

論文番号 fm2016-01

# Web 技術を用いた 3D 系譜図自動描画システムについて

13RD044 尾崎 愛, 13RD194 古姓 竜樹, 13RD211 馬目 哲大

指導: 藤本 衡 准教授

提出日: 2017 年 1 月 27 日

## 概 要

従来から系譜図には 2D 表現が用いられてきた。系譜図は登場人物が増えることで関係を示す線が交差してしまうという欠点がある。この問題を 3D で系譜図を表現することによって解決できないか試みた。

本研究では、3D 系譜図自動描画システムを作成した。その上で第三者からの評価を得るために、プラットフォームを Web として開発を行った。さらに、作成したシステムを評価する方法を提案した。

その結果、3D にしたことのみで視認性が上がったとはいえないが、拡大縮小などの機能を使用することでユーザーにとって見やすい情報量に適宜変更することが可能になった。

## 目次

概要	2
1 はじめに	4
2 系譜図について	6
3 ギリシャ神話について	6
4 WHIteBasE について	8
5 システム概要	9
5.1 データベース . . . . .	10
5.2 ノードの座標計算 . . . . .	12
5.3 Web ブラウザ上における系譜図描画 . . . . .	16
5.4 Web ページデザイン . . . . .	18
6 アプリケーション評価方法	22
7 考察	24
8 まとめと課題	24

## 1 はじめに

系譜図は、歴史上の人物などの出自を説明するためにどんな両親から生まれた子供か、また兄弟やその人物の婚姻相手を示す資料として現代でも利用されている。記録や表示に用いる媒体が革や紙などの平面であったことから、系譜図には歴史的に 2 次元 (2D) 表示が用いられてきた。しかし、系譜図では登場人物が増えるにつれて、関係を表す線の交わりが増えるという欠点がある。例えば 1 人の人物に対して婚姻相手が 3 人以上いて、なおかつそれぞれの間で子供が誕生している場合は関係を結んだ線が交叉してしまう。また、兄弟が数多くいるとその子孫を表現する際に樹形図のようにだんだんと横に広がってしまうという問題点がある。

こうした点を改善する方法として、系譜図の 3 次元 (3D) 表示への移行が考えられる。情報端末におけるユーザインタフェースの高度化により、平面ディスプレイにおいても 3 次元構造をある程度自由に動かすことが可能になっている。さらには、3D モニタや仮想現実 (VR) 用ゴーグルなど、より直感的に 3 次元構造を認識するための表示デバイスも開発されている。

そこで本研究では、系譜図の 3D 表示を実装することで、3D 表示が系譜図に適しているか否かを検証する。3D 表示を用いた系譜図表示 Web アプリケーションは、筆者らの知る限り過去に実現した例がない。また、ユーザに対する使いやすさなどを考慮し、表示プラットフォームとして Web を用いることとした。

本研究では系譜図の例として、ギリシャ神話に登場する神や人物を対象とする。ギリシャ神話は複雑な関係から成り立っているため、本研究の目標とする複雑なデータ

の表示をテストするためには適当と考えられる。それらにおいて発生する問題点を解消しつつ、3D 表示の系譜図を実現化することを目標とする。

以下 2 節では 2D 表示での系譜図についてを、3 節ではギリシャ神話における系譜図表記の課題を、4 節では本研究で用いたデータ構造モデルである WHIteBasE を、5 節では本研究で作成したアプリケーションを、6 節では第三者から作成したアプリケーションの評価を得るためのアンケートの提案を説明する。

執筆分担について、尾崎が 2 節「系譜図について」、3 節「ギリシャ神話について」、5.1.1 節「データベースの作成」、6 節「アプリケーション評価方法」を担当した。古姓が 5.1.2 節「データ追加・変更」、5.4 節「Web ページデザイン」、8 節「まとめと課題」を担当した。馬目が 4 節「WHIteBasE について」、5.2 節「ノードの座標計算」、5.3 節「Web ブラウザ上における系譜図描画」を担当した。

## 2 系譜図について

古来より、代々の系統を表現するために系譜図が利用されている。特定の人物の出自を説明するために両親、兄弟、その人物の婚姻相手、先祖や子孫などを示す資料として現代でも利用されている。系譜図は特定の家庭における家督相続のみならず、植物の品種改良の過程や、分離・合流を繰り返す政党の成り行きを表現することもできる。

