

情報メディア工学特論

基本オブジェクト設計

2004/12/14

京都大学高等教育研究開発センター情報メディア教育部門

学術情報メディアセンター連携研究部門(兼任)

大学院工学研究科電気工学専攻(兼任)

小山田耕二

コース概要 (2/2)

- 特異点ベース可視化技術
 - 渦可視化(11/16)
 - 等値面表示高速化、DT-MRI可視化(11/30)
- システム開発技術/OpenGL基礎
 - オブジェクト指向システム開発技術(12/7)
 - 基本オブジェクト設計(12/14)
- 可視化システム実装演習
 - 等値面表示システム(12/21)
 - ボリュームレンダリング表示システム(1/11)
 - 流線表示システム(1/18)

ユースケース図(1)

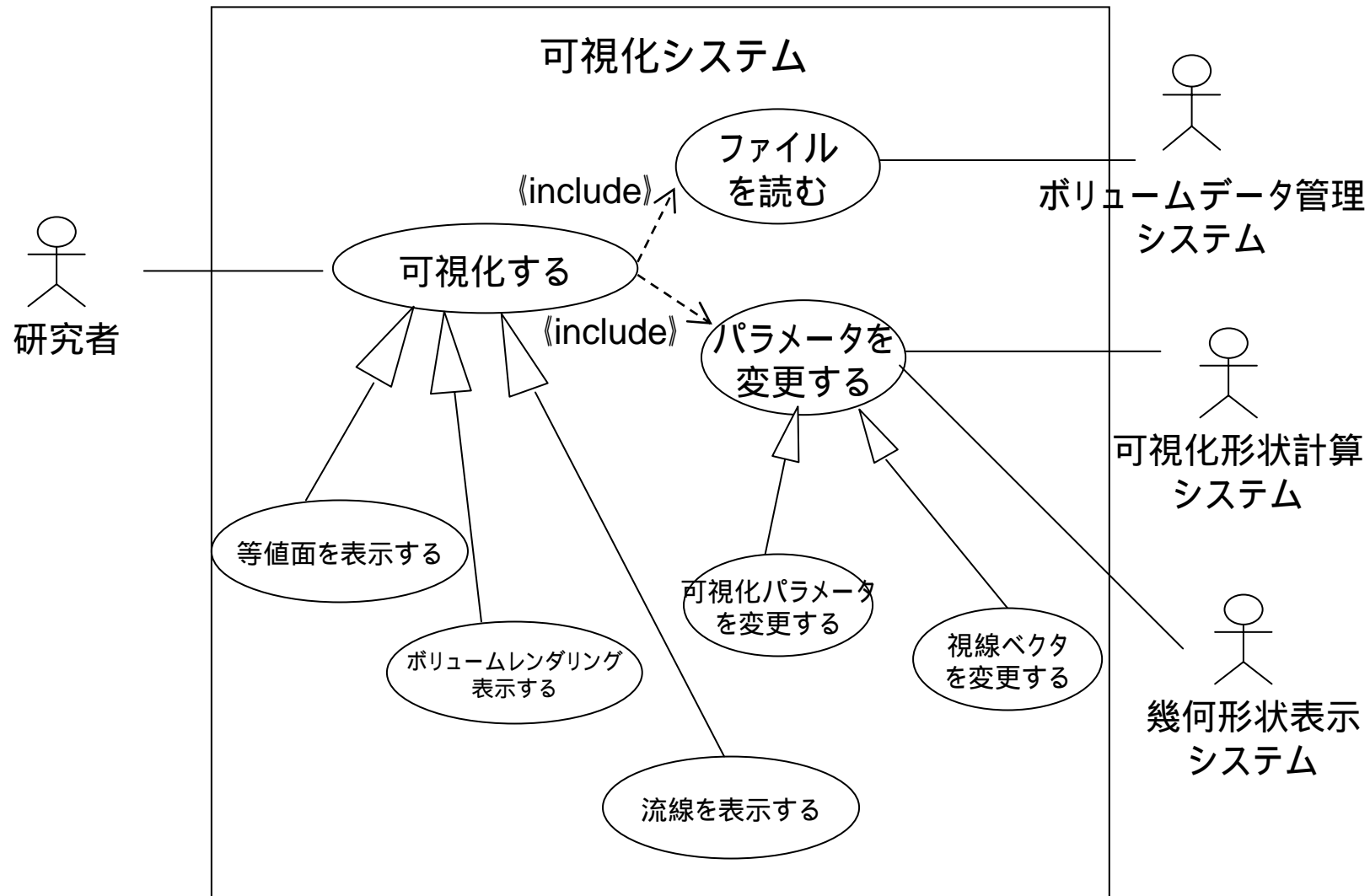
シナリオ

