

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

海底コア CT スキャンデータ可視化・情報共有を 可能とするクラウドサービスの構築

—Web ブラウザ向けクライアントアプリケーションの開発—

高 達生

修士（工学）

（コンピューターサイエンス専攻）

指導教員 田中 二郎

2014年3月

概要

独立行政法人海洋研究開発機構の所有する地球深部探査船「ちきゅう」は、海底の地層サンプルであるコア試料の掘削を行っている。掘削されたコア試料(以下、コアデータ)は、船上に搭載されている X 線 CT スキャナにかけられ、DICOM フォーマットのデジタルデータとして保存される。デジタル化されたコア資料はコアデータと呼ぶ。コアデータを活用することは、地球環境変動、地球内部構造、地殻内生命圏等の解明の研究に対し、非常な研究価値がある。前年度の研究開発にて、コアデータを利用し、コア資料を、ブラウザやタブレットで簡単に閲覧できる機能が持つシステムを開発した。本年度の研究開発にて、コアデータの利用だけでなくシステム利用者の間にも、情報共有を可能とするコミュニティツールを構築するため、前年度開発されたシステムを拡張し、ブックマーク機能とアノテーション機能を追加した。本研究開発の成果により、システム利用者間のコミュニケーションの向上が期待される。著者は、ブックマーク機能とアノテーション機能を実現するため、Web クライアントアプリケーションの URL の生成、コア画像の保存、Twitter との連携、メモの閲覧、添付、編集、削除などを開発した。ブックマーク機能の追加により、閲覧状態の再現はワンクリックで、簡単になった。また、個人が利用するだけでなく、この閲覧状態を他の人と共有することもできた。さらに、アノテーション機能の追加により、地質研究者のコミュニティを活性化させられることも考えられる。システムのブックマーク機能とアノテーション機能利用し、自分のコメントや他人のコメントは、簡単に共有できるようとなり、システム利用者間の学術コミュニケーションや議論を増加することにより、コアデータの解析が飛躍的な進歩が期待され、地質学での新たな発見を促進させることができる。

目次

第1章	はじめに	1
第2章	開発背景と解決すべき問題	3
2.1	統合海洋掘削計画とコア試料	3
2.2	DICOM	4
2.3	既存システム	4
2.4	解決すべき課題	6
2.5	議論	6
第3章	海底コア CT スキャンデータ可視化・情報共有を可能とするクラウドサービス	8
3.1	提案する機能	8
3.1.1	ブックマーク機能	8
3.1.2	アノテーション機能	8
3.2	システム構成	8
3.2.1	既存システムの構成	8
3.2.2	新規システムの構成	11
3.3	開発計画	12
3.3.1	顧客と開発チーム	12
3.3.2	開発すべき項目とスコープ	12
3.3.3	開発スケジュール	14
第4章	WEBクライアントアプリケーションの開発	15
4.1	開発項目と開発計画	15
4.1.1	開発項目	15
4.1.2	自分の担当部分の計画	17
4.2	UIの設計と実装	18
4.3	機能の設計と実装	30
4.3.1	ブックマーク機能	30
4.3.2	アノテーション機能	31
4.3.3	ユーザビリティの評価実験	34
第5章	おわりに	37
	謝辞	38
	参考文献	39
	付録 Webアプリケーションの関数定義	40

図目次

図 2.1 コア試料	3
図 2.2 Android 版	5
図 2.3 Web 版	6
図 3.1 既存システム構成図	9
図 3.2 変更したシステム構成図	11
図 3.3 開発スコープ	13
図 3.4 開発スケジュール	14
図 4.1 Web アプリケーション開発のスケジュール	18
図 4.2 ブックマーク機能の画面遷移	18
図 4.3 アノテーション機能の画面遷移図	19
図 4.4 ユーザ管理機能の画面遷移図	19
図 4.5 既存システム機能の改善機能の画面遷移図	19
図 4.6 新規システムのホームページ	21
図 4.7 コアデータ画像の生成	22
図 4.8 URL の生成	23
図 4.9 メモの添付-添付位置選択	23
図 4.10 メモの添付-添付位置確認	24
図 4.11 メモの添付-添付内容入力	24
図 4.12 Twitter との連携	25
図 4.13 ログイン画面	25
図 4.14 ヘルプページ画面	26
図 4.15 既存システムと拡張システムの比較図	27
図 4.16 コアデータ検索エリア	28
図 4.17 コアデータ分析エリア	28
図 4.18 UI 変更の例	29
図 4.19 サーバに初請求した後に生成したコア画像	30

表目次

表 3.1	開発言語一覧.....	10
表 3.2	レンダリングエンジン API の取りうる値.....	10
表 3.3	レンダリングエンジン API の戻り値.....	10
表 3.4	プロジェクト体制.....	12
表 3.5	役割分担.....	13
表 4.1	Web クライアントアプリケーションの開発項目.....	17
表 4.2	システム権限.....	26
表 4.3	レンダリングエンジン API の取りうる値-メモ閲覧.....	31
表 4.4	レンダリングエンジン API の戻り値-メモ閲覧.....	32
表 4.5	レンダリングエンジン API の取りうる値-メモ添付.....	32
表 4.6	レンダリングエンジン API の戻り値-メモ添付.....	32
表 4.7	レンダリングエンジン API の取りうる値-メモ編集.....	32
表 4.8	レンダリングエンジン API の戻り値-メモ編集.....	33
表 4.9	レンダリングエンジン API の取りうる値-メモ削除.....	33
表 4.10	レンダリングエンジン API の戻り値-メモ削除.....	33
表 4.11	実験タスク.....	34
表 4.12	実験タスクの結果.....	35
表 4.13	アンケート調査の内容と結果.....	35

第1章 はじめに

独立行政法人海洋研究開発機構 (JAMSTEC) の所有する地球深部探査船「ちきゅう」は、人類史上初めてマントルや巨大地震発生域への大深度掘削を可能にする世界初のライザー式科学掘削船である[1]。「ちきゅう」は、国際深海科学掘削計画 (IODP) [2]の主力船として世界で初めて、海底から 7,000 メートルを掘りぬいてマントルへと到達できる[3]。大陸の移動、大規模な火山活動、地球は誕生以来その姿を変え続けている。その原動力は固体でありながら流動する不思議な物質「マントル」だと言われている[4]。地球の中で起こっていることが地上にどう影響するのか調べる。「ちきゅう」は従来の深海掘削船には無かった、X線 CT スキャナを搭載している。掘削されたコア試料は、船上に搭載されている X線 CT スキャナにかけられ、DICOM フォーマットのデジタルデータとして保存される。デジタル化されたコア資料はコアデータと呼ばれ、「Virtual Core Library」というウェブサイトで公開される。現在、世界中の研究者は Web を介して、サンプルの採取要求とコアデータのダウンロードが可能である。しかし、コアデータを用いる分析には主に 2つの課題がある。

1. コアデータのファイルサイズが過大

ユーザは、コアデータを閲覧する際に、ダウンロードしなければならない。コアデータ 1つのファイルサイズは、約 1.5GB もあるので、インターネット通信に時間が掛かる。しかも、ダウンロードしたコアデータは、ユーザのクライアントにはほぞんしなければならないので、ユーザにとって、十分なストレージ容量が必要である。さらに、コアデータの可視化には、高性能コンピューターも必要である。

2. DICOM フォーマットのビューアはコアデータの閲覧に適していない

コアデータを閲覧するため、可視化しなければならない、DICOM フォーマットにより、生成したビューアは、人間の筋肉や臓器など医療用に特化されているので、コアデータの閲覧にとって、適するものでない。

上述の課題に対し、先行研究[5][6][7][8]では、高性能の画像処理能力を持つサーバを活用し、3次元の DICOM フォーマットのコアデータのビューアを、軽量な 2次元画像に変換する機能持つクラウドサービス「Virtual Core Viewer」が開発された。クラウドサービスのクライアントは、タブレット端末からの利用を想定した Android アプリケーションと、PCからの利用を想定した Web アプリケーションの 2種類がある。「Virtual Core Viewer」の開発により、普通の端末性能や通信帯域において、コアデータの閲覧することが可能になった。

しかし、「Virtual Core Viewer」を利用する際に、機能としては、不足しているところも発見した。まずは、既存システムには、コアデータ画像を再現する際に、コア画像が表示するまで、とても、複雑で細かい操作が必要である。しかも、次回に同じ画像を閲覧する際に、全く同じ操作をしないと、コア画像の再現することはできないので、効率が低下である。他のシステム利用者と、同じコア画像を共有する際にも、同じ操作を伝える必要がある。そして、システム利用者がコア画像を閲覧する際に、気になっているところが発見しても、コア画像の気になっているところに、直接メモやコメントなど記入することはできない。それらの問題を解決ため、本研究開発では、既存システムを拡張し、既存機能の上で、ブックマー

ク機能とアノテーション機能を含めるクラウドサービスを構築した。機能の追加により、システム利用者の操作情報を含める URL を生成する機能とコア画像にメモを添付する機能は可能となった。

著者は、このシステムの内、追加機能の Web クライアントアプリケーションの開発を担当した。ブックマーク機能とアノテーション機能の実現するため、Web クライアントで URL の生成、コア画像の保存、Twitter の連携、メモの閲覧、添付、編集、削除などの開発である。

本報告書は、全 5 章で構成されている。以降、第 2 章では、本研究の背景、まだ残っている課題、課題を解決するための対策についての議論を順に述べる。第 3 章では、課題を解決するための提案機能、既存システムと新規システムの構成、開発体制について述べる。第 4 章では筆者が担当した Web ブラウザ向けクライアントアプリケーションの開発について述べる。最後に、第 5 章では本プロジェクトのまとめと今後の発展について述べる。

第2章 開発背景と解決すべき問題

2.1 統合海洋掘削計画とコア試料

統合国際深海掘削計画（IODP）は日本と米国が主導する地球環境変動、地球内部構造及び地殻内生物圏の解明を目的とした国際的な海洋科学掘削計画である。海洋研究開発機構（JAMSTEC）が所有する地球深部探査船「ちきゅう」は、人類史上初めてマントルや巨大地震発生域への大深度掘削を可能にする世界初のライザー式科学掘削船である。「ちきゅう」は統合国際深海掘削計画（IODP）の主力船として地球探査を行っている。マントル掘削プロジェクトの科学目標は、海洋プレートの本質を明らかにすることである。マントル掘削により、約 100 年前に発見されたモホロビッチ不連続面（モホ面）の実体が明らかになり、さらにその直下の地表の変質を受けていないマントル物質が直接採取することにより、海洋プレートの本質ばかりでなく地球型惑星の進化研究を飛躍的に発展させることが期待される。「ちきゅう」には、図 2.1 に示すような掘削コア試料、と呼ばれる海底から掘削された柱上の地層サンプルを分析するための様々な設備が搭載されている。他の掘削船にはない「ちきゅう」の特徴の一つとして、船上には、X 線 CT スキャナが搭載している。海底から掘削してきたコア試料は、X 線 CT スキャナに掛け、デジタルデータ化することにより、コア試料の実物を破壊することなく、かつ物質や構造が変化する前にその内部を可視化することができる。デジタルデータ化された後のコア試料の実物は高知コアセンター[9]にて保管、管理される。デジタルデータ化されたコア試料は高知コアセンターの「Virtual Core Library」[10]にて公開されており、インターネットから自由に利用することができる。



図 2.1 コア試料

2.2 DICOM

コアデータは、DICOM と呼ばれるフォーマットで保存される。DICOM [11][12]とは、Digital Imaging and COmmunication in Medicine の略で、CT や MRI、CR など撮影した医用画像のフォーマットと、それらの画像を扱う医用画像機器間の通信プロトコルを定義した標準規格のことである。ACR（北米放射線学会）と NEMA（米国電機工業会）によって 1985 年に最初の規格 ACR-NEMA300-1985 が制定された。その後も進化を続け、1993 年に RSNA において承認された新しい規格が DICOM と名づけられ、現在に至っている。一般で用いられる jpeg や bmp フォーマットと違い、DICOM フォーマットは、同一ファイル内全てに、DICOM タグと呼ばれる形式で、患者名、ID 番号、生年月日、撮影日、撮影施設、部位を始めとする個人情報に関わるかなりのデータが含まれている。さらに、DICOM ビューアを閲覧する際に、DICOM フォーマットに対応した OsiriX[13]等ソフトが必要である。

2.3 既存システム

実際には、DICOM フォーマットのコアデータを利用し、コア試料を分析する際に、2 つ問題がある。

1. コアデータのファイルサイズが過大

ユーザは、コアデータを閲覧する際に、ダウンロードしなければならない。コアデータ 1 つのファイルサイズは、約 1.5GB もあるので、インターネット通信に時間が掛かる。しかも、ダウンロードしたコアデータは、ユーザのクライアントにほぞんしなければならないので、ユーザにとって、十分なストレージ容量が必要である。さらに、コアデータの可視化には、高性能コンピューターも必要である。

2. DICOM フォーマットのビューアはコアデータの閲覧に適していない

コアデータを閲覧するため、可視化しなければならない、DICOM フォーマットにより、生成したビューアは、人間の筋肉や臓器など医療用に特化されているので、コアデータの閲覧にとって、適するものでない。

上述の課題に対し、先行研究[5][6][7][8]では、高性能の画像処理能力を持つサーバを活用し、3次元の DICOM フォーマットのコアデータのビューアを、軽量の 2次元画像に変換する機能持つクラウドサービス「Virtual Core Viewer」が開発された。クラウドサービスのクライアントは、タブレット端末からの利用を想定した Android アプリケーションと、PC からの利用を想定した Web アプリケーションの 2種類がある。「Virtual Core Viewer」の開発により、普通の端末性能や通信帯域において、コアデータの閲覧することが可能になった。図 2.2 に示すのは、Android 版のコアデータを選択する画面の様子である。図 2.3 に示すのは、Web 版のコア画像が表示される画面の様子である。システムの機能としては、主に以下の 3つがある。

1. コア試料の画像操作

大量のコアデータの中で、閲覧したい1つコアデータを指定し、選択する機能である。現在公開されているコアセクションの数は4000も超えるので、閲覧したいコアデータの特定は難しい。そこで、「航海番号」、「掘削サイト番号」、「ホール番号」、「コア番号」、「ビットタイプ」、「セクション番号」を入力することで、ユニークなコアデータを選定できる。

2. コア試料の分析

コアデータを閲覧する際に、コア画像に対し、拡大、縮小、回転、移動など基本操作機能である。この機能により、コア画像に対し、各視点からの閲覧が可能となる。さらに、コア画像の内部までを閲覧するため、コア画像の縦方向から切断機能も提供されている。この機能により、コア画像の切断面を見ることは可能となる。

3. CT値に対する色づけ

コア試料に含まれる物質の違いにより、色付け機能である。CT値を利用し、物質の密度を表すことができる。コア画像を閲覧する際に、強調したいところがあれば、CT値の設定により、コア画像の色付けすることができる。コアデータには、水や筒といった分析に不要な物体が写りこんでいる。表示するCT値幅を設定することで、分析に関係のない物体を取り除くことができる。

「Virtual Core Viewer」の開発により、一般に普及している端末性能や通信帯域において、コア試料の閲覧などのコアに対する操作や、CT値に対する自由な色付けが可能になった。