典型的な系譜図の表現方法は次の通りである。また、図 6 にも示す。歴史的に用いられてきた媒体の制約上、系譜図が表現すべき時間推移と系統間の相互関係を平面で表現する必要があった。そこで 2 人の人物を横に結ぶことで婚姻関係を、その線から下に伸ばすことで上が親を、下が子供を示す。すなわち基本的には下方向に進むにしたがって世代が進んでいく。

図 1 親子関係を示す一般的な系譜図

## 3 ギリシャ神話について

本研究では 3D 表現する系譜図の例として、ギリシャ神話に登場する神、人物その他の存在に関する婚姻・親子関係を扱う。ギリシャ神話には数多くの神々や英雄、怪物が登場する。その神々は人智を超えた現象を引き起こしたり、人間のように恋愛や争いを繰り広げたりしていたことが伝えられている。数多くの登場人物がいるということは、すなわち数多くの関係があったと言える。ギリシャ神話の中でも特に有名なのは全知全能の神ゼウスであるが、絶対的に強力な力を持つ神々の王であることに加

えて、非常に浮気者であったことも有名である一因となっているだろう。その結果ゼウスは非常に多くの子孫を残している。

これらの関係を従来の系譜図に書き起こすと膨大な情報量となってしまう、その上同じ名前の神が複数箇所に登場するようになってしまう。このため、ある一柱の神と血縁的または配偶的な関係を持つ神々を全て探し出すのは非常に困難である。神々における系譜図を作成するにあたり留意すべき課題を以下に挙げる。

#### 複数の婚姻関係と近親婚

上で述べたように一柱の男神に対して妻となる女神は一柱とは限らない。また逆に一柱の女神に対して多数の男神と婚姻関係が存在する例もある。また配偶者が近い血縁者であることは珍しくなく、同じ両親から生まれた兄妹で婚姻関係を持っていたり母と息子で持っている例もある。ただしこれらの課題はギリシャ神話特有というわけではなく、近親婚は近代以前ではしばしば見られる事例であり、複数の婚姻関係は現在でも一般的に起こることである。

#### 数世代にわたって同じ神が登場する

一部のエピソードにおいて戦死する例もあるが、基本的に神々は不老不死である。人間における系譜図は下方向に進むにしたがって世代が進んでいくが、神々における系譜図はそうであるとは限らない。数世代にわたって同じ神が登場することがあり、それに伴って関係がより複雑化している。

#### 男女のペアから誕生したとは限らない

特定の神の身体の一部から誕生したものや、女神が自らの力のみで子を産んだものもある。例えばウラーノスの切り落とされた男性器にまわりついた泡

から誕生した神もいる [?]

#### 後に名前が変わる

生前と死後では違う名前となる例がある。例えば人間であったセメレーはゼウスの雷光に耐えられず死んでしまうが、その時に身籠っていたゼウスとの息子であるディオニューソスが死後の世界からセメレーを連れ戻した後セメレーは女神となり、その後天に昇りテュオーネーと名を改めた。

#### 親が 1 組とは限らない

様々な異説により、親が複数パターンにわたるものもある。

#### 性別がないものもある

本研究に使用したデータの中には性別のない生命体や無機物、概念も含まれている。例えば恒星や遊星といったものや、ガイアが生み出した山などがこれにあたる。

データ表現を決定するにあたっては、以上の点を考慮する必要がある。

## 4 WHIteBasE について

本研究では、前節で示したような複雑な人物や関係をデータベース化するモデルとして、WHIteBasE モデル [?] を用いる。

WHIteBasE(Widespread Hands to InTErconnect BASic Elements) とは、系譜図における関係性を扱うための管理手法として、杉山らが 2011 年に提案したものである [?]. 2D の系譜図における関係性を示す線分が交叉することを考慮し、婚姻関係と



子の関係を一つのイベントとして不可視結節点 (図 2) を用いて管理される。この不可視結節点を WHIteBasE と呼ぶ。ここで、個人のノードを管理するための不可視結節点を使った結合モデルを図 3 に示す。

