。実は、さまざまな研究により、人がアバターを用いて身体を操る、ということはその身体を通し世界を感じることであり、身体の特徴に応じて知覚も変容するということがわかっています。さらには知覚的な影響に加え認知的な影響の効果も生じ、さまざまな認知や行動にまで影響が現れることもわかっています²。つまりバーチャルリアリティの世界でアバターが変容する、ということは単なる見た目だけの問題にとどまらずその人の認知にまで影響する、言い換えれば自己の変容に相当するのです。

一方で、これまで仮想空間におけるアバターについては各アプリケーションやゲームの中に閉じた設計が主でした。自分自身、つまりプレイヤーの姿はアプリケーションごとに異なることがあたりまえで、自己の姿を変化なく持ち運ぶという概念はそもそも存在すらしなかったのです。今後メタバースのような世界観が普及していくことを考えると、この自己の姿と認知の関係は遠からず課題になってくるでしょう。認知のレベルで統一した自己を保つためには、さまざまなアプリケーションの上でも統一した身体を維持することが必要になる。つまり自己の姿のポータビリティが求められる。このような思想に基づき、「メタバース時代には身体（アバター）のポータビリティが必要である」という社会への問題提起を含め設計し提案されたのが VRM、というわけです。

アバターのポータビリティ

身体（アバター）のポータビリティ、と一言で言いますが、アプリケーションからデータを利用するにはどのように制御されるべきかを想定しなければなりません。そこで VRM での取り扱いには「『人型』のキャラクタやアバター」に限定することにしました。策定時に人型以外のアバターにも一定の需要（猫になりたい！という人も多いでしょう）があることは想定されましたが、アプリケーションからの制御方法があまりにも多種多様になると対応アプリケーション開発側の負担が大きすぎるため、VRM では、「『人型』のキャラクタやアバター」の取り扱いに限定することにしました。ただし、手足や頭は見た目としてなくても、

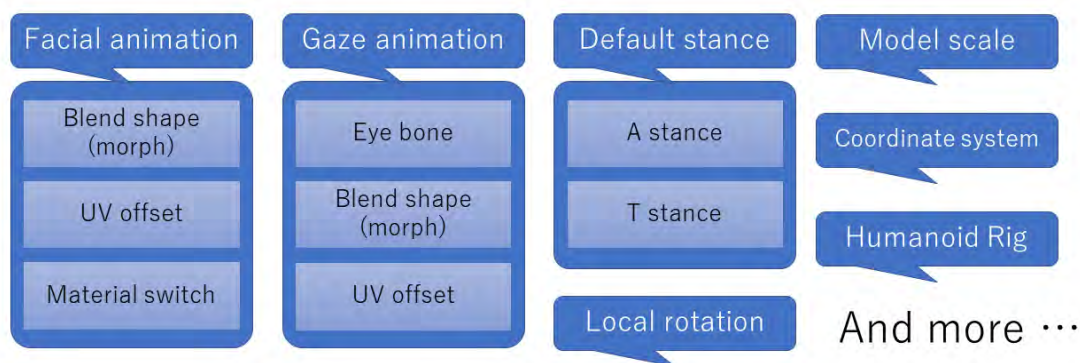
² 鳴海拓志, ゴーストエンジニアリング: 身体変容による認知拡張の活用に向けて, 認知科学, Vol.26, No.1, pp.14-29, 2019

制御データ上アプリケーションから手足や頭を持った「人型」として制御さえできれば、見た目は人型に縛られません。また、表情の制御も人型を基準とはしますが、人間の顔を直接制御することを前提とすると、見た目が人間の顔をしていないアバターへの表現の幅が狭まるため、VRM データでは喜怒哀楽の「感情」をまず設定し、それら感情に相当する表情をキャラクタ表現として定義できるようになっています。アプリケーションからは特定の感情を指定するだけで各キャラクタならではの表現で感情を示すことができるのです。