研究者がボリュームデータの格納されたファイル名を指定すると可視化画像を生成する。

研究者は、可視化パラメータ変更し、新たな可視化形状を計算する。

研究者は、可視化形状を注視するための視線ベクタを変更し、新たな画像を表示する

ユースケース図



オブジェクト図

可視化部分のモデル化

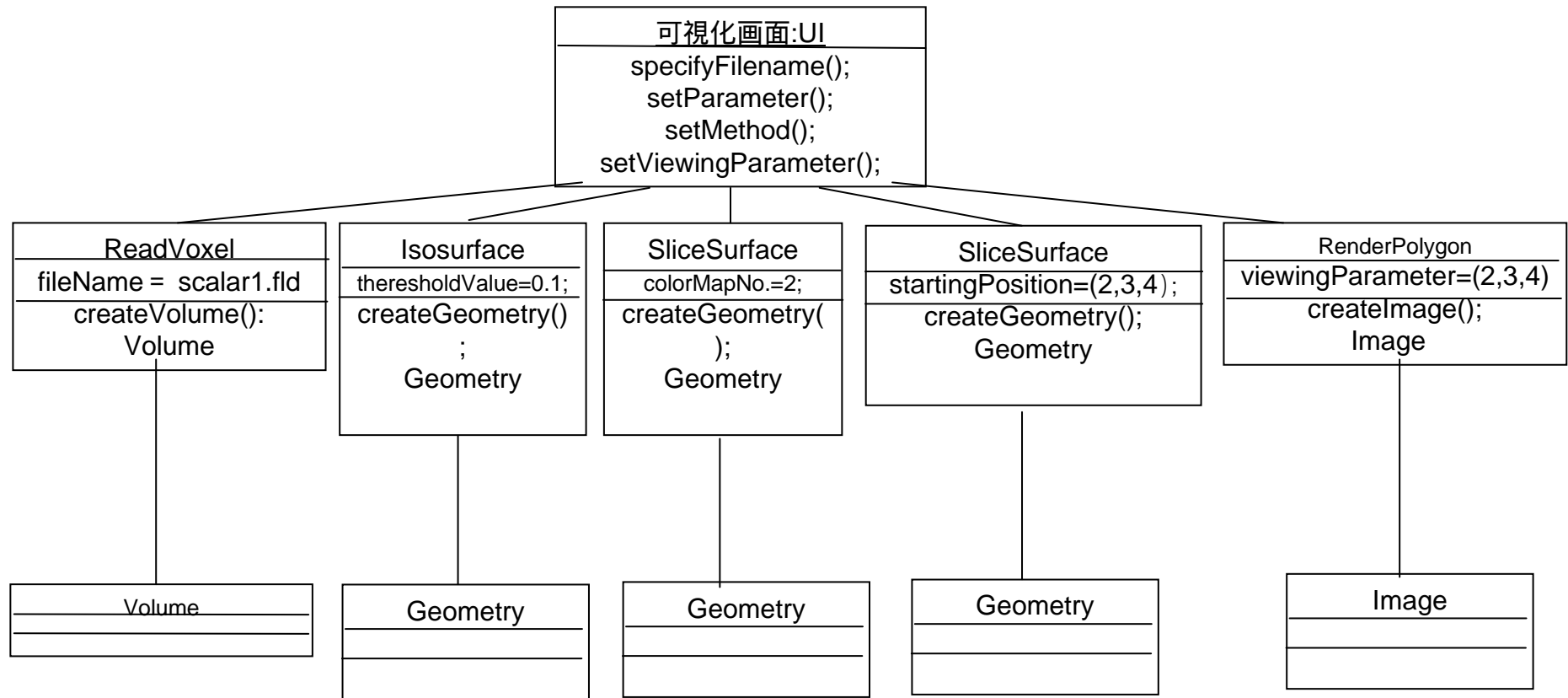
研究者Aさんは、スカラーボリュームデータscalar1.fldを読み込み、閾値を0.1と設定して等値面を表示した。また、カラーマップ2番を利用して、ボリュームレンダリング表示を行った。
また、研究者Aさんは、ベクタボリュームデータvector1.fldを読み込み、開始点を(1,2,5)と設定して流線を表示した。これら表示結果については、視線ベクトルを(2,3,4)とした。可視化業務概要をクラス図やオブジェクト図を用いてモデル化する。