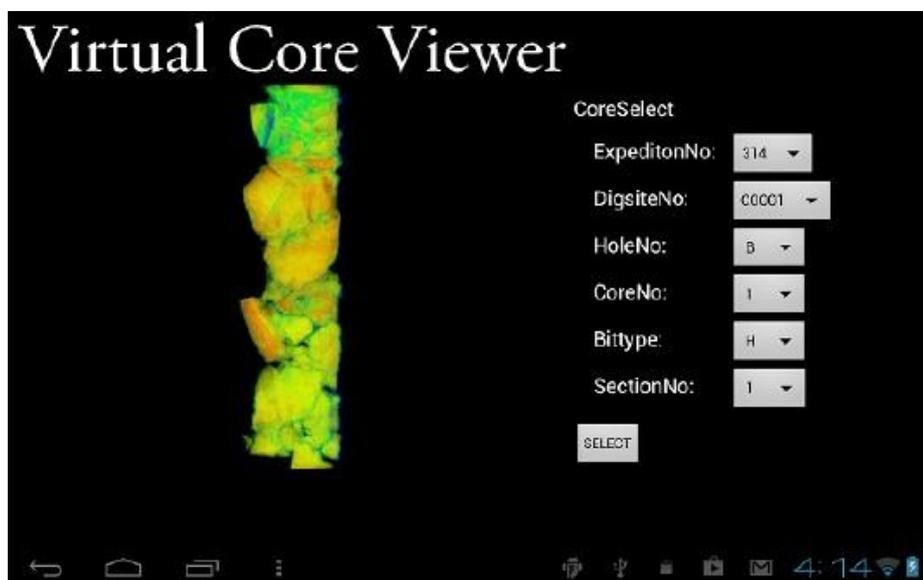


図 2.2 Android 版

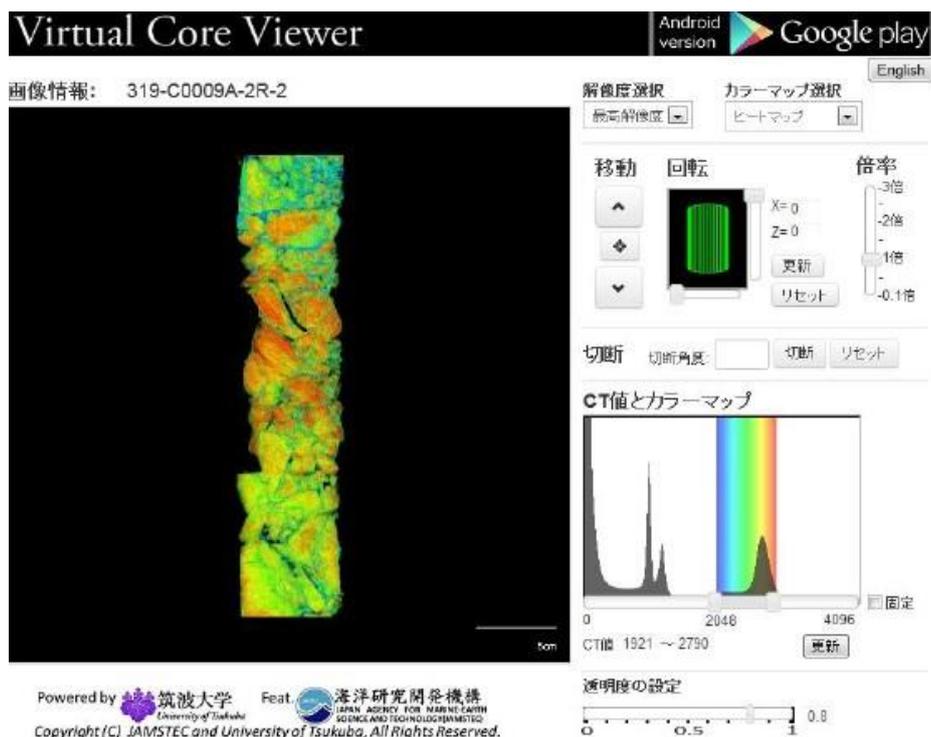


図 2.3 Web 版

2.4 解決すべき課題

しかし、既存システム「Virtual Core Viewer」を利用する際に、以下の2つ問題がある。

1. コアデータ画像を閲覧する際に、コア画像が表示されるまで、コアデータを選択し、回転、拡大縮小、CT 値で色付けなど、十数種類パラメータを設定するととても複雑で細かい操作が必要である。そして、同じなコア画像を再度に閲覧する際に、前回と同じな操作が行わないと、コアデータ画像の再現することができないので、手間がかかることだけではなく、コアデータの回転、拡大縮小、CT 値で色付けなど十数種類パラメータも覚えなければならない。さらに、閲覧しているコア画像を他人と共有する際に、同じなコア画像を表示するため、コアデータを選択し、回転、拡大縮小、CT 値で色付けなど、十数種類パラメータも共有しなければならないので、手間もかかる。
2. システム利用者がコア画像を閲覧する際に、コア画像を閲覧することにより、個人知見をシステムに保存することも考えられる。しかし、既存システムには、直接にメモやコメントなど記入することはできない。さらに、システム利用者の間、同じなコアデータについての見地や意見を交流する際に、他の通信手段を利用しなければならないので、システム利用者として、効率は悪い。システムとして、使用性は低下である。

2.5 議論

先行研究で、開発した「Virtual Core Viewer」により、コアデータのファイルサイズが大

きいとコアデータを閲覧するための専用ソフトウェアが存在しないという問題が解決され、コア画像は簡単に閲覧できるようとなったが、上述のように、コア画像を再現する際に、効率な低下とコア画像にメモを添付できないという2つ課題まだ残されている。これらの問題を解決ため、2つ機能を提案した。

1. ブックマーク機能

コア画像の再現するため、表示されるコア画像の情報を記録し、操作情報を含めるURLを作成する機能である。コアデータを選択し、回転、拡大縮小、CT値で色付けなど、十数種類パラメータをURLに保存することにより、操作時間を短縮し、コア画像を効率的に再現が可能となる。

2. アノテーション機能

コア画像に、システム利用者の知見やコメントなど情報を入力し、コア画像に保存する機能である。システム利用者がコア画像に添付した知見やコメントは、システム利用者の間に共有することも可能となる。

提案する機能の実現により、上述に、挙げられた2つ課題は解決されることが可能となる。ブックマーク機能の追加により、閲覧状態の再現はワンクリックで、実現できる。また、個人が利用するだけでなく、この閲覧状態を他の人と共有することもできる。さらに、アノテーション機能の追加により、地質研究者のコミュニティを活性化させられることも考えられる。システムのブックマーク機能とアノテーション機能利用し、自分の知見やコメントと他人知見やコメントは、簡単に共有できるようになり、システム利用者間の学術コミュニケーションや議論を増加することにより、コアデータの解析が飛躍的な進歩が期待され、地質学での新たな発見を促進させることができる。

ブックマーク機能とアノテーション機能は、既存システム「Virtual Core Viewer」を基づく追加機能である。既存システム「Virtual Core Viewer」のクライアントとなる、Webブラウザ版のWebクライアントアプリケーションとAndroidタブレット版のAndroidクライアントアプリケーションを拡張しなければならないので、本プロジェクトでは、著者がWebクライアントアプリケーションの開発を担当した。具体的の開発すべき項目、開発計画、実装した内容など、次章から説明する。

第3章 海底コア CT スキャンデータ可視化・情報共有を可能とするクラウドサービス

3.1 提案する機能

3.1.1 ブックマーク機能

ブックマーク機能は、閲覧しているコア画像に対する操作情報を保存し、コア画像を再現する機能である。コアデータを閲覧する際に、適切な閲覧効果を得るため、コアデータに対し、拡大縮小、回転、切断など基本操作と CT 値による色付け、透明度の設定など複雑で細かい設定が必要である。さらに、この閲覧状態を再現する際に、コアデータに対し、全く同じ操作が必要なので、システム利用者にとって、無駄な時間がかかる。非効率的なことである。ブックマーク機能の追加することにより、コアデータに対する操作情報を保存し、同じ閲覧状態の再現する URL を作成することが可能となる。PC 上の Web ブラウザまたは「Virtual Core Viewer」がインストールされている Android 端末からその URL を開くと、各クライアントにて同じコア画像が再現される。さらに、URL の生成する機能を利用し、Twitter との連携し、ワンクリックで Twitter のタイムラインに投稿できる。

3.1.2 アノテーション機能

アノテーション機能は、コアデータに対して、システム利用者の知見やコメントは、メモの形で、コア画像の指定された位置に添付する機能である。システム利用者は、コアデータを閲覧中に気になる部分があった場合、コアデータ画像の気になる位置をクリックすることにより、その位置に、システム利用者の知見やコメントを含めるメモを添付することができる。次回で、システムを利用する際に、自分が添付したものに対し、振り返ることができる。さらに、このメモは個人だけでなくすべてのシステム利用者が閲覧することが可能である。

3.2 システム構成

3.2.1 既存システムの構成

最初に、今回拡張を行った既存システムについて説明する。既存システムは(1)クライアントアプリケーション、(2)ゲートウェイ、(3)レンダリングエンジンという 3 つの部分からなっている。その構成を図 3.1 に示す。

クライアントアプリケーションは、Web ブラウザから利用できる Web アプリケーションと Android 端末から利用できる Android アプリケーションの 2 つ部分から構成される。また、既存システムを利用する際に、コア画像を閲覧するユーザインタフェースの機能も備える。次に、ゲートウェイでは、サーバの API が定義され、クライアントアプリケーションからのリクエストを受け、DB からコアデータを検索し、DICOM フォーマットのコアデータ

を2次元描画に変換するために、レンダリングエンジンにコアデータの描画リクエストを送信する。最後に、レンダリングエンジンでは、ゲートウェイから送られてきたリクエストに応じ、コアデータの描画というビジネスロジック計算を行う。描画したデータの結果は、クライアントアプリケーションに送信される。このような構成で、コアデータの描画に必要な重い計算は、すべて、ゲートウェイとレンダリングエンジンで構成されたサーバで行うので、クライアントにとって、高性能なPCがなくても、コアデータの閲覧が可能になる。その結果として、タブレットやPCのブラウザを通じて、簡単にコアデータを閲覧できるようになった。

システムのハードウェア構成を以下に説明する。ゲートウェイはCPU(Corei79302.8GHZ)、メモリ12GBで構成され、レンダリングエンジンはCPU(Intel Xeon E5645 2.40GHz(6cores))×2、GPU (Tesla M2050 1.55GHz) × 2、メモリ12GBで構成されたノードが計16ノードで構成されている。

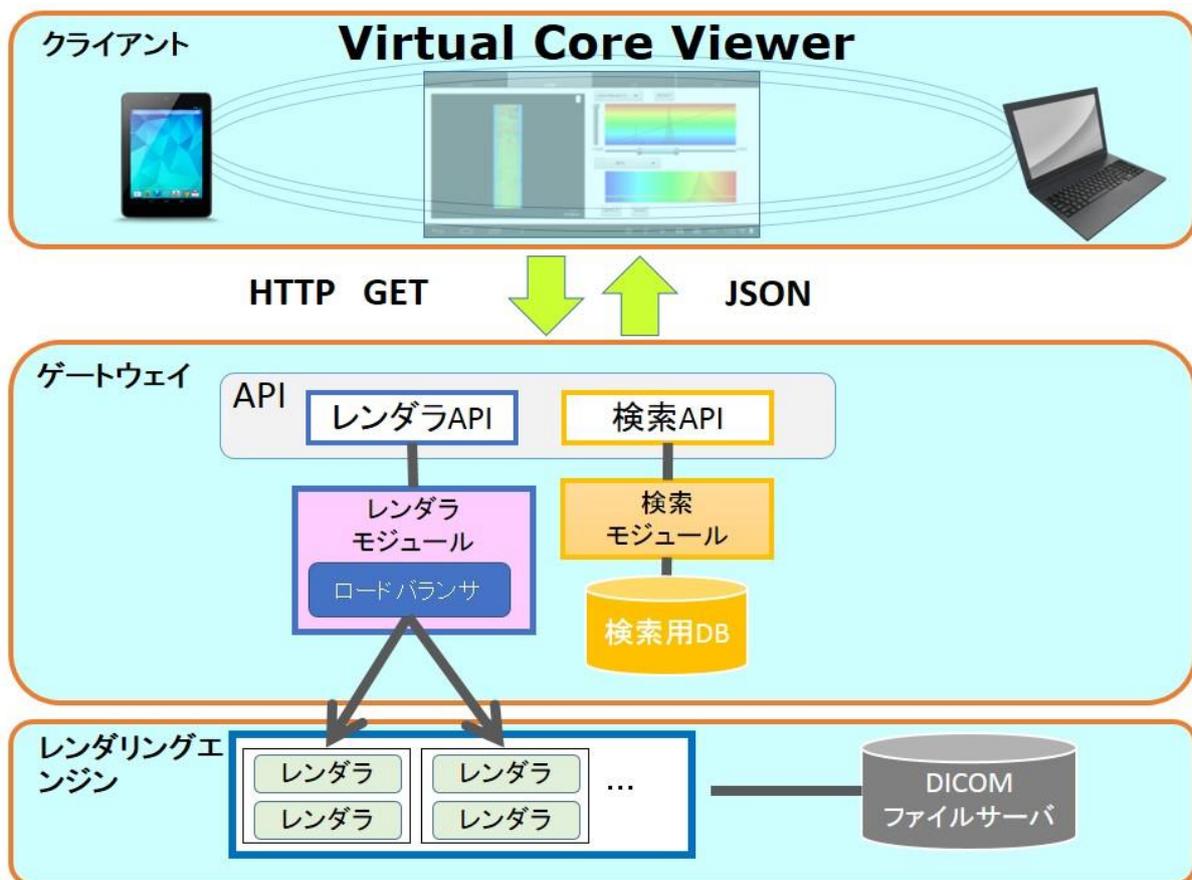


図 3.1 既存システム構成図

クライアント、ゲートウェイ、レンダリングエンジンの機能を実現する際に、表 3.1 に示す開発言語は、使っている。

表 3.1 開発言語一覧

構成		言語
クライアント	Web クライアント	JAVASCRIPT、HTML、CSS、PHP
	Android クライアント	JAVA、XML
ゲートウェイ		PHP
レンダリングエンジン		C++、

ゲートウェイで定義された API について説明する。既存システムのクライアントは、Web ブラウザと Android タブレット 2 種類がある。システムの汎用性を高めるため、現在によくある Web サービスに利用され、Web API が定義された。クライアントから、HTTP と呼ばれる通信規約を利用し、ゲートウェイに定義された API に従い、送信すれば、誰でも、何処でも、コア画像を閲覧するサービスを利用できる。通信データの形式は、現在良く利用される JSON と呼ばれるテキストフォーマットを使っている。API の定義は、表 3.2 と表 3.3 に示す。

表 3.2 レンダリングエンジン API の取りうる値

パラメータ	概要
corenum	コア番号
zoom	拡大・縮小の値
cut	切断角度
resolution	解像度
rot_type	回転方式
rotx、roty、rotz	x、y、z 方向の回転角度
depth	コアの上下移動
color_type	色付け方法
cf	CT 値と色の対応情報を含む配列
af	CT 値と透過度の対応情報を含む配列
eye_x、eye_y、eye_z	視点の座標
up_x、up_y、up_z	視点の上方向ベクトル
movex、movey	マウスの移動ベクトル
degree	回転角度
ctmin	色付けする最小の CT 値
ctmax	色付けする最大の CT 値