WHIteBasE モデルは、ノードを結合するための 3 種類の「鍵穴」を持つ。このうち 2 種 ( $S_L, S_R$ ) はそれぞれ 1 つずつ存在し、婚姻関係にある 2 ノードを結合する役割を持つ。またこの婚姻関係から生じた  $N$  個の子ノードを結合するために、鍵穴  $D_k (k = 1, 2, \dots, N)$  が用いられる。実際の表示位置としての左右や、男親・女親の違いは順不同である。また、 $D_k$  は下向きに子のノードを結合するための「鍵穴」である。

個人ノード (Individual Node) は 2 種類の鍵  $A, M_j (j = 1, 2, \dots, N)$  を持ち、これらの鍵と WHIteBasE の鍵穴により、不可視結節点に個人を結合することを表す。 $A$  (Ascendant) は鍵穴  $D_k$  に結合して親への鍵を、 $M$  (Married) は  $S_L$  または  $S_R$  に結合して婚姻の鍵をそれぞれ表す。子の個人ノードはそれぞれの鍵  $A$  を用いて WHIteBasE と結合される。婚姻関係の個人ノードに対しては鍵  $M_j$  を用いてそれぞれ WHIteBasE に結合する。

このモデルを用いた座標計算については、5.2 節で述べる。

図 2 2D での系譜図表示スタイル

図 3 WHIteBasE モデル

## 5 システム概要

本研究で提案するシステムは、クライアント (Web ブラウザ)、アプリケーションサーバ (データベース + 座標計算スクリプト) からなる。本システムの基本設計を図 4

に示す。

クライアントは系図を操作するためのユーザインタフェース (UI) を提供する。クライアントが JavaScript によってアプリケーションサーバにノードの座標計算のためのリクエストを送信する。データベースは個人ノードに関するデータを保有しており、アプリケーションサーバがデータベースを参照し、計算した座標とともにクライアントに JSON データとして送信される。その JSON データをもとに、クライアントが three.js を用いて系図を描画する。データベースの扱いについては 5.1 節、ノードの座標計算については 5.2 節、クライアントの描画システムについては 5.3 節で詳しく述べる。

図 4 システムの基本設計

## 5.1 データベース

本節では、ギリシャ神話の系譜図から作成したデータベースと、データベースへのデータ追加や変更を行うデータベース管理プログラムの機能について説明する。

### 5.1.1 データベースの作成

神々の関係について記入したデータベースの作成を目指した。データベースは千葉 [?] が作成した A4 用紙 388 枚におよぶ紙面上の系譜図をメインとして、一部は『神統記』 [?], 『ギリシャ神話』 [?] を参照する。データベースは個人ノードに必要な項目を持ち、それぞれ「ID」「名前 (日本語)」「名前 (英語)」「名前 (ギリシャ語)」「性別」「親

の ID」「婚姻相手の ID」「参照した元となるデータの場所」「同名で別人の可能性がある場合の識別番号」とする。

「ID」は他の神を参照する際に使用する。名前が重複するものもいるため、名前による参照は行っていない。名前の項目に日本語、英語、ギリシャ語の 3 言語を用いたのは千葉からの要望のためである。「性別」は男もしくは女を示す値が入力されていない場合は不明とする。「親の ID」は子供から親を参照するために記入する。なお「親の ID」については親が一柱のみの場合と男女の両親がいる場合があり、加えて親の組み合わせが数通りある場合もある。「婚姻相手の ID」については、複数の夫もしくは妻がいる場合はそれらを全て記入し、リストとして管理する。「参照した元となるデータの場所」はデータの修正や再度参照する際に紙面から探し出す手間を減らす目的で設ける。「同名で別人の可能性がある場合の識別番号」については、日本語名は同一であっても英語名が異なる場合や、千葉によって同一人物の可能性はあるが敢えて別個に記入されている場合、名前や性別などのプロフィールが不明な場合、異説がある場合は全て別々にレコード [?] を設け、それぞれを別人として扱う。

また、「親の ID」を参照して「子の ID」を抽出させるスクリプトを作成し、それをもとに「子の ID」項目を補完した。

#### 5.1.2 データ追加・変更

Python3 で記述したスクリプトを作成し、それを利用してデータベースへのデータ追加・変更を行っている。新規データベースを作る場合は CSV ファイルを読み込み、データの一括追加を行うことができる。変更・修正をする場合はコンソールで名前を検索し、該当した名前を一度表示してから変更するかを確認する。名前検索し、該当