(ヒント)

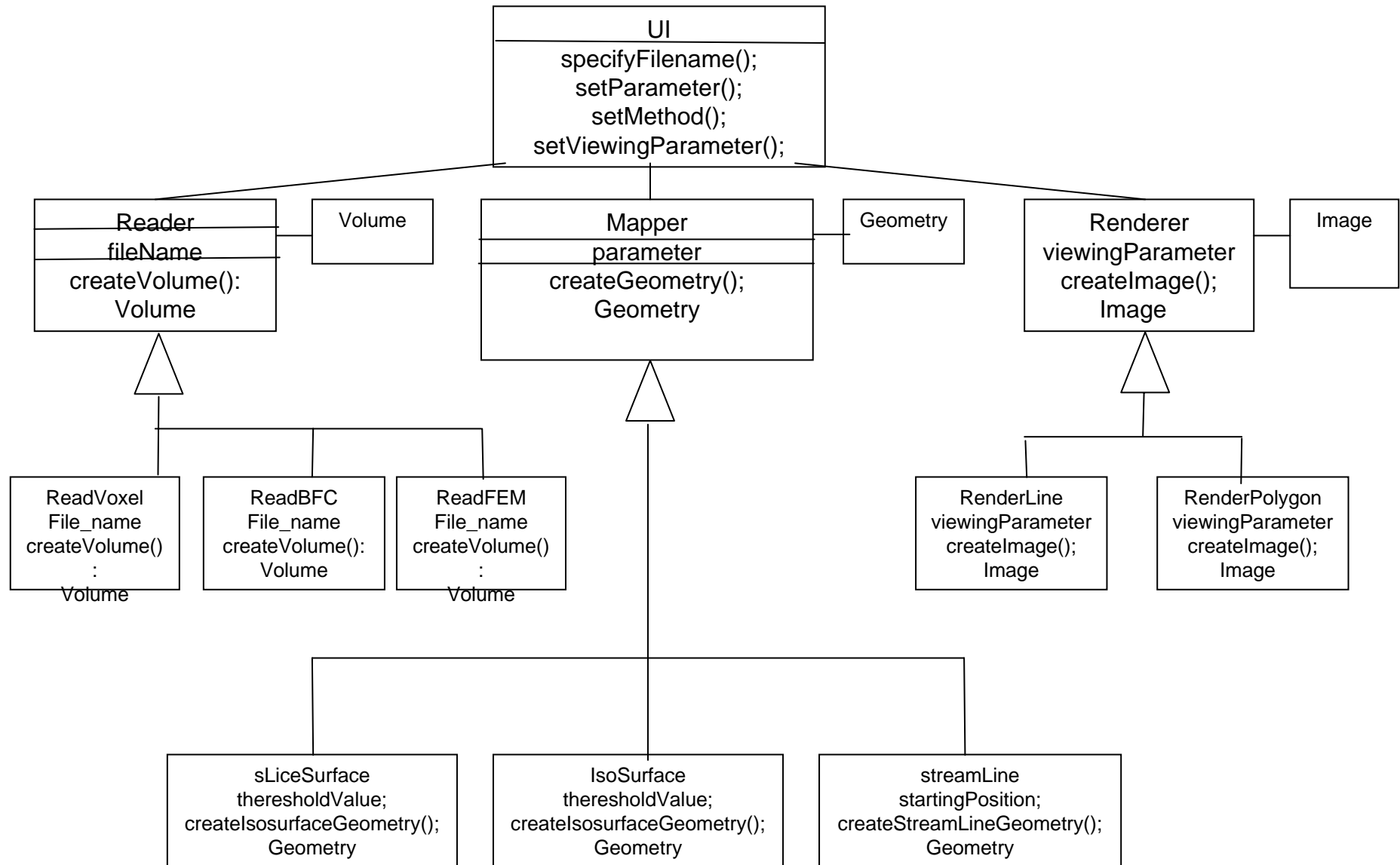
「scalar1.fld: ボリューム」、「閾値0.1の等値面: 可視化形状」、
「開始点(1,2,5)の流線: 可視化形状」、
「カラーマップ2番のボリュームレンダリング表示: 可視化形状」、