表 3.3 レンダリングエンジン API の返り値

パラメータ	概要
img_url	生成された画像の URL
histogram	CT 値のヒストグラムの配列
scale	1 ピクセルあたりの長さ
eye_x、eye_y、eye_z	現在のカメラ座標
up_x、up_y	現在のカメラの上方向ベクトル

3.2.2 新規システムの構成

本年度の研究開発で、ブックマーク機能とアノテーション機能を実現するため、既存システムの構成を基づき、拡張した。変更したシステム構成は、次に、図 3.2 に示すように、(1)クライアントアプリケーションを修正・追加した。(2)ゲートウェイに、新規のアノテーションAPI、モジュール、DBを追加した。

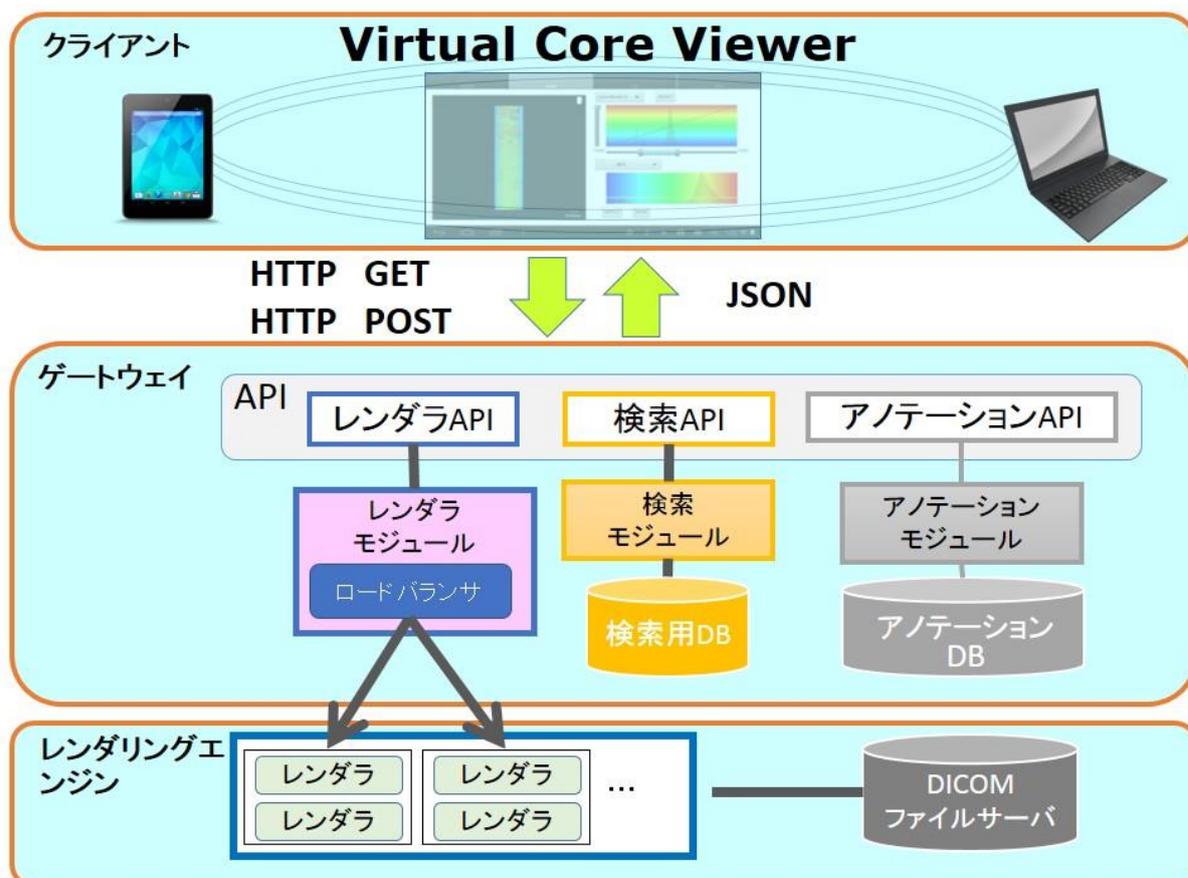


図 3.2 変更したシステム構成図

次に、変更する理由について、述べる。

1. クライアントアプリケーション変更

ブックマーク機能とアノテーション機能を実現するため、クライアントには、その機能を利用するボタンやテキストエリアなどを追加しなければならないので、UIを変更しなければならない。新規システムのユーザビリティを考えた上で、既存システムのUIも変更した。さらに、ブックマーク機能とアノテーション機能は、新規機能である。これらを実現するため、ゲートウェイとの通信や、クライアント側のビジネスロジック処理も必要なので、以上な理由で、WebクライアントとAndroidクライアントアプリケーションについて、変更作業をおこなっていた

2. ゲートウェイの変更

クライアントアプリケーションの変更することにより、ゲートウェイとクライアントアプリケーションの API を追加しなければならない。さらに、アノテーション機能を実現するため、新規システムに、添付したメモを保存するデータベースも必要となる。以上な理由で、ゲートウェイについても、変更した。

本プロジェクトでは、著者の担当部分は、Web クライアントアプリケーションである。上述の変更点について、Web クライアントアプリケーションの変更作業を中止とし、取り組んでいた。

3.3 開発計画

3.3.1 顧客と開発チーム

本研究開発の顧客は、(独)海洋研究開発機構(JAMSTEC)高知コア研究所の久光敏夫様(以下、久光様)である。久光様は主にコアの管理業務を担当している。コアの分析は、地球環境の変動のメカニズムやプロセスの解明に利用されている。本研究開発の実施体制を表 3.4 に示す。共同研究者、課題担当教員の指導の下、著者を含む開発メンバ 5 名で開発を行った。

表 3.4 プロジェクト体制

役割	所属	名前
共同研究者	海洋研究開発機 (JAMSTEC) 高知コア研究所	久光 敏夫 技術副主幹
課題担当教員	筑波大学大学院 システム情報工学 コンピューターサイエンス専攻	和田 耕一 教授 山際 伸一 准教授
開発メンバ	筑波大学大学院 システム情報工学 コンピューターサイエンス専攻	豊田 恭子 伊藤 弘貴 黒田 祐登 城崎 亮 高 達生

3.3.2 開発すべき項目とスコープ

提案したブックマーク機能とアノテーション機能の実現のため、以下の項目を実現する必要がある。

1. ブックマーク機能

システム利用が閲覧しているコア画像を再現するため、閲覧する際に、コアデータの選択、コア画像に対する回転や拡大縮小など操作を含める URL を生成する機能が必要となる。さらに、この情報を、Web ブラウザやタブレットで簡単に、他人と共有するため、Twitter との連携する機能も必要となる。

2. アノテーション機能

コア画像にシステム利用者のメモやコメントなどを保存し、システム利用者の中で、簡単にコミュニケーションや議論を行えるシステムを構築するため、Web ブラウザやタブレットを通じ、サーバにシステム利用者のメモやコメント情報を保存する機能が必要となる。そうすると、添付したメモやコメントに対する編集、削除機能も不可欠となる。

3. ユーザ管理機能

上記2のアノテーション機能を実現するにより、システム利用者は、サーバにメモやコメントを残すこともできる。そうすると、メモやコメントの悪意がある削除や改竄と無意味なメモやコメントを大量につけることも考えられる。それを防止するため、ユーザ管理機能が必要となる。システム利用者のシステム利用範囲は、権限により、限定されている。

4. ユーザビリティの向上

ブックマーク機能とアノテーション機能の追加により、システムの UI も変更した。さらに、既存システムの利用により、顧客様からのフィードバックや現状のシステムについての改善したいところもある。それも考えて、開発していく必要がある。

本プロジェクトの開発期間は2013年6月から2014年1月までであり、開発内容をスコープで分け、反復型開発[14]を行っていた。我々は開発内容を図3.3に示すように、S0、S1、S2、S3と大きく4つのスコープに分けて開発を行っていた。S0を最初のスコープとし、バグフィックスとユーザフィードバックより機能追加や修正を行っていた。S1とS2の段階で、3次元コアデータのビューに対してブックマーク作成及び3次元コアデータに対してアノテーション付加を行っていた。最後に、S3の段階で、Twitter連携、S1、S2に対するフィードバック反映、公開準備等を行っていた。各スコープで、久光技術主任を通じて地質学者の方々に利用していただき、本システムのフィードバックを得た。そのフィードバックを基にユーザインタフェースの改良や機能追加を重ねた。

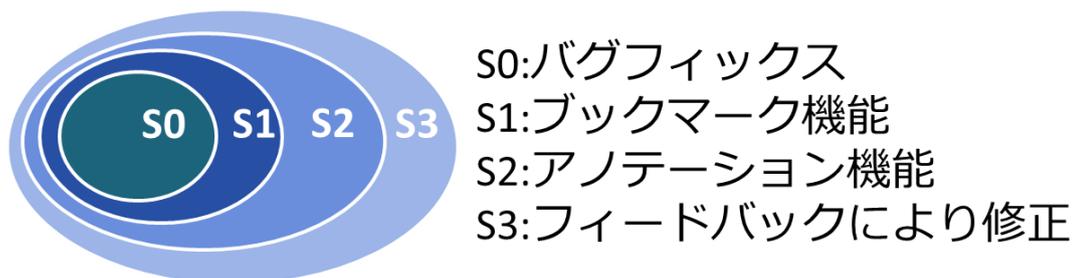


図 3.3 開発スコープ

本プロジェクトの役割分担は、表 3.5 に示す。

表 3.5 役割分担

担当者	役割	担当範囲
豊田 恭子	リーダー	Android 版クライアントアプリケーション
伊藤 弘貴		Web 版テスト

黒田 祐登		ゲートウェイ (サーバ)
城崎 亮		ゲートウェイ (サーバ)、レンダリングエンジン、 Android 版テスト
高 達生		Web 版クライアントアプリケーション

3.3.3 開発スケジュール

本プロジェクトの開発計画は、図 3.4 に示すような進む予定で実施した。

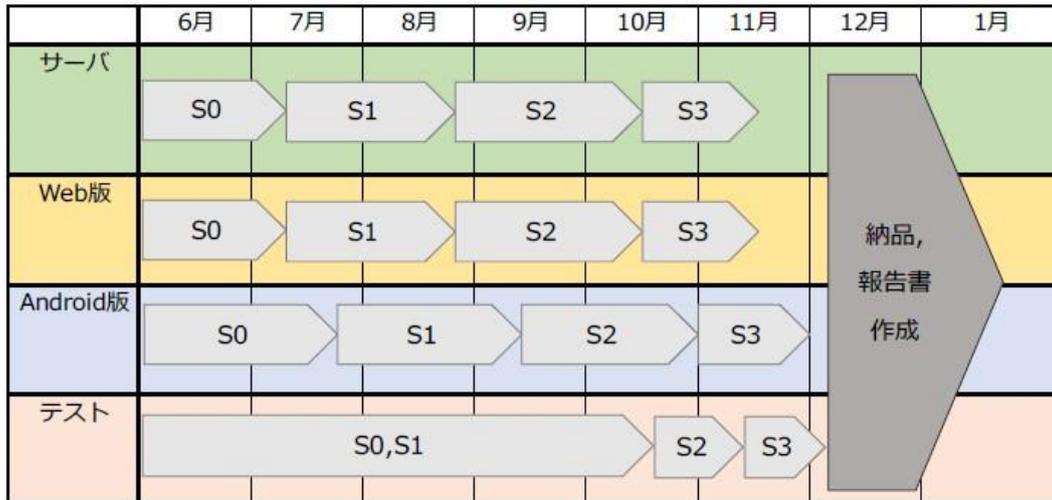


図 3.4 開発スケジュール

第4章 WEB クライアントアプリケーションの開発

本章では、本プロジェクトにおいて、筆者が担当した Web クライアントアプリケーションの設計・実装について説明する。

4.1 開発項目と開発計画

4.1.1 開発項目

著者は担当した部分は、Web クライアントアプリケーションのブックマーク機能とアノテーション機能の実装である。この 2 つ機能を実現するため、開発項目としては、UI の設計と実装と、Web クライアントアプリケーション機能の設計と実装である 2 部分に分けられる。さらに、この 2 つ部分において、WBS で細かいタスクを分解し、合計 15 個サブタスクを作成した。次に、開発項目について述べる。

Web クライアントアプリケーション UI の設計と実装

1. コア画像表示エリアの設計と実装

このエリアは、コア画像とコア画像に追加されたマーカーが表示されるエリアである。コア画像にメモを添付する際に、添付した位置を表明するため、マーカーは、コア画像に表示させる必要がある。また、マーカーのサイズを設計必要がある。

2. メモ表示エリアの設計と実装

このエリアは、システム利用者が添付したメモをまとめ、メモリストの形式で、添付したメモを表示するエリアである。まとめられたメモのサイズ、表示文字数の制限、文字のフロント、番号の設計する必要がある。

3. コメントエリアの設計と実装

このエリアは、システム利用者が添付したメモの詳細内容を表示するエリアである。表示項目としては、メモを記入したシステム利用者の名前とコメント内容 2 部分で構成される。記入したメモの内容編集と削除の操作もここで行う。メモ内容の文字表示仕様、フロント、記入者の名前のフォーマット、編集・削除ボタンの仕様、コメント欄の仕様などの設計する必要がある。

4. 操作エリアの設計と実装

このエリアでは、コア画像の保存、操作情報を含める URL の生成、生成した URL の Twitter への投稿、アノテーション機能のメモ添付、コア画像上に表示しているマーカーの表示と非表示の制御と呼ばれる操作を実行できる。URL 生成ボタン、Twitter との連携ボタン、Switch ボタン、メモ添付ボタンの仕様と配置の設計する必要がある。

5. ログインエリアの設計と実装

ユーザごとにシステム利用する権限つけるために、ユーザ管理機能を追加した。ユーザ管理機能とは、ユーザの登録、システムへのログイン、ログアウトする機能である。Web クライアントアプリケーションでは、ログイン、ログアウト機能を利用する際に、UI 設計が必要となる。

6. ヘルプページの作成

既存システムの機能も含め、新規システムの操作を説明する Web ページである。説明文は英語である。ヘルプページにリンクするボタンの仕様とヘルプページの仕様の設計する必要がある。

7. UI の改善

新規システムのユーザビリティを向上するため、既存システムの UI 設計の改善と顧客のヒアリングから得られた要望対し、以下の作業を行っていた。

- ① 位置のスケールを表示する。
- ② Web クライアントアプリケーションを起動する際に、CT 値をデフォルト値は 0 にする。
- ③ コアデータに対する操作は、リセットできるようにする。

Web クライアントアプリケーションの JAVASCRIPT 機能の設計と実装

1. URL の生成機能

Web ブラウザから、表示しているコア画像の回転、拡大縮小、CT 値による色付けなど操作情報を抽出し、コア画像を再現できる操作情報を含める URL を生成する機能である。