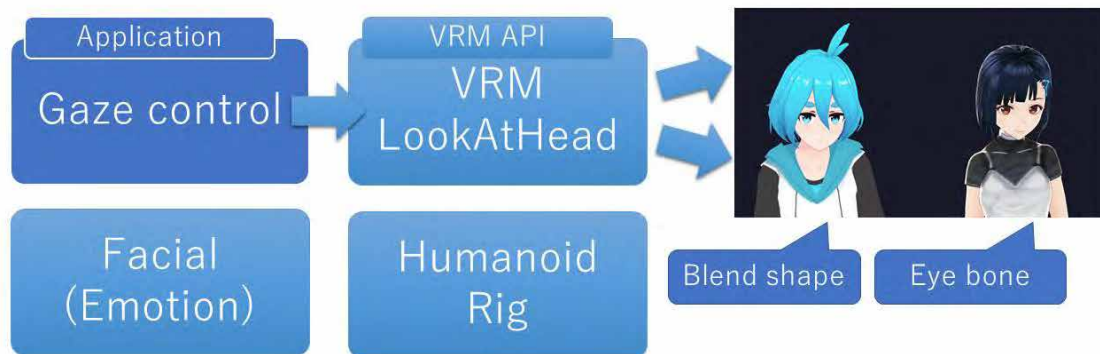
そして、この感情表現の指定に加え、このポーズで視線はここを見ている、などと指定することができます。このように、抽象化し統一した制御方法を設定することでアプリケーションからのアバターの取り扱いを簡便化し、結果として、VRMアバターはどのアプリケーションに持ち込んでも同じような取り扱いが可能となります。結果として、VRM で作られたアバターをどのアプリケーションに持ち込んでも同じように取り扱うことができるようにしています。

データ構造の中間層

VRM ではアバターの表情に対して「感情」という中間層を設けましたが、こういった中間層の定義が必要である技術的理由もあります。一般に 3D データはその作成ツールの特性やデータ作者の癖などの理由で、その構造が多様であり統一的な制御が難しいという課題があるのです。



たとえば、キャラクタの表情の変化をモデリングする際、「モーフ・ブレンドシェイプ (頂点の移動)」という手法によりモデルの形状を変えて表現する場合もあれば、「テクスチャ (画像)」を動かして表現する場合があります。「マテリアル (素材)」を差し替えて表情の変化を表現するデザイナーもいるでしょう。どのパターンでもアプリケーションから実現したいのは「表情 (感情) の変化」でしかありませんが、キャラクタ 3D モデルの制御は手法によってまるで異なってしまいます。



ほかにも、視線の表現手法（眼球を動かすのか目の画像を動かすのか）や基準となるモデルデータの姿勢（初期姿勢がどのようなポーズか）、さらには 3D データの座標系についても標準が定まっていないのが実情です。

そこで、VRM では基準（座標系やスケールなど）を定義し、また表情など手法が複数存在するものに対しては、差異を吸収する中間層として「どの手法を使用するか」と「目標値」のデータをあわせて持つことで、どのような状況で作られたデータも制御できるようにしているのです。

仮想空間で利用するアバターならではの定義

VRM の定義のなかには「仮想空間で利用するアバター」のための非常に特徴的な設定（視点の位置や一人称視点のときに非表示とするべきデータの設定など）もあります。たとえば、単なる 3D のキャラクターデータとしては必要ありませんが、バーチャルリアリティでアバターとして利用するためには、アバターを自分で「着る」（自分がアバターの姿となって活動する）ため、視点などの定義が必要となります。自分の視線はアバターのどの位置にあるのが正しいのか、また、その視点からレンダリングしたときに、顔の裏側や髪の毛などが視界に入って邪魔にならないよう、一人称視点のときにはどの部分を非表示にするべきか、といった情報をデータとして持つことが可能になっています。これらのような一人称視点を前提とした情報は、今までの 3D モデルデータでは必要とされなかったものであり、仮想空間におけるアバターファイルフォーマットならではの定義と言えましょう。また、VRM の規格上、利用者にとってそのアバターをどのように扱ってよいのかがわかりやすくなるよう、ファイルには利用条件などのライセンス情報が必ず内包されるようになっています。