これらの関係を考える

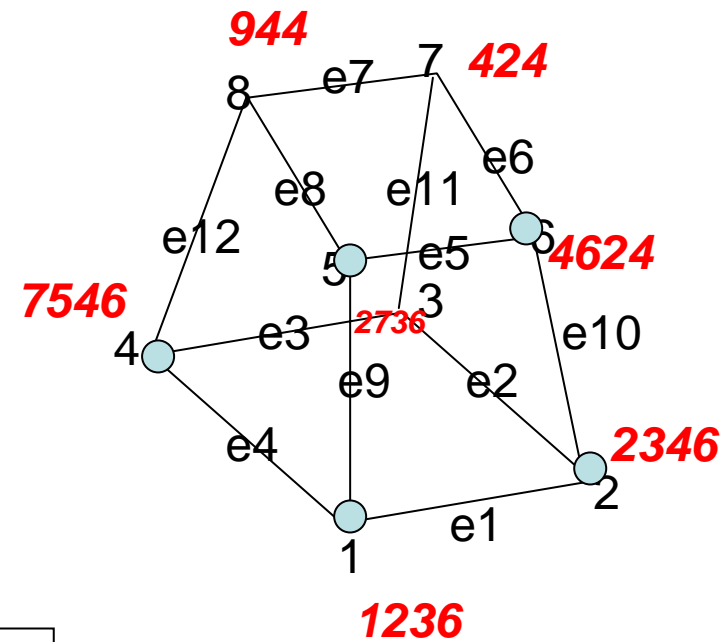
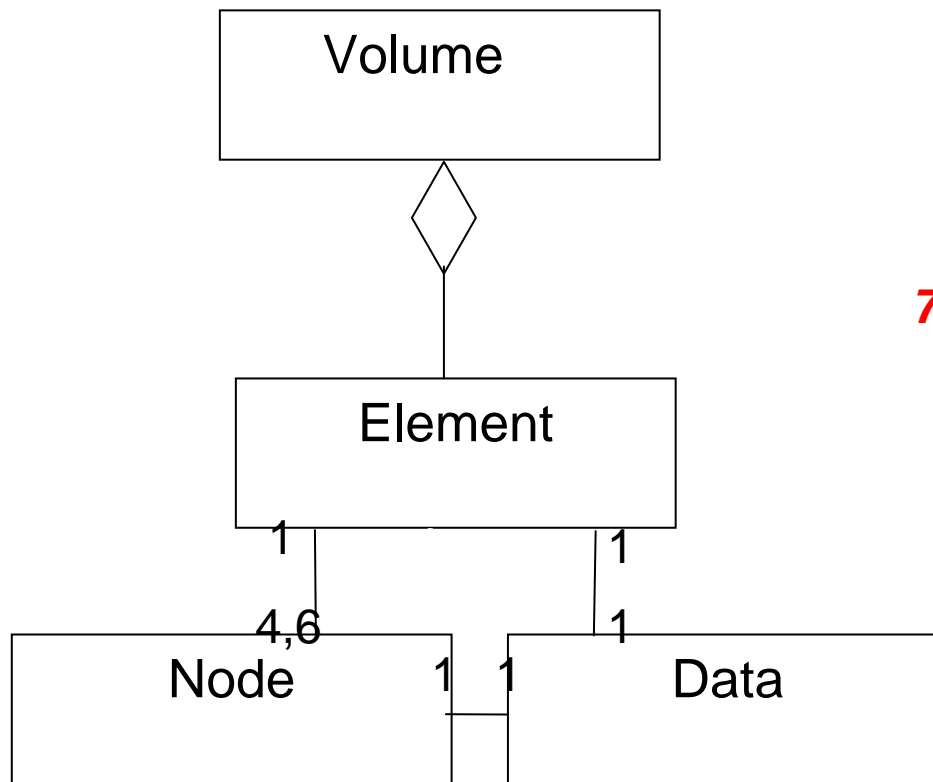
オブジェクト図