2. 画像の保存機能

Web ブラウザから、表示しているコア画像をローカルに保存する機能である。

3. Twitter との連携機能

Twitter の API を利用し、ワンクリックで、生成した URL を Twitter のタイムラインに投稿する機能である。

4. メモの表示と非表示の機能

メモの添付することにより、コア画像の閲覧に邪魔する時も考えられる。システム利用者の都合で、添付したメモを表示と非表示の制御する機能である。

5. メモリストの表示機能

添付したメモ詳細内容を短縮し、頭文字 10 文字だけメモリストで表示する機能である。

6. コメントの表示する機能

メモリストやコア画像エリアで、選択したメモの詳細内容をコメントエリアで表示させる機能である。

7. ログインエリアの機能

システムにログインする機能である。システム利用者にとって、ログインしているかどうか、所有している権限が違う。

8. メモの添付、編集、削除の機能

ゲートウェイの API に従い、Web クライアントで、メモの添付、編集、削除に使う JSON データを作成し、そのデータを HTTP の POST・GET メソッドを利用し、ゲートウェイに送る機能とゲートウェイから、返してきた結果を Web クライアントに表示させる機能である。

開発の進み方としては、まず、顧客様へヒアリングを行い、要件が決められ、その要件を反映した設計したプロトタイプやモックアップを作成し、顧客様に確認する。その後、設計、実装を行う。各スコープの開発が終了した時点で、段階的な成果物を顧客様のユーザテストをしてもらう。顧客からのフィードバックや修正点は、次のスコープで、反映させる。顧客からのフィードバックや要望を次のスコープで反映させるというような反復型の開発手法で進行していた。

4.1.2 自分の担当部分の計画

表 4.1 Web クライアントアプリケーションの開発項目

スコープ	開発内容	所属項目
S2	コア画像表示エリアの設計と実装	アノテーション機能
S3	ヘルプページの作成	フィードバック
S2	メモ表示エリアの設計と実装	アノテーション機能
S2	コメントエリアの設計と実装	アノテーション機能
S1	操作エリアの設計と実装	ブックマーク機能
S1	ログインエリアの設計と実装	アノテーション機能
S0,S1,S2,S3	UI の改善	フィードバック
S1	URL の生成機能	ブックマーク機能
S1	Twitter との連携機能	ブックマーク機能
S2	メモの表示と非表示の機能	アノテーション機能
S2	メモリストの表示機能	アノテーション機能
S2	コメントの表示する機能	アノテーション機能
S1	ログインエリアの機能	アノテーション機能
S2	メモの添付、編集、削除の機能	アノテーション機能
S0,S1,S2,S3	バグ修正	安定化

Web クライアントアプリケーションのスコープは表 4.1 に示す。Web クライアントアプリケーションの開発計画は、図 4.1 に示す、S0 では、既存システムのバグ修正、顧客からのフィードバックの対応を行っていた。さらに、既存システムの構成や機能などを理解ため、関連研究の論文や開発に使う技術の準備作業も行っていた。S1 では、ブックマーク機能の開

発と S0 に対し、顧客からのフィードバックの反映を行っていた。S2 では、アノテーション機能の開発と S1 に対し、顧客からのフィードバックの反映を行っていた。最後の S3 で、S2 に対し、顧客からのフィードの対応と納品準備を行っていた。ユーザテストでは、開発者は、シナリオやテスト仕様書を設定せず、顧客は自由に操作できる。

	6月	7月	8月	9月	10月	11月	12月	1月
予定	S0		S1		S2		S3	報告書作成
実績	S0		S1		S2		S3	報告書作成
ユーザテスト	7/29 S0		9/13 S1		11/2 S2		12/3 S3	

図 4.1 Web アプリケーション開発のスケジュール

4.2 UI の設計と実装

本年度の研究開発においてお客様の要件に従い、システムの拡張機能とし、ブックマーク機能やアノテーション機能など新機能を追加することになった。新機能の追加することにより、UI も変更しなければならないこととなった。さらに、既存システムの Web クライアントには、ホームページ以外の画面がないので、ブックマーク機能を実現するため、ホームページの UI を変更した上で、URL 生成画面、Twitter との連携画面を追加した。追加した後の画面遷移図は、図 4.2 に示す。

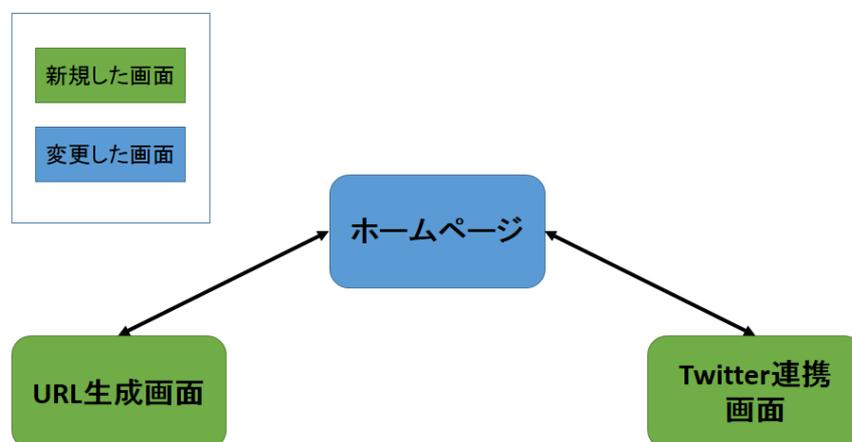


図 4.2 ブックマーク機能の画面遷移

アノテーション機能の実現するため、メモ添付ダイアログ、メモ削除ダイアログ、メモ編集ダイアログ、添付操作確認ダイアログ、削除操作確認ダイアログ、編集操作確認ダイアログ 9 つを追加した。追加した後の画面遷移図は、図 4.3 に示す。

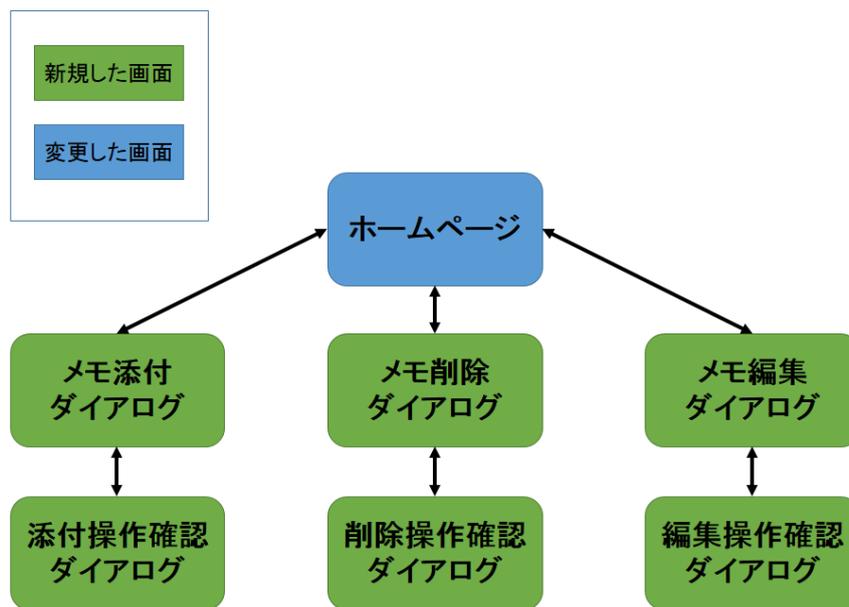


図 4.3 アノテーション機能の画面遷移図

ユーザ管理機能の実現するため、ユーザ登録画面、ログイン画面を追加した。追加した後の画面遷移図は、図 4.4 に示す。

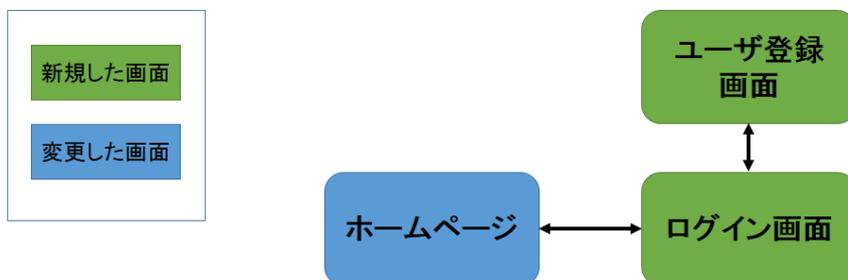


図 4.4 ユーザ管理機能の画面遷移図

既存システム機能の改善のため、ヘルプ画面、コア画像保存画面を追加した。追加した後の画面遷移図は、図 4.5 に示す。

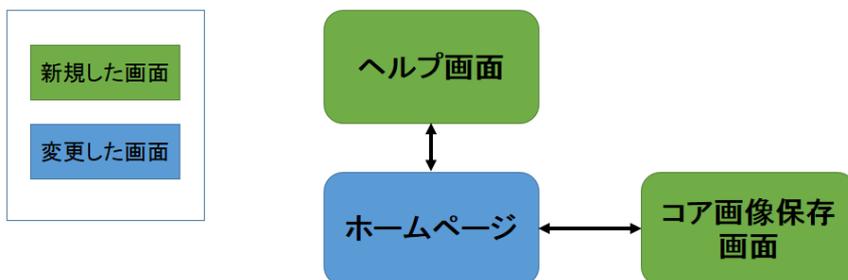


図 4.5 既存システム機能の改善機能の画面遷移図

さらに、ブックマーク機能とアノテーション機能の実現だけではなく、システム自身のユ

ユーザビリティも考えなければならぬので、次に、ユーザビリティ 5 つの質的な構成要素を考えながら、設計と実装を行っていた。

1. 学習しやすさ

すぐに、そして簡単に使用することが可能

2. 効率性

学習後は高い生産性を創出可能

3. 記憶しやすさ

簡単に使い方を記憶することが可能

4. 間違いにくさ

間違いを起しにくく、また起こしても簡単に回復可能

5. 主観的満足度

ユーザが満足できるよう楽しく利用することが可能

そのため、新規システムを設計・実装する際に、2 つ原則を決めた。1 つは、新規機能の追加は、できる限り、既存システムに、利用していないエリアを活用する、もう 1 つは、システム利用者に対し、慣れてきた使用習慣を変えないようにした。その結果は、変更した UI は、図 4.3 に示すように、赤枠で囲まれ、新たなエリア①、②、③、④、4 つを追加された。本報告での説明する際に、①エリアは、メモリスト表示エリアと呼び、②エリアは、機能操作エリアと呼び、③エリアは、ログインエリアと呼び、④エリアは、メモ内容表示するエリアと呼ぶこととなる。

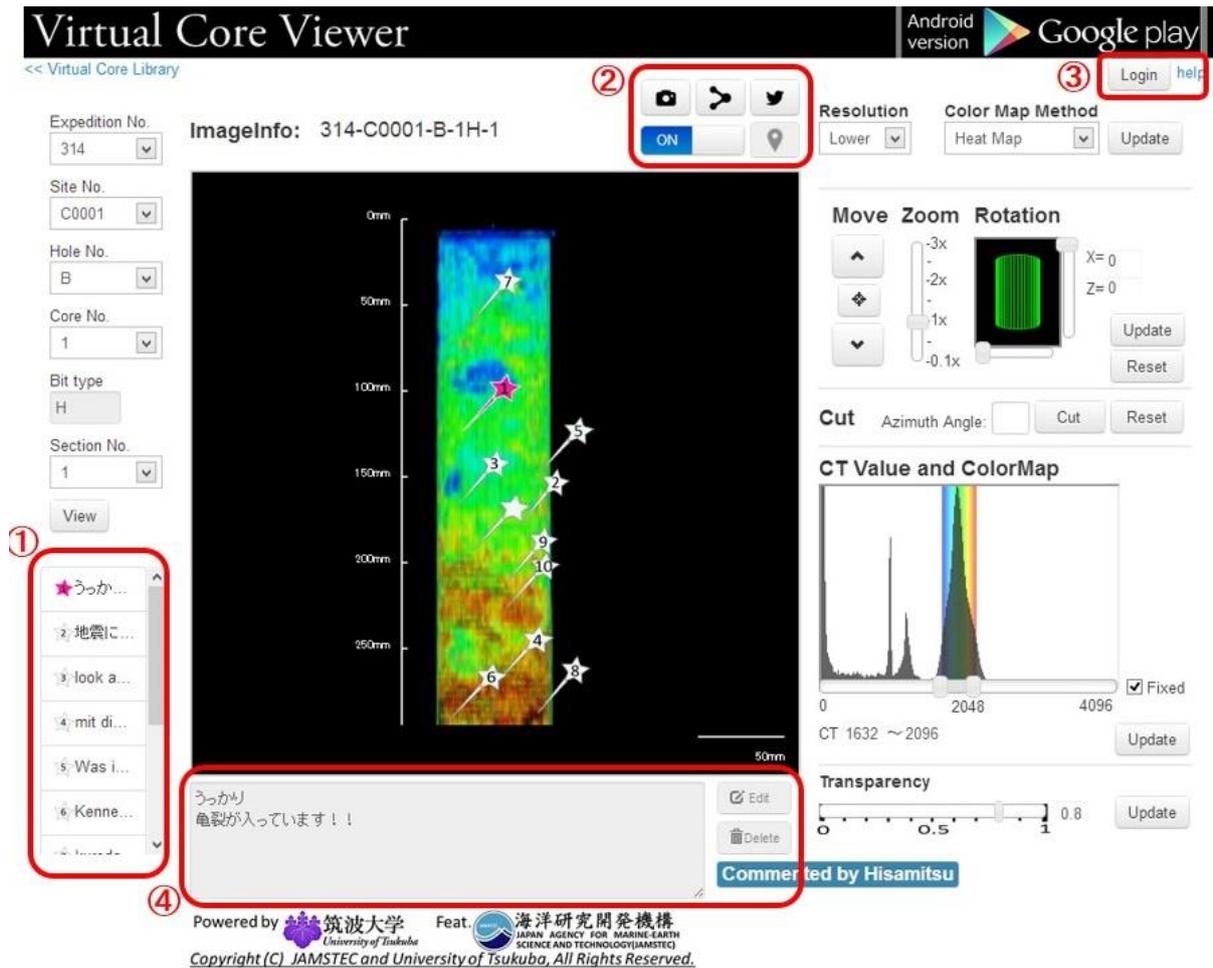


図 4.6 新規システムのホームページ

次に、各エリアの仕様とこのエリアで実行できる機能について、説明する。

1. メモリスト表示エリア

ホームページの中心に表示されてるコアデータ画像に、アノテーションを添付する際に、コメントが記入できる。このエリアで、記入された内容を 10 文字までの形式で、略して表示されてる。検索機能を提供していないので、その機能の代わりとしても使える。このエリアで選定されたメモは、中心におけるコアデータが表示するエリアに添付された対応しているマーカーもハイライトで表示される。さらに、4.メモ内容表示エリアにも、メモを記入した際に、詳細な内容を表示される。実際に利用するイメージは、図 4.3 に示すようになっている。

2. 機能操作エリア

このエリアには、コアデータ画像の保存、閲覧している画像の URL の生成、生成した URL の共有、メモの添付、添付したメモの表示と非表示の制御のボタンが 5 つがある。各ボタンに対する操作することにより、コアデータごとの、メモ添付、表示制御、URL 共有など機能を実現できる。これから、幾つの実際に利用するイメージに沿って述べる。

1) コアデータ画像の生成機能

カメラのようなアイコンが付いているボタンを押すと、以下のようなコアデータが画像が表示される新しいタブは、ブラウザから開かれ、この画像をローカルに保存することができる。図 4.4 に示す。コア試料のどの部分を読覧しているのは、分かりにくいため、コア画像を閲覧の画面に、スケールが設置されている。画面上のコア画像と実物のコア試料の比率が表されている。