アバターの「人格権」

ライセンス情報のうち、アバターのキャラクターとしての振る舞いについて定義したものを「アバターの人格権」と呼んでいます。人格権、とここでは呼称しましたが、もちろん法律上の定義としての人格権ではなく、VRM のライセンス設定において近い概念として表現したものととなります。ソフトウェアのライセンスや著作権などではあまり意識されません

が、キャラクターデザインなどでは、一般的にそのキャラクターに対してどのような振る舞いをさせるか（性格付けなど）をコントロールしたいという必要があります。そこで、モデルデータの作者は VRM ファイルを作成するときに、「このキャラクターに『人格を与える』ことができるのは誰か（操作する権利を誰にライセンスするか）」、「このキャラクターにどのような表現を許すか」を設定できるようになっています。



たとえば、アバターに人格を与える、つまりアバターを「着て」操作することができる対象を「作者のみ」と設定した場合は、たとえその VRM ファイルを所持していても作者以外が操作することはライセンス上許されません。また、操作する権利を「誰でも」と設定されていたとしても、「過剰な暴力表現をすることを不許可」と設定されていれば、そのアバターを用いて過剰な暴力を表現することは許されません。このように、単なるキャラクターのモデルデータというだけでなく、キャラクターにどのような「人格」を与えることができるのか、という観点が VRM には含まれているのです。

VRM コンソーシアム

この「VRM」は、株式会社ドワンゴで開発し、ファイルフォーマットの定義とそのハンドリングをする Unity³向け標準ライブラリの提供を 2018 年 4 月に開始しました。その後、VRM 開発チームは株式会社バーチャルキャスト⁴へ移籍しました。2019 年 4 月に VRM の標準化を目指し一般社団法人 VRM コンソーシアム⁵を設立、現在 VRM の仕様策定ならびに標準ライブラリの開発を行っています。

VRM では基盤とするファイルフォーマットとして「glTF 2.0⁶」を採用しており、VRM はすべて glTF の拡張ならびに制約として定義されています。VRM コンソーシアムではこの

³ Unity <https://unity.com/>

⁴ 株式会社バーチャルキャスト <https://corp.virtualcast.jp/>

⁵ VRM コンソーシアム <https://vrn-consortium.org/>

⁶ The glTF 2.0 Specification

<https://github.com/KhronosGroup/glTF/tree/main/specification/2.0>

「glTF 2.0」を策定しているクロノスグループ⁷と連携しています。

VRM 1.0・VRM 1.0β

2018年4月時点の当初のVRM定義にはいくつかのエラーや考慮漏れがありました。また、その後利用が拡大するにつれあらたな需要や必要な情報などがわかってきたこともあり、VRMコンソーシアムでは、あらたな定義としてエラーを修正し、今後必要になると考えられる機能を追加した「VRM 1.0⁸」の策定を進めています。これまで利用してきたVRM定義は「VRM 0.x」と分類され非推奨となり、今後はVRM1.0が正式な規格となっていく予定です。

このVRM 1.0は複数の部品にわかれて策定が進んでおり、それぞれの部品が、規格として確定している(正式版)か、確定したとは言えないが充分利用できる(β段階)か、まだ大きく変更される可能性のある(ドラフト段階)かのいずれかの状態にあります。

2022年1月時点で、一部β段階に達していない部分(VRM_node_constraint: コンストレイント、拘束機能)も残されていますが、全体を通して利用できる段階にあるため、この機能を除いたものをVRM 1.0βとして公開しています。

⁷ Khronos Group <https://jp.khronos.org/>

⁸ <https://vrm.dev/docs/vrm1/>