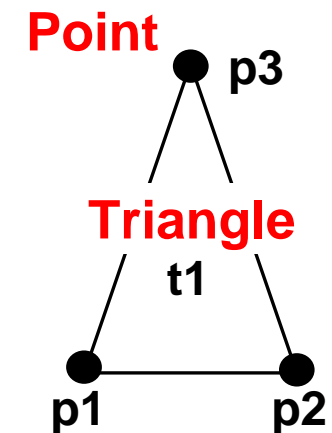
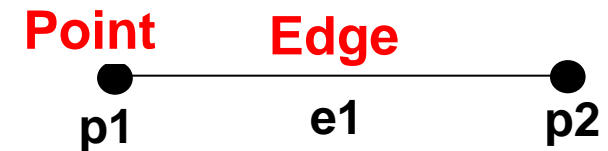
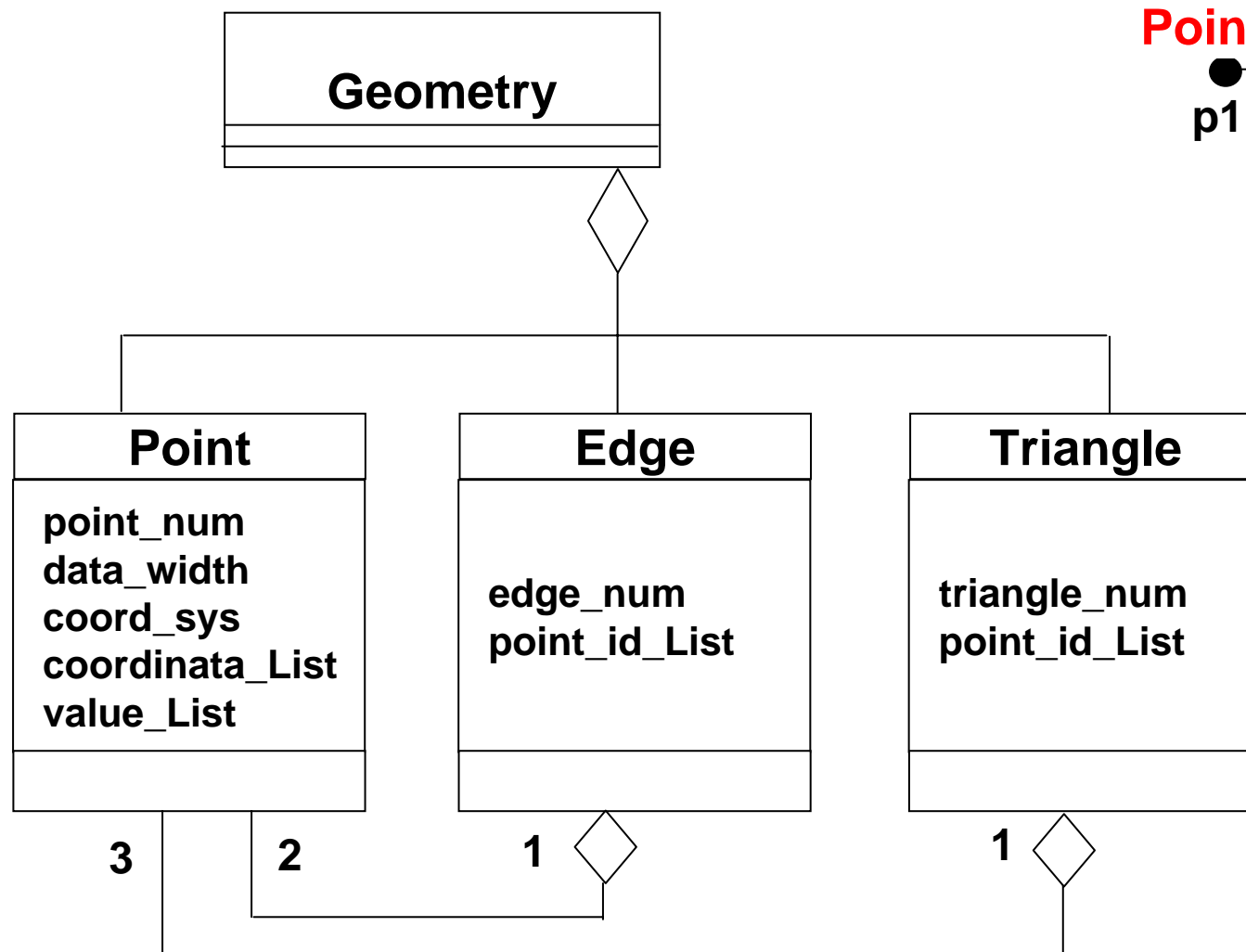
クラス図