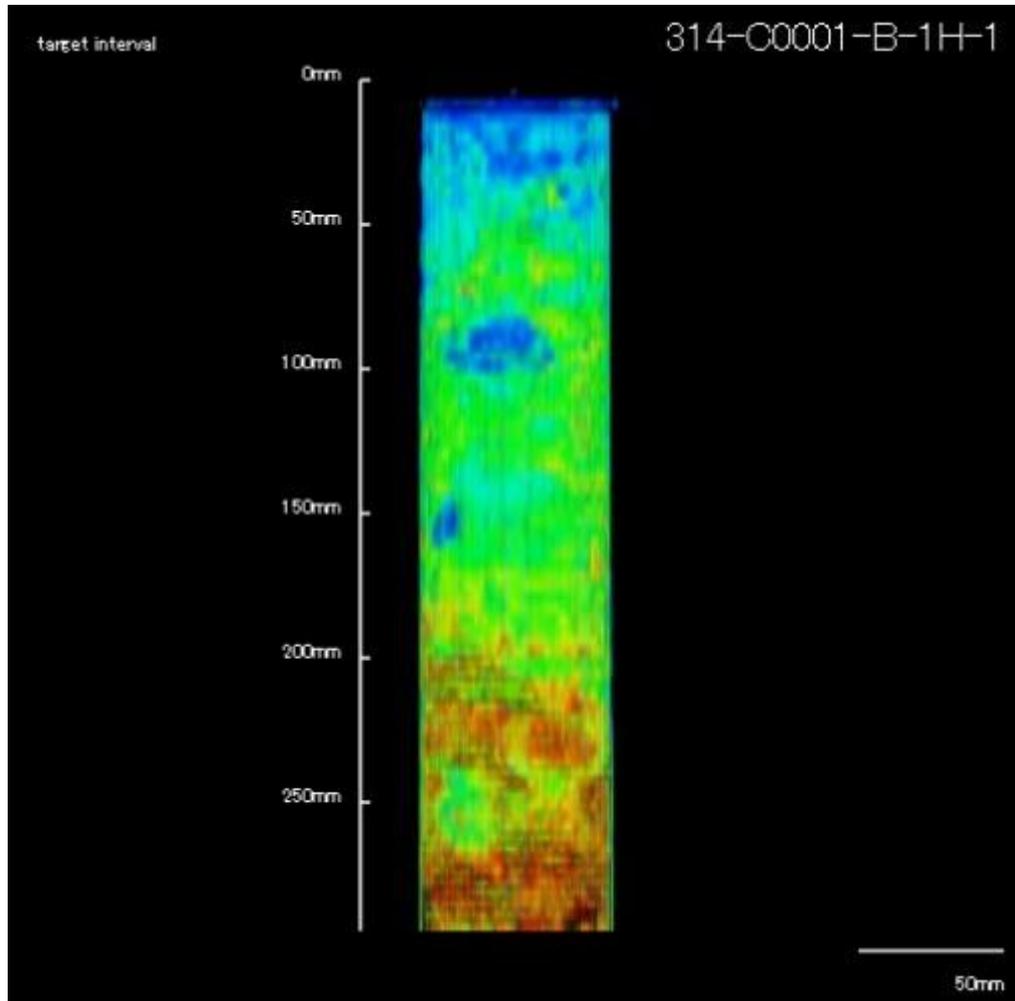


図 4.7 コアデータ画像の生成

2) URL の生成機能

操作エリアの上の3つボタンの間中のボタンを押すと、現在に表示されているコアデータ画像の情報を含める URL を生成し、この URL を利用し、現在が閲覧しているコアデータ画像の再現することができる。4.5 図に示す。

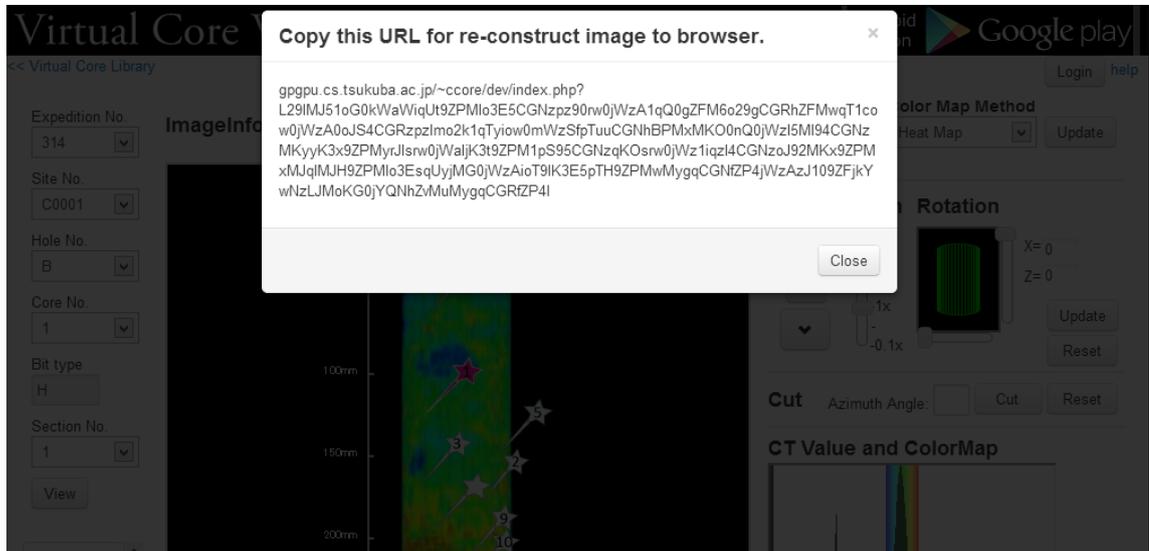


図 4.8 URL の生成

3) メモの添付機能

操作エリアには、マーカーのようなイメージが付いているボタンを押すと、コアデータ画像の位置を示すマーカーが出てき、添付する位置は、左クリックで確定する。そして、コメントが入力できるポップアップが表示され、ダイアログにコメントを入力すると、コメントはシステムに添付される。図 4.6、図 4.7、図 4.8 に示す。

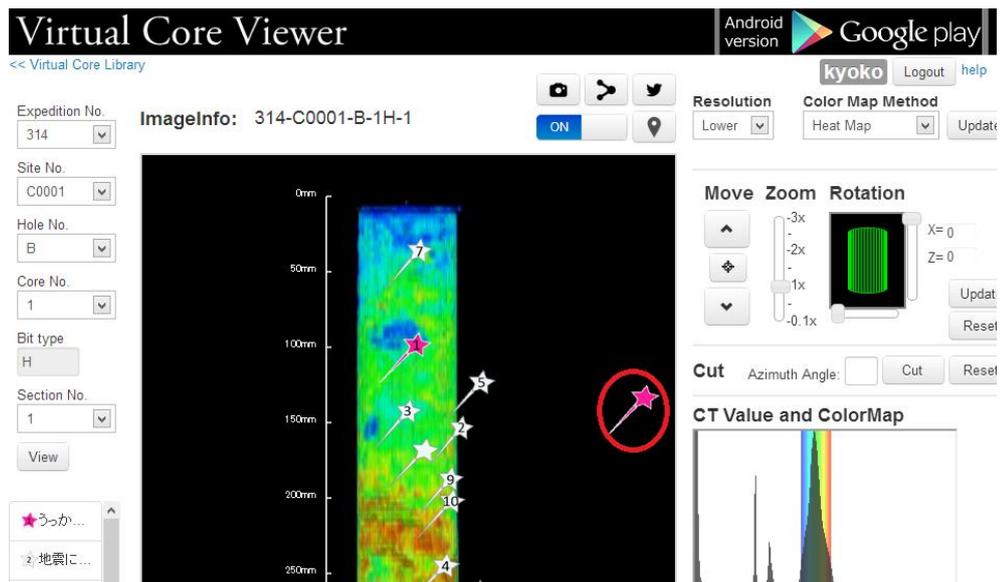


図 4.9 メモの添付・添付位置選択

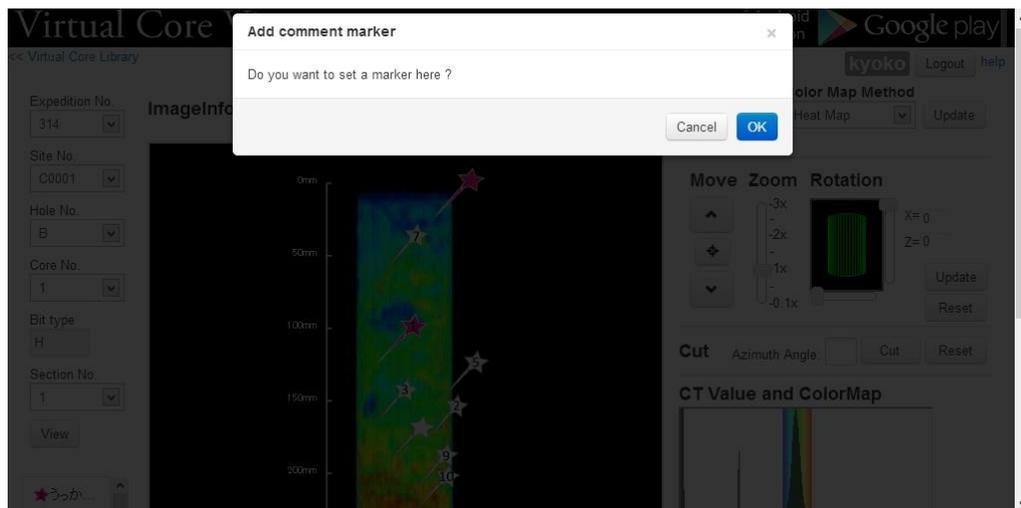


図 4.10 メモの添付-添付位置確認

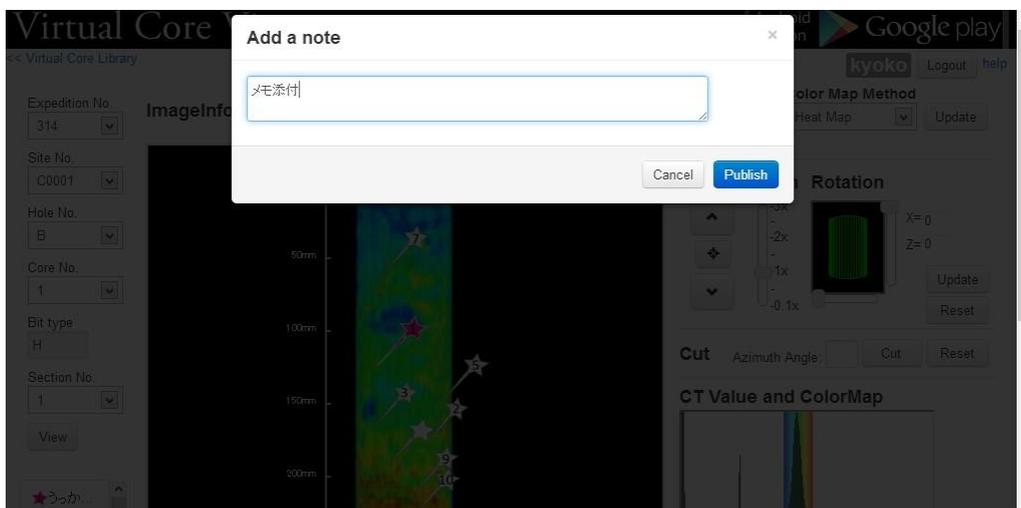


図 4.11 メモの添付-添付内容入力

4) Twitter との連携

操作エリアには、Twitter のマークが付いているボタンを押すと、生成した URL を Twitter のタイムラインに投稿することができる。図 4.9 に示す。



図 4.12 Twitter との連携

3. ログインエリア

このエリアには、ログインというボタンとヘルプページに遷移するリンクボタンがある。システムにログインすることにより、利用できる機能も増えるのである。また、システムに対して、不明点がある場合は、ヘルプページにリンクし、マニュアルを読むこともできる。実際の画面は以下に、図 4.10 と図 4.11 に示す。

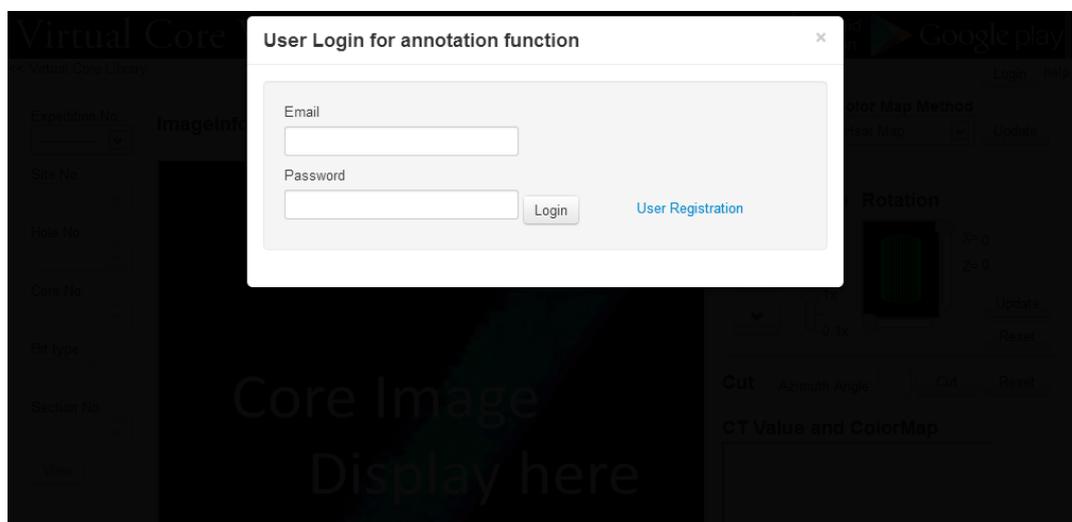


図 4.13 ログイン画面

User Guide

- 1.Core Number
- 2.Core Image
- 3.Bookmark and Annotation
- 4.Display Method
- 5.Moving and Zooming
- 6.Cut Image
- 7.CT Value
- 8.Transparency
- 9.Commentary

About Application

By using the functions of these areas, you can see the core image you want, add comments, save the image and share URL via Twitter

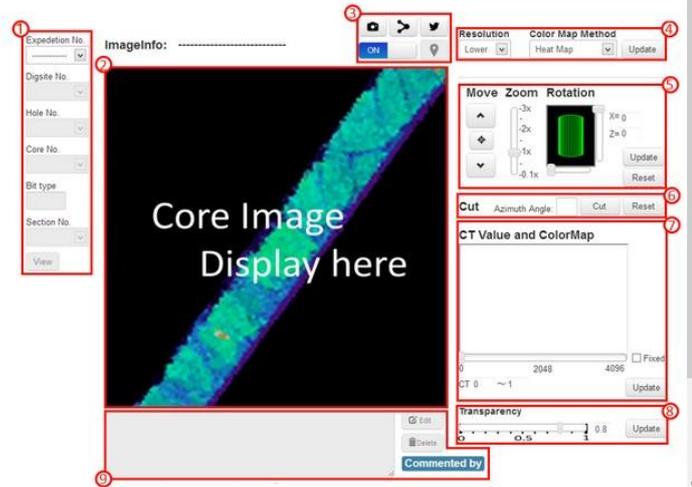


図 4.14 ヘルプページ画面

4. メモ内容表示エリア

このエリアで、選定されたメモ内容が表示されているエリアである。さらに、システムログインしている状態であれば、選定されたメモ、即ち、表示されているメモに対し、削除や内容変更など操作ができる。実際のイメージは、添付する方法と、似ているので、ここで図の説明を省略する