しない場合はデータを新規に追加する。名前検索した際に複数の名前が該当した場合、該当した名前をすべて表示する。変更したい項目を選択し変更内容を入力する。新規追加の場合、「名前 (日本語)」「性別」「親の ID」「子の ID」「婚姻相手の ID」をそれぞれ入力し追加する。データの変更を確認せずに行うとデータベースの破壊が起きてしまう可能性があるため、変更する場合は必ず確認ができるように変更前・変更後をそれぞれ表示している。さらに新規追加の場合は追加したかを明確にするために確認文を表示する。

データ追加・変更の流れは以下の通りである。

1. コンソールで名前検索する。
2. データベースに存在するか確認する。
3. 存在するなら変更項目選択、しないならば新規登録する。
4. 内容を入力する。
5. 入力内容を確認した後、終了する。

## 5.2 ノードの座標計算

本節では、データベースの WHIteBasE モデル化と座標計算を行うスクリプトについて説明する。座標計算スクリプトは Python3 で記述され、系譜図の描画に必要な座標を計算する機能と同時に、系譜図のデータベースを WHIteBasE モデル化する機能を有する。クライアントで中心に描画したいノードを指定し、アプリケーションサーバにノード名を HTTP 通信で POST し、その情報をもとにアプリケーションサーバが座標を計算し、座標を含む個人ノードのデータをレスポンスとしてクライアントに

送信する。

データベースへのデータ登録については 5.1 節で述べている。系譜図データベースを WHIteBasE モデル化する機能については 5.2.1 節、データをもとに座標計算する機能については 5.2.2 節で説明する。

#### 5.2.1 WHIteBasE モデル化

WHIteBasE モデルは 4 節で述べたように、2D 系譜図表示における線の交わりを考慮したデータベース管理手法である。しかしこの手法は、各ノードをオブジェクトとして考え、不可視結節点のオブジェクトを新たに設けることで、プログラムとの親和性が高まっているため、2D 表示であるかにかかわらず、系譜図を表すデータ構造として適していると期待される。よって、本研究では座標計算を行うためのデータ構造として、WHIteBasE モデルを利用する。

本システムで扱いやすいように、WHIteBasE オブジェクトと個人ノードオブジェクトを作り、WHIteBasE モデルを実現する。WHIteBasE オブジェクトは 4 節をもとに、 $S_R$ 、 $S_L$ 、 $D$  のほか、WHIteBasE オブジェクト ID を要素として持つ。個人ノードオブジェクトは 5.1 節をもとに、個人ノードオブジェクト「ID」「名前 (日本語)」「性別」「婚姻相手の ID」「親の ID」「子の ID」「その他情報」「座標」を要素として持つ。

データベースは MariaDB を用いているので、SQL を記述することでデータベースにアクセスすることが出来る。データには親子や婚姻関係に関するデータが格納されているので、本スクリプトでそのデータを取得する。クライアントから中心に描画したいノード (以下中心ノード) の名前を受け取り、そのノードをキューに追加する。その後次の手順をキューが空になるまで繰り返す。

1. キューの先頭を取り出し、個人ノードオブジェクトを作成する。
2. キューの先頭から取り出したノード (以下注目ノードと呼ぶ) の婚姻関係の数だけ空の WHIteBasE オブジェクトを作成し、婚姻者の個人ノードオブジェクトを作成する。
3. 注目ノードをそれぞれの  $S_R$  に追加し、各婚姻者をそれぞれの  $S_L$  に追加する。
4. 注目ノードの子ノードをすべてキューに追加する。
5. 注目ノードと婚姻者の間に出来た子供を  $D$  に追加する ( $D$  はリスト化する)。
6. 婚姻関係のない子どもがいる場合は、別途空の WHIteBasE オブジェクトを作成し、 $S_R$  に pop したノード、 $D$  に子ノードを追加する。