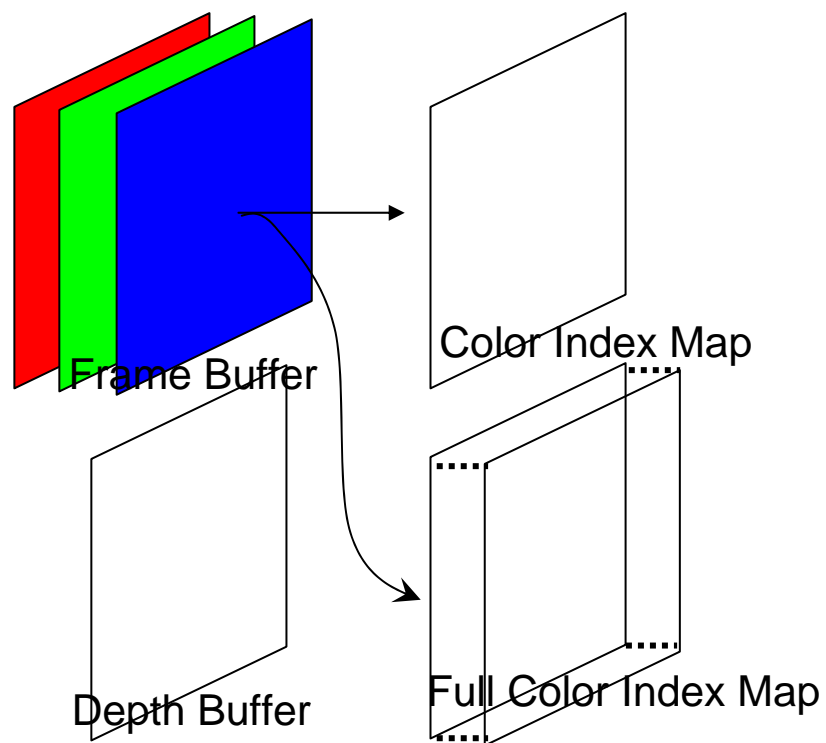
クラス図 ~ ボリュームクラス ~



クラス図 ~ ジオメトリクラス ~



クラス図 ~ イメージクラス ~



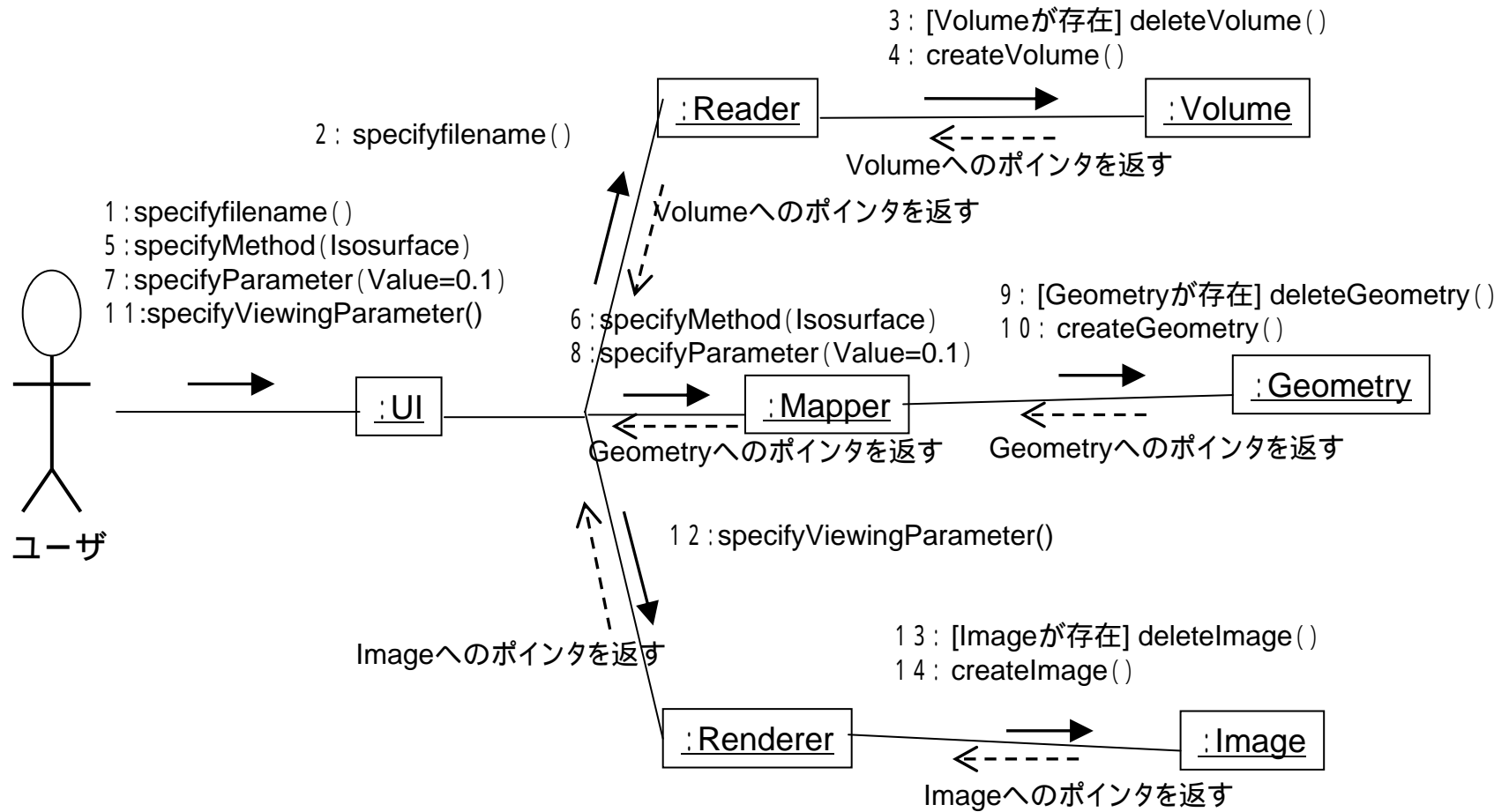
配列クラス

Array		Array
タイプ: 整数か実数か 最大値: 実際に獲得された サイズ レコードサイズ: 配列における 1レコードのサイズ 現在値: 実際に使われているサイズ 項目: 配列中身	→	int array_type; int array_max; int array_num; int array_width; void*array_item;
オブジェクト生成 サイズを大きくする オブジェクト削除 タイプを返す 最大値を返す 現在値を返す レコードサイズを返す	→	create_ARRAY(int, int, int); increase_ARRAY(float, ARRAY*); delete_ARRAY(ARRAY*); get_array_type(ARRAY*); get_array_max(ARRAY*); get_array_num(ARRAY*); set_array_num(int, ARRAY*); get_array_width(ARRAY*); get_array_item(ARRAY*);
先頭ポインタを返す 項目を返す 項目を更新する 初期化する 項目を比較する 項目が存在するかどうか調べる 項目を末尾に追加する 項目を除去する	→	get_array_item_by_number(int, void*, ARRAY*); set_array_item_by_number(int, void*, ARRAY*); make_null(ARRAY*); compare_items(void*, void*, ARRAY*); member(void*, ARRAY*); insert(void*, ARRAY*); delete(void*, ARRAY*);

ハッシュテーブルクラス

Hash	Hash
バケットサイズ データ項目サイズ キー項目サイズ バケットポインタのリスト Quereオブジェクトのポインタ 最初のガベージ領域へのポインタ ハッシュ関数へのポインタ	<pre>int bucketsize; int itemsize; int keysize; struct dataCell * *bucketlist; QUEUE *pGarbage_QUEUE; struct HASH_Memory *pHASH_Memory; int (*calc_key)();</pre>
オブジェクト生成 オブジェクト削除 オブジェクトをコピーする データセルどおしを比較する データセルが存在するか調べる データセルをリストの先頭に追加する データセルを取り除く 全データセルを配列オブジェクトの形で返す 全データセルの個数を返す	<pre>create_HASH(HASH*); delete_HASH(HASH*); copy_HASH(int, void*, HASH*); compare_item(int, void*, HASH*); member(HASH*); insert(void*, HASH*); delete(void*, HASH*); get_item_ARRAY(void*, HASH*); get_item_number(void*, HASH*)</pre>

コラボレーション図



まとめ

- UMLを用いた可視化システムの表現について学習した.

小テスト(氏名:)

- 本日説明したコラボレーション図に対応するシーケンス図を作成せよ。結果を画像またはPPTファイルで送付せよ。