上述にメモの閲覧、添付、削除、編集について説明した際に、権限な話について触れた。もっと分かりやすいために、表 4.1 に示すようにまとめた。

表 4.2 システム権限

	閲覧	添付	編集	削除
ログインしている状態	○	○	○	○
ログインしていない状態	○	×	×	×

○は権限があり、×は権限なしを表している。

なぜシステム利用者に対して、権限をつけたのは、本プロジェクトでは、アノテーション機能を利用する際に、メモの添付、削除、閲覧、編集により、アノテーション DB に対する操作も行う。そのため、直接に DB を操作できるようになった。悪意があるユーザにより、大量の無意味なメモを DB に添付することも考えられる。大量の無意味なメモが添付されるなら、サーバに負担がかかるし、ユーザは、システムを利用する際に、コア画像の閲覧にも妨害することも考えられる。また、ユーザの添付したメモに対し、悪意の削除・改竄することもできる。メモ内容は、ユーザから、コア試料の研究にとって、大変貴重な知見やコメントので、研究の価値が高いメモが簡単になくなり、改竄されるなら、研究者にとって、大きな損失である。このような悪意がある操作を防ぐため、メモの閲覧以外の操作、メモの添付、メモの編集、メモの削除に対し、ユーザ管理機能を追加し、ユーザの権限により、アノテーション機能の操作を制限することを決めた。そのため、本プロジェクトでは、ユー

管理機能を追加し、メモの添付、閲覧、編集、削除の操作には、システム利用者に対し、権限を区別しているような仕様にした。ユーザ登録の手順としては、メールアドレス、ユーザ名、パスワードの3つを使用する。この3つのデータをJAMSTECにユーザ申請し、申請が通ったユーザは次回からそのアカウントを利用しログインできるようになる。ログインしているユーザはメモの添付、自分が添付したメモに対する編集と削除の権限を持っている。

さらに、利用者は操作する権限が持っていない場合は、操作に使うボタンも無効状態になる。たとえば、ログイン出来ない状態で、新規システムのホームページにあるメモ添付のボタンは無効となる。

本節の最後で、ユーザビリティの向上のため、UIの改善について説明する。

新規機能の追加することにより、UIは変更したりしたが、システムの使用性とUIの一致性を考え、次の5つ構成要素に従い、昨年度に開発されたシステムに設計したUIについて、ボタン、テキストエリアなどコンポーネントのサイズと位置、フォントサイズ、表示名などに対し、昨年度に使用されたBootstrap[15] StyleのUI設計フレームワークを利用し続け、細かい変更を行った。既存システムを拡張する際に、メモリスト表示エリアと、機能操作エリア、ログインエリア、メモ内容表示するエリアを追加するとともに、既存システムの従来に存在している①コアデータ検索エリアと②コア画像に対し、拡大縮小、回転、移動など祖操作を行うコアデータ分析エリアのUIについて、変更作業も行った。図4.15に示す。

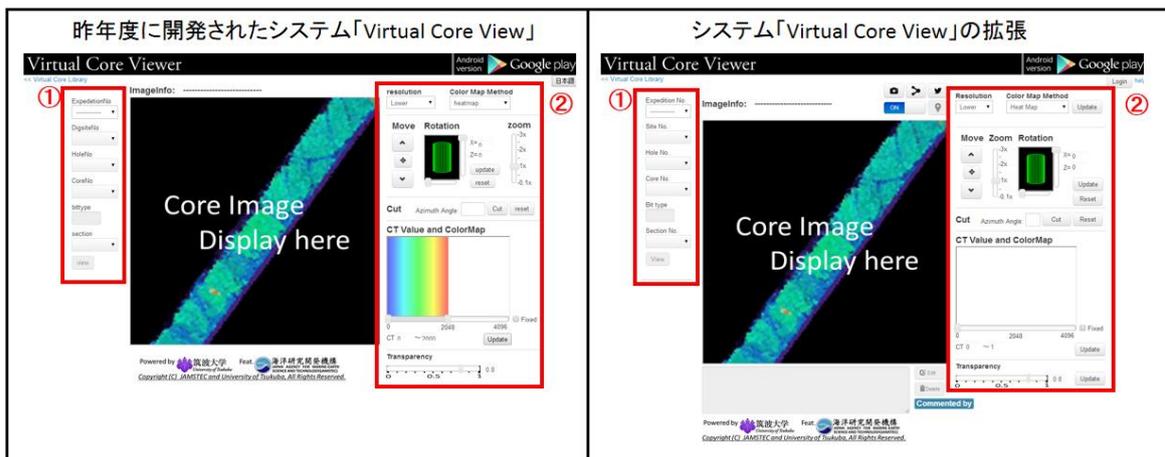


図 4.15 既存システムと拡張システムの比較図

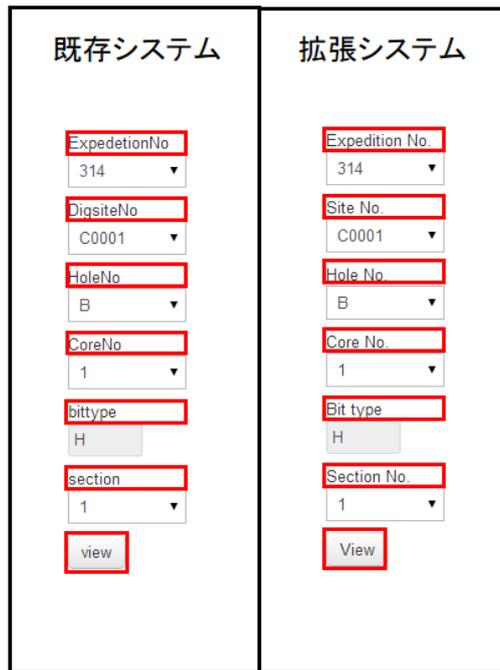


図 4.16 コアデータ検索エリア

変更前と変更後のコアデータ検索エリアのUIは、図 4.16 に示す。既存システムのコアデータ検索エリアでは、表示名は「ExpeditionNo」「DigsiteNo」「HoleNo」「CoreNo」「bittype」「section」となっている。UI の設計から考えると、表示名の頭文字の大文字と小文字が一致していないところは問題点である。そこで、システムの表示名「ExpeditionNo」「DigsiteNo」「HoleNo」「CoreNo」「bittype」「section」から、「Expedition No.」「Site No.」「Hole No.」「Core No.」「Bit type」「Section No.」に変更した。さらに、検索動作を確認するボタン「view」の表示名も、他の操作エリアのボタンの表示名を一致するため、頭文字が大文字の「View」に変更した。

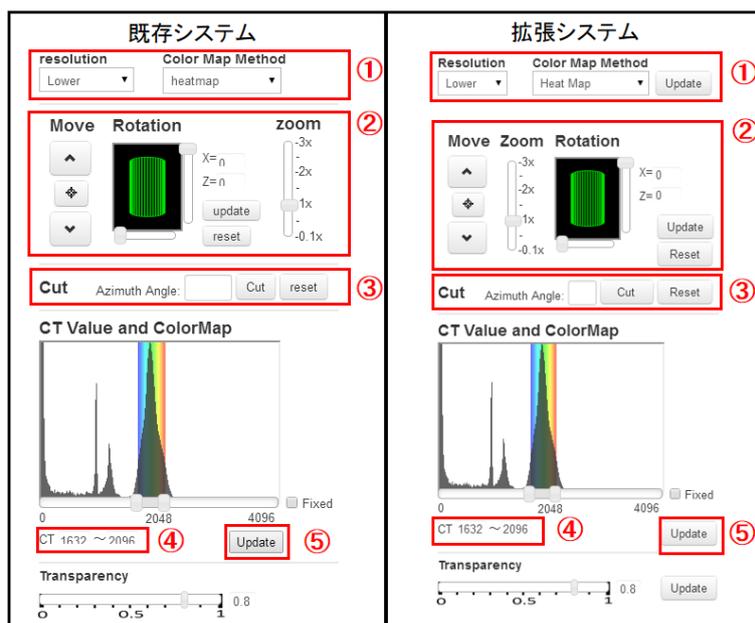


図 4.17 コアデータ分析エリア

変更前と変更後のコアデータ分析エリアの UI は、図 4.17 に示す。このエリアの変更点が多いため、付けた番号で、エリアを分け、説明する。

- ① プルダウンメニューの表示名は「resolution」から、「Resolution」に変更した。「Update」ボタンを1つ追加した。
- ② スライダーの表示名「zoom」から「Zoom」に変更した。また、Move、Rotation、Zoom というコンポーネント3つの表示場所を変更した。分かりやすいために、この3つのコンポーネントは、コアデータ分析エリアから、抽出し説明する。図 4.18 に示すように、Move、Rotation、Zoom というコンポーネント3つがある。今年の UI 設計で②、③の位置が置き換えた。それは、他の操作エリアには、Update というボタンはすべて操作エリアの右になっている、そのため、去年の UI を変更することとした。

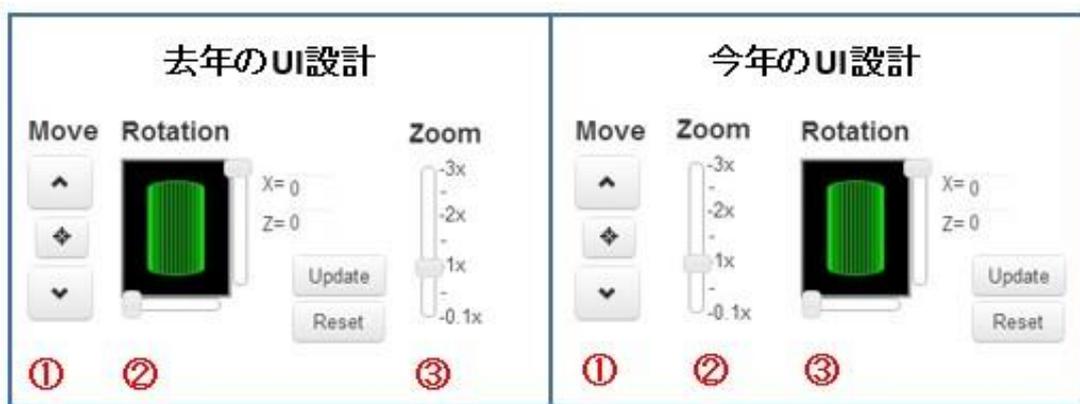


図 4.18 UI 変更の例

- ③ このコア画像に対する切断角度の入力するところである。角度を入力する際に、179度は最大値となるため、テキストエリアを3桁文字まで表示できるように設定した。また、更新ボタンの表示名は「reset」から「Reset」に変更した。
- ④ CT 値を入力するテキストエリアのサイズを拡大した。既存システムでは、入力する CT 値は、テキストが表示できるサイズより、やや大きいので、入力値の下は崩れている。そのため、表示に使うテキストを拡大した。
- ⑤ 「Update」ボタンの仕様を変更した。既存システムでは、このボタンの仕様は HTML のデフォルトボタンの仕様を使用しているため、他の操作エリアのボタンを使用しているスタイルと相違しているため、「Bootstrap」というフレームワークを利用し、同じスタイルのボタン仕様に変更した。

次に、UI の変更以外にも、ユーザビリティの向上するため、操作方法も変更した。ユーザはシステムを利用する際に、適切なコア画像を得るため、CT 値を設定しなければならない。既存システムシステムの場合、コアデータを選定した後、サーバにリクエスト時に色付けする CT 値の初期値を 0 から 2000 に固定していたため、サーバから、送ってきた画像は、閲覧に適切ではない、CT 値を変更し、サーバに再処理リクエストを送信しないといけない。変更したシステムは、サーバに送信するデフォルト CT 値をコアデータごとに、生成するため、サーバに初リクエストした後で、ある程度、閲覧に適切なコア画像が送られる。図 4.19 に示すのは、コアデータの「Expedition No.」「Site No.」「Hole No.」「Core No.」「Bit type」「Section No.」は「314」「C0001」「B」「1」「H」「1」で検索した後、実際に送られたコア

画像である。

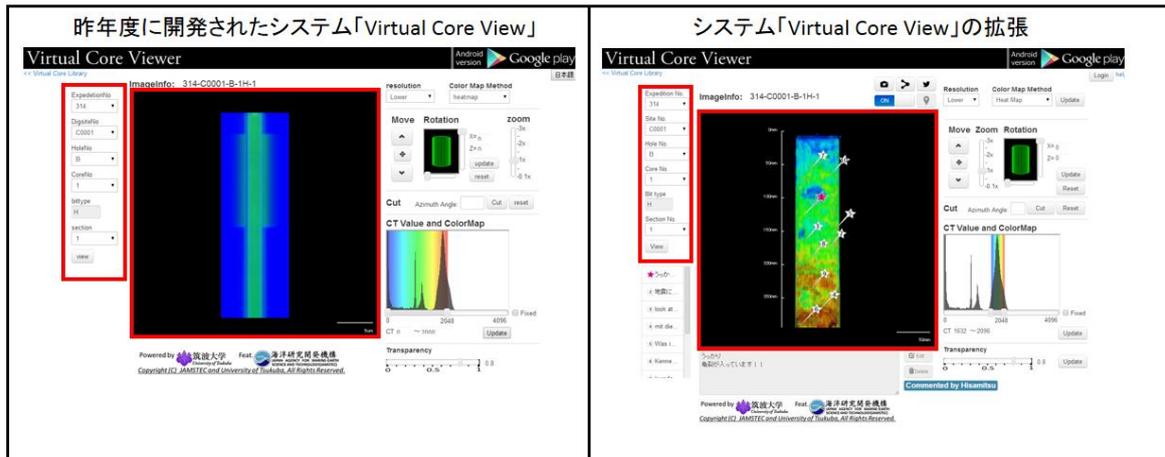


図 4.19 サーバに初請求した後に生成したコア画像

4.3 機能の設計と実装

4.3.1 ブックマーク機能

ブックマーク機能は、閲覧しているコア画像に対する操作情報を保存し、コア画像を再現する機能である。コアデータを閲覧する際に、適切な閲覧効果を得るため、コアデータに対し、拡大縮小、回転、切断など基本操作と CT 値による色付け、透明度の設定など複雑で細かい設定が必要である。さらに、この閲覧状態を再現する際に、コアデータに対し、全く同じ操作が必要なので、システム利用者にとって、無駄な時間がかかる。非効率的なことである。ブックマーク機能の追加することにより、コアデータに対する操作情報を保存し、同じ閲覧状態の再現する URL を作成することが可能となる。PC 上の Web ブラウザまたは「Virtual Core Viewer」がインストールされている Android 端末からその URL を開くと、各クライアントにて同じコア画像が再現される。さらに、URL の生成する機能を利用し、Twitter との連携し、ワンクリックで Twitter のタイムラインに投稿できる。