キューが空になると、中心ノードと血の繋がった、もしくは直接関係のある子方向のノードがすべて WHIteBasE モデル化し、これを利用して座標計算を行う。

### 5.2.2 座標計算

Web ブラウザでグラフィックスを描画する際、画面上での座標を用いるが、WebGL で描画する際は、HTML5 の canvas 要素を利用して描画される。canvas 内は直交座標系<sup>\*1</sup>を持ち、three.js では canvas の中心が座標  $(x, y, z) = (0, 0, 0)$  となり、画面右方向が  $x$  軸の正方向、画面上方向が  $y$  軸の正方向、画面手前 (視点) 方向が  $z$  軸の正方向である。

一般的な 2D の系譜図では、親子関係が上下、婚姻関係は左右に配置されることが多い。本システムでも、階層すなわち世代を明確に区別できるように、子供は  $y$  軸負方向 (画面下方向) に配置し、婚姻関係は原点  $(x, y, z) = (0, 0, 0)$  から遠ざかるよう  $x$ - $z$

---

<sup>\*1</sup> 互いに直交する座標軸を有する座標系である。3 次元空間内においては、 $x$  軸、 $y$  軸、 $z$  軸が互いに直交する。

平面上に配置する。ここで、本システムでのノードの配置例を図 5 に示す。図 5 において、赤線は婚姻関係、青線は親子関係を示す。点線は円周上に並ぶことを明示するための補助線で、実際に系譜図に描画されるものではない。

婚姻関係や子供が複数存在する場合、ノードを中心にその婚姻関係や子供を円状に配置する。子供は、婚姻関係がない親のもとに生まれた場合は親から直接真下 (y 軸負方向) に配置し、そうでない場合は注目ノードの婚姻者の斜め下 (y 軸負方向かつ原点から離れる方向) に配置する。

WHItEBasE モデル化によって既に木構造のようなデータ構造になっているので、子に向かって探索しながら次のような手順で座標を決定する。

1. 注目するノードの婚姻者を、WHItEBasE を用いてすべて探索する。
2. 婚姻者の座標を決定する。注目ノードを中心にした x-z 平面の円周上に等間隔で配置する。
3. 注目するノードと婚姻者の間に出来た子供を、WHItEBasE を用いてすべて探索する。
4. 婚姻者の座標をもとにその子供の座標を決定する。婚姻者の下に、かつ婚姻者を中心にした x-z 平面の円周上に等間隔で配置する。
5. 子供を注目するノードに設定する。

以上の手順を再帰的に繰り返し、すべての座標を計算して個人ノードオブジェクトの要素として追加する。

ここまでで作成された個人ノードオブジェクトのリストを JSON 形式に変換し、

URL エンコードを行ってクライアントに送信する。

図 5 ノード配置例

### 5.3 Web ブラウザ上における系譜図描画

本節では、Web ブラウザ上で表示される系譜図の描画システムの UI について述べる。本システムでは、系譜図描画は JavaScript とそのライブラリである three.js[?] によって行われる。three.js で、指定した場所に Canvas 要素を自動で生成し、その Canvas 要素内に系譜図を描画する。系譜図の描画例を図 6 に示す。

Web ブラウザにおいて 3 次元コンピュータグラフィックスを表示させる際、OpenGL[?] を JavaScript[?] で制御することができる WebGL[?] を用いる。しかし WebGL の API \*2は低レベルであり、そのまま使う際に冗長な準備をしなければならないため、JavaScript ライブラリなどを通して WebGL を利用することで、高レベル API として扱うことができるようになる。このようなライブラリの一つとして、three.js が利用されることが多い。

three.js には様々なジオメトリを作成する機能がある。本システムでは、「ノードを表す立体オブジェクト」「それらを結ぶ線」「ノードの名前」で構成した。ノードを表す立体オブジェクトは、性別で形を変え、男性・女性・不明の 3 つに分けた。不明な性別は、男性・女性で表示しきれない性別も含む。ここで、男性は立方体、女性は球、不

---

\*2 Application Programming Interface の略で、ソフトウェアの機能や管理データなどを他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約を指す。



明は正四面体で表した。なお、不明を正四面体で表す理由として、ひと目で明確に男性・女性と区別できるようにするためである。ノードの色は、対比現象により周囲より彩度を高くすることで注目されやすくなる [?] ことから、何を中心に描いたのかを明示するために、中心となるノードの彩度を高く設定した。また、ノードを結ぶ線は赤線を婚姻関係、青線を親子関係とした。ノードの名前は three.js の TextGeometry を利用した。この際、文字のフォントを別途読み込む必要が有るため、TrueTypeFont<sup>\*3</sup>を JSON 形式に変換する JavaScript 製ツールである Facetype.js[?] を利用して文字フォントデータを扱った。