次に、ブックマーク機能を実現するため、ゲートウェイとレンダリングエンジンによる構成されたサーバとの通信の流れを説明する。

URL を生成するボタンをクリックすると、ユーザがコア画像に対し、拡大縮小、回転、視点移動、画像切断、CT 値に対する色付けなど操作情報をクライアントから抽出され、パラメータを含めた URL が生成される。この URL を利用し、Web ブラウザで開くと、操作情報が分解され、HTTP 規則に使う JSON 形式のデータが生成される。JSON データは、ゲートウェイに定義された API に従い、送信し、ゲートウェイとレンダリングエンジンで構成されたサーバで、操作情報により、描画処理を行い、処理した結果は、クライアントに送信する。このような一連の流れで、コア画像の再現が実現される。また、Twitter に投稿するボタンをクリックと、Twitter の API と関連し、生成した URL は、Twitter のタイムラインに投稿することができる。

次に、Web クライアントアプリケーションで、ブックマーク機能を実現するために、利用

されたメソッドとその仕組みについて、説明する。説明に使用するメソッドの詳細は付録に記載してあるので、詳細内容は付録に参考してください。

ブックマーク機能を実現するために、まず、Webクライアントで、表示している画像情報を記録できるメソッド `createURL` を作成した。さらに、レンダリングエンジンにパラメータを送信する前に、事前に生成した URL を置き換えれば、URL に含めた情報をすべて再現することが可能になると考え、Webクライアント側から、JavaScript で、送信するパラメータに対し、判断メソッドを追加した。ブックマーク機能を利用している場合は、`createURL` メソッドで生成したパラメータを置き換えることで、ブックマーク機能を実現した。メソッドの詳細は付録に記載してあるので、ここで省略する。

このように、従来システムのクライアントアプリケーションの修正と追加することで、ブックマーク機能を実現した。

4.3.2 アノテーション機能

アノテーション機能は、コアデータに対して、システム利用者の知見やコメントは、メモの形で、コア画像の指定された位置に添付する機能である。システム利用者は、コアデータを閲覧中に気になる部分があった場合、コアデータ画像の気になる位置をクリックすることにより、その位置に、システム利用者の知見やコメントを含めるメモを添付することができる。次回で、システムを利用する際に、自分が添付したものに対し、振り返ることができる。さらに、このメモは個人だけでなくすべてのシステム利用者が閲覧することが可能である。

次に、ブックマーク機能を実現するため、ゲートウェイとレンダリングエンジンによる構成されたサーバとの通信の流れを説明する。

アノテーション機能は、メモの添付、閲覧、編集、削除の4つ機能から構成される。各機能の実現する流れとしては、同じになっている。まず、Webクライアントから、メモに対する操作情報を抽出し、Webクライアントで、ゲートウェイで定義されたAPIに従い、JSON形式のデータを作成する。その後、HTTP規則を利用し、ゲートウェイに、JSON形式のデータを送信する。ゲートウェイで、リクエストに応じ、レンダリングエンジンと連携し、ビジネスロジック計算を行う。計算した結果は、Webクライアントに送信する。Webクライアントで、ゲートウェイからの計算結果を利用し、ユーザに見えるようなGUIを作成する。この一連の流れで、アノテーション機能が実現される。

次に、Webクライアントアプリケーションで、アノテーション機能を実現するために、利用されたAPIについて、説明する。

メモの閲覧

表 4.3 レンダリングエンジン API の取りうる値-メモ閲覧

変数名	概要
<code>corenum</code>	コア番号
<code>rotx</code> , <code>roty</code> , <code>rotz</code>	x、y、z 方向の回転角度
<code>cut</code>	切断角度
<code>scale</code>	1 ピクセルあたりの長さ
<code>depthto</code>	コアが描画される範囲の始点
<code>depthfrom</code>	コアが描画される範囲の終点

表 4.4 レンダリングエンジン API の返り値-メモ閲覧

変数名	概要
id	メモに一意に付けられた数値
x、y	メモを表示する x、y 座標[px]
annotate	メモの内容
name	メモを記入したユーザの名前
commission	メモを編集・削除する権限があるか

メモの添付

表 4.5 レンダリングエンジン API の取りうる値-メモ添付

変数名	概要
annotate	保存するメモの内容
x、y	メモを表示する x、y 座標[px]
corenum	コア番号
rotx、roty、rotz	x、y、z 方向の回転角度
cut	切断角度
scale	1 ピクセルあたりの長さ
depthto	コアが描画される範囲の始点
depthfrom	コアが描画される範囲の終点

表 4.6 レンダリングエンジン API の返り値-メモ添付

値	概要
0	成功
-1	コアの範囲外を選択 (x 座標)
-2	コアの範囲外を選択 (y 座標)
-3	データベースに登録できない
-4	ユーザ未登録

メモの編集

表 4.7 レンダリングエンジン API の取りうる値-メモ編集

パラメータ	概要
id	編集するメモの id
annotate	編集するメモの内容

表 4.8 レンダリングエンジン API の返り値-メモ編集

値	概要
0	編集成功
-1	編集失敗
-2	ユーザ未登録

メモの削除

表 4.9 レンダリングエンジン API の取りうる値-メモ削除

パラメータ	概要
id	削除するメモの id

表 4.10 レンダリングエンジン API の返り値-メモ削除

値	概要
0	削除成功
-1	削除失敗
-2	ユーザ未登録

メモの閲覧、添付、編集、削除の Web クライアント処理について説明する。

最初にメモの閲覧について解説する。閲覧するコアデータを選定し、サーバにコアデータ画像生成する請求を送信する。同時に、上述した通りの閲覧に必要な API も HTTP GET でゲートウェイに送信する。ゲートウェイにて、請求を受け、表示するメモ内容を DB で検索する、ビジネスロジック処理の結果は、Web クライアントに送信する。クライアントで、返してきた結果を利用し、JAVASCRIPT で書いたメソッドを呼び出し、その結果を計算し、最後に、Web クライアントで、計算した結果を、HTML と CSS で処理し、表示させる。

次にアノテーション添付の処理の流れについて説明する。添付機能が持っているボタンを押すと、コアデータ画像が表示されているエリアに、マーカーが出る。そのマーカーを使い、メモを添付したい箇所で、左クリックで添付位置を選定できる。選定した後、確認ダイアログが表示される。そこで、内容を入力し確認ボタンを押すと、Web クライアントからサーバにメモ添付 API が送信される。サーバで請求を受け、ビジネスロジック処理をした後、結果を Web クライアントに送信する。成功した場合、Web クライアントで「メモ添付成功」とメッセージが表示され、不成功の場合は「失敗」というメッセージが表示される。

最後に、メモの削除と編集の流れについて説明する。編集の場合は、Web クライアントの編集機能が持っているボタンを押すと、ダイアログが表示される。それを編集し、確認ボタンを押すと、メモ編集 API が、サーバは請求に送信する。サーバで、その請求を応じ、処理結果を返す。削除の場合も、同じく、Web クライアントから、サーバに送信し、処理を行う。

メモの「閲覧」、「添付」、「削除」、「編集」について、Web クライアントに JAVASCRIPT でメソッドを作成した。作成したメソッドは 2 種類がある。ボタンやテキストエリアコンポーネントのイベントメソッドとイベントが喚起される際に、呼び出しに使われるメソッドである。

Web クライアントアプリケーションには、JAVASCRIPT で 23 個の関数が定義された。関数内容の詳細説明について、付録に記載した。

4.3.3 ユーザビリティの評価実験

実験目的

本年度の研究開発で、既存システムに対し、ブックマーク機能とアノテーション機能を拡張した。その結果として、UI や操作方法も変更した。新しいシステム UI を利用する際に、拡張したシステムのブックマーク機能とアノテーション機能の操作方法を説明しない状況で、この2つの機能を使用できるかどうかを目的として、この実験で検証を行った。

実験内容

ユーザビリティの評価手法には、主に3つの種類がある。

1. ヒューリスティック評価

ヒューリスティック評価とは、ユーザビリティの専門家が、評価対象のサイトを見て、様々な問題点を指摘する手法である。

2. ユーザビリティテスト

ユーザビリティテストとは、ユーザに実際に評価対象のサイトを使ってもらい、その使用過程を観察することで、様々な問題点を発見する手法である。

3. 認知的ウォークスルー

認知的ウォークスルーとは、ヒューリスティック評価とユーザビリティテストをミックスしたような評価手法である。

今回、Webクライアントアプリケーションのユーザビリティの評価は、ユーザビリティテストという手法で行った。被験者は、筑波大学大学院の学生2名、IT関連の仕事をしている社会人2名、計4名20代の若者である。著者は、拡張システムの概要と目的を被験者に説明した後、被験者に、拡張したWebクライアントアプリケーションを使用して、実験タスクを実施してもらった。実験項目は、表4.11に示す。その後、各実験タスク内容について、5段階評価のアンケート調査を行っていた。

表 4.11 実験タスク

実験タスク番号	操作内容	検証範囲
1	メモ1つを添付する	アノテーション機能
2	メモを非表示にする	アノテーション機能
3	メモを表示にする	アノテーション機能
4	添付したメモ内容を変更する	アノテーション機能
5	添付したメモを削除する	アノテーション機能
6	URLを生成する	ブックマーク機能
7	生成したURLで閲覧状態を再現する	ブックマーク機能
8	Twitterのタイムラインに投稿する	ブックマーク機能

被験者は、実験を行っている際に、被験者のシステムの使用状況を記録した。Webクライアントアプリケーションを使用し、実験タスクの8項目が完成していなかった被験者はいなかった。機能を使用する際に、ヘルプページを参照した被験者は1人がいた。操作に一番時間がかかったのは、メモの添付で、4名の被験者の平均時間は7秒であったが、メモの非表示にするとTwitterのタイムラインに投稿するに使う時間は、一番短く、1秒であった。その他の操作は、基本的に、2秒から4秒までであった。詳細内容は、表4.12に示す。

表 4.12 実験タスクの結果

操作内容	完成の平均時間	ヘルプページを利用した人数
メモ1つを添付する	7秒	1
メモを非表示にする	2秒	0
メモを表示にする	1秒	0
添付したメモ内容を変更する	4秒	0
添付したメモを削除する	2秒	0
URLを生成する	2秒	0
生成したURLで閲覧状態を再現する	2秒	0
Twitterのタイムラインに投稿する	1秒	0

この結果により、ブックマーク機能とアノテーション機能の完成度が高いとも言えるのではないだろうか。しかし、メモ添付機能は、被験者の1人がヘルプページを参照してから、完成したので、この部分はまだ改善できる箇所があるのではないだろうか。以上は、すべての実験の結果であった。次に、アンケートの調査結果について、説明する。今回のUI設計としては、新規されたエリアは、メモリスト表示エリア、機能操作エリア、ログインエリアメモ内容表示するエリアの4つから構成されている。各エリアについて、アンケート内容を設計し、5段階評価の調査お行っていた。アンケート調査の内容と結果は、表4.13に示す。

表 4.13 アンケート調査の内容と結果

調査番号	アンケート内容	平均得点
メモリスト表示エリア		
1	メモを選定し、コア画像にハイライトで表示される	4
機能操作エリア		
2	コア画像の保存	4.5
3	操作情報を記録したURLの生成	5
4	生成したURLを利用し、閲覧状態の再現	4.75
5	Twitterのタイムラインに投稿する	5
6	メモの非表示と表示	4.25
7	メモの添付	3.25
ログインエリア		
8	システムにログインする	4
9	システムにログアウトする	5

10	ヘルプページにリンクする	4.75
メモ内容表示するエリア		
11	メモ内容の編集	3.5
12	メモの削除	4
13	メモの詳細内容の閲覧	3.5
全項目の平均得点		4.27

アンケート調査の結果から見ると、平均得点は4.27となる。ブックマーク機能とアノテーション機能の使いやすさは良いといえるのではないだろうか。しかし、その中で得点は、3.25となる「メモの添付」や3.5となる「メモ内容の編集」と「メモの詳細内容の閲覧」など得点が低い項目まだ存在している。システムのユーザビリティはまだ改善する余地があるのではないだろうか。特に、アノテーション機能に関するところ、得点は良くなかった。

一方で、本プロジェクトの進み方とし、S0,S1,S2,S3,4つのスコープに分かれ、各スコープの終了の時点で、ユーザテストを行った。そのため、S0以外の各スコープで、ユーザからのフィードバックの反映も各スコープの作業となる。最終S3で完成したWebクライアントアプリケーションは、顧客と共同作業した結果とも言える。そのため、最後のフィードバックは、ユーザビリティの評価は、顧客様のフィードバックを参考にした。フィードバックの結果から見ると、ブックマーク機能とアノテーション機能はよかったが、コアデータの描画するスピードはまだ改善する必要がある。

第5章 おわりに

本プロジェクトでは、コアデータの利用だけではなくシステム利用者の間にも、情報共有を可能とするコミュニティツールを構築するため、既存システムに「Virtual Core Viewer」を拡張し、ブックマーク機能とアノテーション機能を追加した。新機能の追加により、新たな「海底コア CT スキャンデータ可視化・情報共有を可能とするクラウドサービス」を構築した。本プロジェクトの成果により、システム利用者間のコミュニケーションの向上が期待される。

著者は、ブックマーク機能とアノテーション機能を実現するため、Web クライアントアプリケーションを開発した。各チームメンバーの共同努力下、新しいクラウドサービスが構築された。ブックマーク機能の追加により、閲覧状態の再現はワンクリックで、簡単になった。また、個人が利用するだけでなく、この閲覧状態を他の人と共有することもできた。自分が見ていて気になったので他の人にも見てもらいたい、といった場面において、以前は、深さや回転角といった操作の情報を詳細に伝えなければならなかったが、この機能を実装したことにより、簡単に共有できるようになった。これは、同じ研究所の人とメールで共有することもできるだけでなく、Twitter のような SNS を利用することにより世界中の研究者と共有することもできる。さらに、アノテーション機能の追加により、地質研究者のコミュニティを活性化させられることも考えられる。システムのブックマーク機能とアノテーション機能利用し、自分のコメントや他人のコメントは、簡単に共有できるようとなり、システム利用者間の学術コミュニケーションや議論を増加することにより、コアデータの解析が飛躍的に進むと期待され、地質学での新たな発見を促進させることができる。

今後の展望として、コアデータの研究は、人類歴史の解明や未来の発展にとって、とても意義があることである。新規システムの SNS 機能の活用することにより、各分野の研究者を巻き込み、システム利用者的人数を増やし、ユーザビリティを向上したい。また、コアデータだけではなく、すべての DICOM フォーマット画像が対応できる汎用システムを進化させたい。

謝辞

本研究開発の遂行にあたり、終始適切な助言を賜り、また丁寧に指導して下さった山際伸一准教授、和田耕一教授には、心より感謝いたします。共同研究者である、高知コア研究所の久光敏夫様には、誠心誠意なご協力をいただき、ここに感謝いたします。本当にお世話になりました。指導教員である田中二郎教授には、二年間の勉強や生活において多くの励ましを頂き、ここに深く感謝いたします。高度 IT 人材育成のための実践的ソフトウェア開発専修プログラムの担当である山戸昭三先生、中沢研也先生には、授業や生活はもちろん、今後の進路についても、貴重なアドバイスや支援をいただき、お世話になりました。また、筑波大学大学院システム情報工学研究科コンピューターサイエンス専攻高度 IT 人材育成のための実践的ソフトウェア開発専修プログラムの担当事務員の皆様には、この二年間も大変お世話になりました。ここに深く感謝いたします。プロジェクトメンバである、豊田恭子さん、黒田くん、城崎亮くん、伊藤弘貴くんには、一年間プロジェクトでお世話になりました。ここに感謝いたします。最後に、これまで支えてくださった、先生、友人、家族をはじめ、お世話になった全ての方々に感謝を申し上げます。