系譜図描画の Web ページを開いた際、JavaScript によって中心に描きたいノードがアプリケーションサーバに送信され、そのレスポンスとして 5.2.1 節で作成された JSON 形式のデータを受け取り、読み込む。この座標を利用して、各ノードのジオメトリに座標を設定し、各ノードを Canvas 要素に描画する。また、ノード同士を結ぶ線は、婚姻関係と親子関係をデータから参照し、それぞれの座標を端点とする直線を引く。さらに、各ノードの座標に名前を表示した。

Web ブラウザ上で表示された系譜図は Canvas 要素上でマウスによる操作が可能である。これは three.js の TrackballControls.js によって実現している。ここでは、右利き用マウスで操作説明をする。系譜図が表示されている画面で、左クリックを押しながらマウスを動かすと原点を中心に回転する。右クリックを押しながらマウスをうごかすと系譜図全体がパン (x-y 方面の移動) する。また、マウスホイールを動かすと拡大縮小する。回転はキーボードの 'A' を、パンは 'S' を、拡大縮小は 'D' を押しなが

---

<sup>\*3</sup> アップルコンピュータとマイクロソフトが共同開発した、デジタルフォントの符号化方式の 1 つである。

ら、かつ左クリックを押しつつマウスを動かすことでも、同様の操作が出来る。

図 6 系譜図表示の例

## 5.4 Web ページデザイン

Web で公開するにあたり Web ページのデザインを提案する。本研究では、わかりやすくシンプルなデザインの Web ページを作成すること、そして訪問者が迷わず操作できることを目標とした。本研究で作成した Web ページは、HTML, CSS および JavaScript で記述している。単一のページにコンテンツをまとめると訪問者がわかりづらくなってしまうため、情報を分散させるために Web サイトを「トップページ」「概要ページ」「描画ページ」「アンケートページ」の 4 つのページに分けた。以下で各ページの詳細を説明する。

図 7 は訪問者が最初に閲覧するトップページである。このページが基本ページになっており、他ページの説明と選択したページに移動できるようになっている。また、訪問者が移動したいページがわからなくなってしまった場合、このページに戻り、移動先がわかるようになっている。

図 8 は、概要ページである。描画システムの概要が書かれている。訪問者が概要、特徴を想像しやすいように、アイコンを使用して簡潔に説明をしている。

図 9 は、描画ページである。本研究のメインであるギリシャ神話の系譜図自動描画が見られるページである。

図 10 は、アンケートページである。描画システムの訪問者からアンケートを取り、

今後の改善に役立てることを目標とする。また、訪問者がアンケートに答えやすいようにアンケートページは新規タブで開かれるようになっている。

図7 トップページ

図 8 概要ページ

図 9 描画ページ

図 10 アンケートページ

## 6 アプリケーション評価方法

5 節において作成したアプリケーションを第三者からの評価を得るために、5.4 節で作成した Web ページ内にアンケートを設ける。なお質問内容の提案のみで実施は行っていない。質問内容は以下の通りである。

### (1) 回答者のプロフィールに関する質問

回答者の分類を行うためにプロフィールに関する質問を設ける。「年齢」「性別」についての質問と、「ギリシャ神話を知っているかの主観的な評価」についての 4 段階評価、また、「パソコンを利用する頻度」についての 5 段階評価とする。「ギリシャ神話を知っているかの主観的な評価」については、以降の質問に登場する神々の関係を予め知った上で関係が明瞭であったかを評価する目安とする目的で設ける。また、「パソコンを利用する頻度」については、マウスを利用してアプリケーションを操作するため、マウス操作に慣れているか否かで操作感が変化するかを評価する目安とする目的で設ける。

### (2) 特定の神とつながる別の神との関係についての質問

はじめに特定の神を探し出し、そこからその神と血縁もしくは配偶関係のある別の神を辿ることができるか、また、辿った結果はじめに探し出した神とどのような関係があると判断することができたかについての質問を設ける。関係については「父親もしくは母親」「兄弟もしくは姉妹」「息子もしくは娘」「夫もしくは妻」とする。これらの選択肢はすべて複数選択可能とする。はじめに探し出す神は、一定の知名度がある神を個人的に選定する。次に辿る神は知名度を

問わずはじめの神にとって「父親もしくは母親」「兄弟もしくは姉妹」「息子もしくは娘」「夫もしくは妻」「息子もしくは娘かつ夫もしくは妻」のいずれかとなるように選定する。

(3) 上項目を踏まえた関係の主観的な評価

はじめの神から別の神を辿った結果、それぞれの神はどのような関係にあると捉えられたかを判断するのは容易であったか否かという質問を設ける。関係についての質問の正答率と併せて評価する。

(4) 意見感想の自由回答欄

上記の質問はすべて選択式であるため、具体的な意見や感想を任意で回答してもらう目的で設ける。

## 7 考察

本研究ではギリシャ神話の系譜図を描画することを目的としているが、ギリシャ神話は非常に情報が多く複雑な関係が存在しているので、従来の系譜図表示では関係がわかりにくくなってしまうという問題点がある。これは3節で挙げた点を解消することで改善がみられると考えられる。本研究で挙げた問題点のうち線の交わりが増える問題については、系譜図を3Dで表示することによって解消され、関係の把握をしやすくなったと考えられる。[?]において、視覚に入った4~5個のものの個数を数えずに把握する時間は0.2秒で、個数が4~5個を超えると即座の把握は難しくなると記述されている。このことを踏まえると、3Dにしたことのみで視認性が上がったとはいえないが、拡大縮小などの機能を使用することでユーザーにとって見やすい情報量に適宜変更することができると考えられる。

またアプリケーションの評価をいくつかの質問項目で行っているが、系譜図はノードの関係を表すグラフであるため、ユーザがグラフの探索をする際に正しく探索できているかを調査するほかに、特定のノードまでたどり着くまでの時間を計測することで、系譜図の探索をスムーズに行えているかどうか数値での評価ができると考えられる。探索開始から終了までの時間を計測できる機能を付随させることができれば、より一層システムの改善を行えると期待される。

## 8 まとめと課題

本研究では、Webブラウザ上でギリシャ神話を元に系譜図を自動で3D描画するシステムを構築した。本システムでの3D表示により、2D表示では解決できなかった線



の交わりと、複数の婚姻関係・子関係によって系譜図が左右方向に広がることを解消することができた。また、Web ブラウザ上で描画するので特別なソフトウェアを用意する必要はなく、様々な端末で手軽に系譜図を見ることができる。さらに、マウスを用いて視点を変更することが容易となり、インタラクティブな操作が可能のため利用者が系譜図をわかりやすく操作することができると期待される。そして、Web ブラウザで答えられるアンケートを作成し利用者からフィードバックを得ることで、描画システムの改善を図ることができるだろう。

今後の課題としては、親から子への探索、描画はできているが、子から親への探索、描画ができていないのでわかりやすくするためには解決する必要がある。また、3D のため描画された系譜図をマウスを用いて視点変更していくと、どちらの方向が親方向なのか、子方向なのかわかりにくくなってしまう。本研究では、アンケートの実施まで至らなかったのも、実際にアンケートを取って利用者の意見を参考にして、描画部分の改善をしていきたいと考える。アンケートを実施するにあたって、はじめに指定されている人物を見つけ出すことが現状では困難なので、検索機能を実装し特定の人物を発見しやすくする必要がある。

将来的には、ギリシャ神話の系譜図をより利用者にわかりやすくするために VR や拡張現実 (AR) を利用した系図描画システムを構築することも考えられる。

本研究ではギリシャ神話を対象とした系譜図自動描画を行なったが、ギリシャ神話以外を対象として関係性の描画をすることもデータベースに情報を持たせることができれば可能であると考えられる。

## 謝辞

本研究での題材となるギリシャ神話の系譜図を提供してくださり、多くの意見をくださった画家の千葉政助氏、アートフォース株式会社の門山光氏、株式会社画天プロジェクトの井田清氏に、感謝の意を込めて謝辞を送りたいと思います。また、本研究の御指導や実験への協力をして下さいました藤本准教授とシステム評価研究室の皆様に対し、ここに心より深く御礼申し上げます。