参考文献

- [1] 倉本 真一. 交流のひろば 地球深部探査船「ちきゅう」とは?. 建設の施工企画.. (752):2012.10. 55-58 ISSN 1349-547X
- [2] IODP. <http://www.iodp.org/>
- [3] 浦辺 徹郎. 国際陸上科学掘削計画 (ICDP) の発足とわが国の陸上掘削. 地学雑誌、 Vol. 122、 No. 2、 p. 250-257、 2013.
- [4] 森下 知晃. 掘削科学 21世紀マントル掘削計画：海洋プレートの1点突破から全貌解明へ. JGL = 日本地球惑星科学連合ニュースレター：Japan geoscience letters / 広報・アウトリーチ委員会 編.. 9(4):2013.11. 4-6 ISSN 1880-4292
- [5] Yuichiro Sakamoto、 Shin Sasaki、 Takaya Okamoto、 Shinichi Yamagiwa、 Toshio Hisamitsu、 and Koichi Wada. A visualization cloud service for x-ct dicom images applied to deep-sea drilled core database. Bulletin of Networking、 Computing、 Systems、 and Software、 Vol. 2、 No. 1、 pp. pp-53、 2013.
- [6] 佐々木 慎. 海底コア CT スキャンイメージ可視化のためのクラウドサービスの開発. 筑波大学大学院博士課程システム情報工学研究科特定課題研究報告書、 2012
- [7] 坂本 侑一郎. 海底コア CT スキャンイメージ可視化のためのクラウドサービスの開発. 筑波大学大学院博士課程システム情報工学研究科特定課題研究報告書、 2012
- [8] 岡本 昂也. 海底コア CT スキャンイメージ可視化のためのクラウドサービスの開発. 筑波大学大学院博士課程システム情報工学研究科特定課題研究報告書、 2012
- [9] Kochi Core Center. <http://www.kochi-core.jp/>
- [10] Virtual Core Library. <http://www.kochi-core.jp/VCL/index.html>
- [11] DICOM. <http://ja.wikipedia.org/wiki/DICOM>
- [12] M.Yakami、 K.Ishizu、 T.Kubo.T.Okada.andK.Togashi、”DevelopmentandEvaluation ofaLow-CostandHigh-CapacityDICOMImageDataStorageSystemforResearch、”Journalof DigitalImaging、 Springer、 vol.24、 no.2、 pp.190-195、 April2011.
- [13] OsiriX. <http://www.osirix-viewer.com/>
- [14] 阿部 功、大沼 和矢.開発プロジェクトのための進捗管理システムの導入、 2012
- [15] Bootstrap. <http://getbootstrap.com/>
- [16] JSON. <http://www.json.org/>

付録 Web アプリケーションの関数定義

関数名

saveImage0{...}

返り値

なし

処理内容

コアデータ画像を Web クライアントに保存する。

関数名

setPin0{...}

返り値

なし

処理内容

Web クライアントには、メモ位置を表示するマーカーを作成する。

関数名

showPin(XXX、YYY、id、memo、name、commision、pf) {...}

返り値

なし

処理内容

サーバを返してきた結果をメソッドの引数として入力し、Web クライアントにメモを作成する。

関数名

pinSet0{...}

返り値

なし

処理内容

Web クライアントには、メモ位置を表示するマーカーの位置を確認した後で、サーバにマーカーの座標を送信する。

関数名

pinGet0{...}

返り値

なし

処理内容

サーバに、メモを作成するリクエストを送信する。

関数名

createUrl0{...}

戻り値

なし

処理内容

現在には、表示されているコアデータ画像の情報を含める URL を作成する。この URL を利用し、コアデータ画像の再生ができる。

関数名

twitterPost0{...}

戻り値

なし

処理内容

createURL0で、生成した URL を Twitter のタイムラインに投稿できるようにする。

関数名

textDel0{...}

戻り値

なし

処理内容

メモを削除するダイアログが表示される。

関数名

textEdit0{...}

戻り値

なし

処理内容

メモを編集するダイアログが表示される。

関数名

addMemo0{...}

戻り値

なし

処理内容

メモを添付するダイアログが表示される。

関数名

addCancel0{...}

戻り値

なし

処理内容

メモ添付することを中止する。

関数名

`getQueryParam(paramName、 url) {...}`

返り値

なし

処理内容

ブックマーク機能を利用する際に、生成した URL を解析する。

関数名

`meta_to_escape(str){...}`

返り値

暗号化の URL

処理内容

生成した URL に対し、暗号化する。

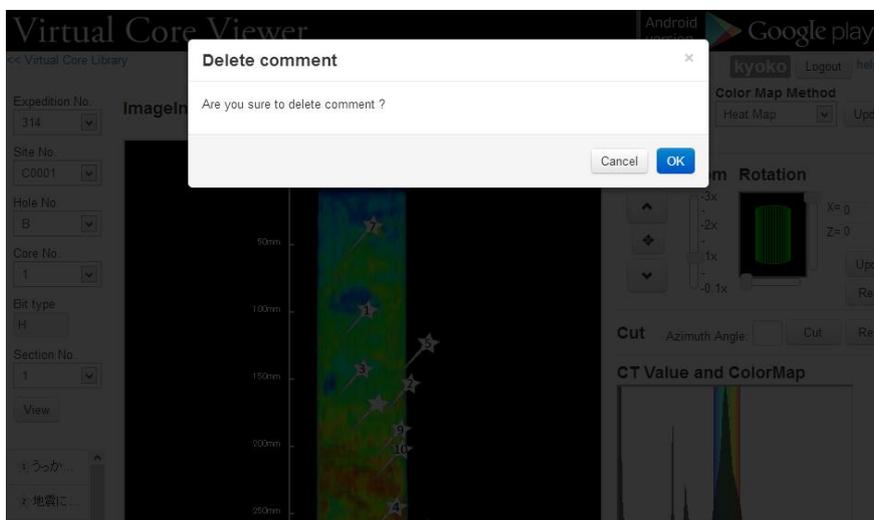
関数名

`$("#DOK").live("click", function(){...});`

処理内容

ポップアップしたダイアログにある削除ボタンをクリックした後で、サーバに削除した ID を送信する。

表示画面



メモ削除の確認画面

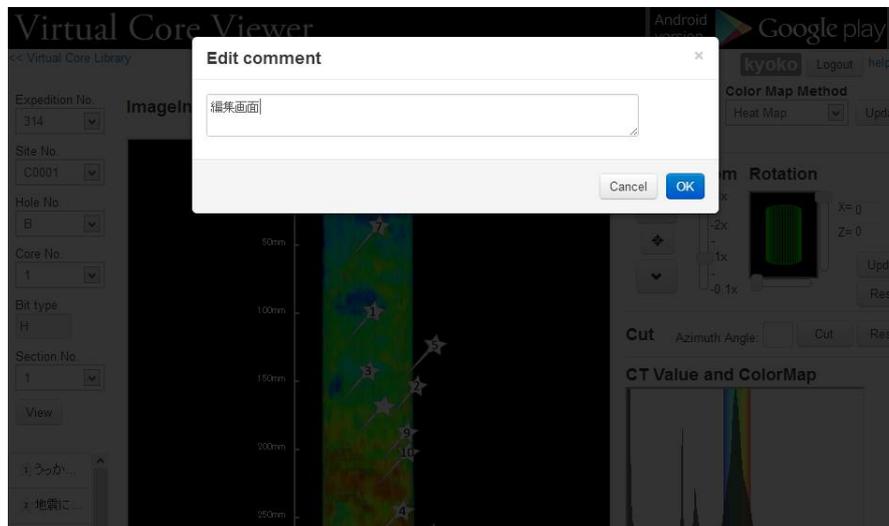
関数名

`$("#EOK").live("click", function(){...});`

処理内容

ポップアップしたダイアログにある編集ボタンをクリックした後で、サーバに削除した ID を送信する。

表示画面



メモ編集の確認画面

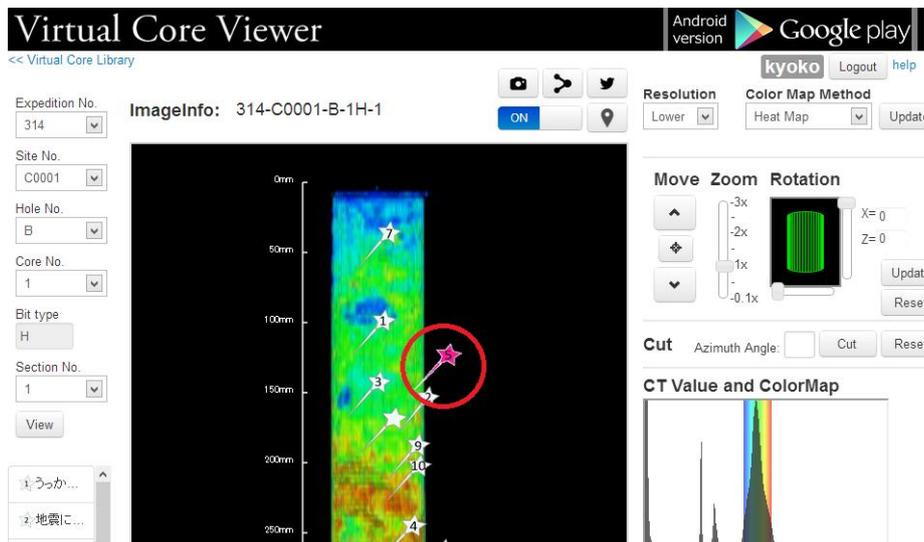
関数名

```
$(".pin").live("click", function(){...});
```

処理内容

コアデータ画像に表示されるピンマーカーをハイライトで表示する。

表示画面



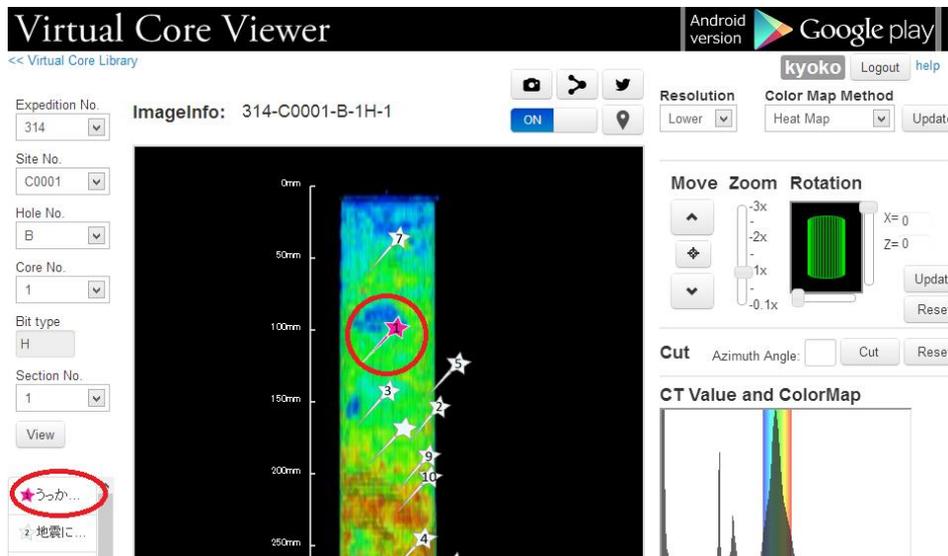
ピンをクリックするイメージ図

関数名

```
$(".memotd").live("click", function(){...});
```

処理内容

メモリストの中で、1つをクリックした後で、ハイライトで表示する。
表示画面



メモリストをクリックするイメージ図

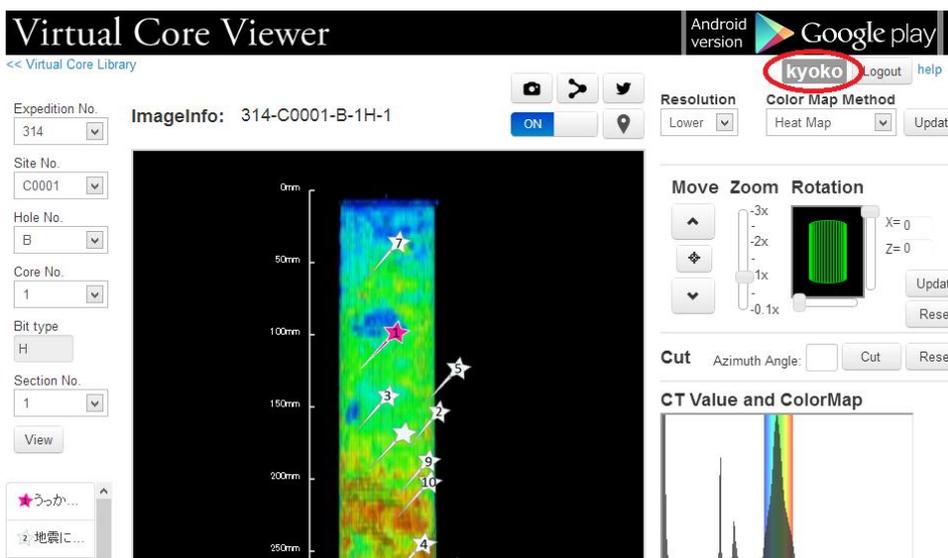
関数名

```
$("#lib").live("click", function(){...});
```

処理内容

ユーザのログインの状態を確認する。ログインしているユーザに対し、ユーザを表示する。

表示画像



ユーザ名を表示するイメージ図

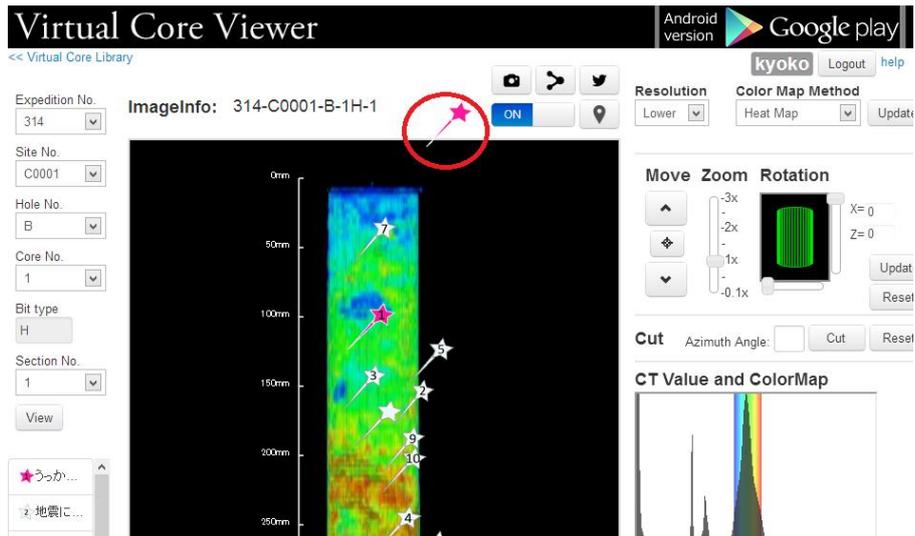
関数名

```
$("#cpb1").live("click", function(){...});
```

処理内容

メモを添付する時、ピンマーカーの位置を選定する前のダイアログ状態から、メモの添付を中止する。ESC ボタンを押すと、メソッドが呼び出される。

表示画像



ピンの位置指定する際のイメージ図

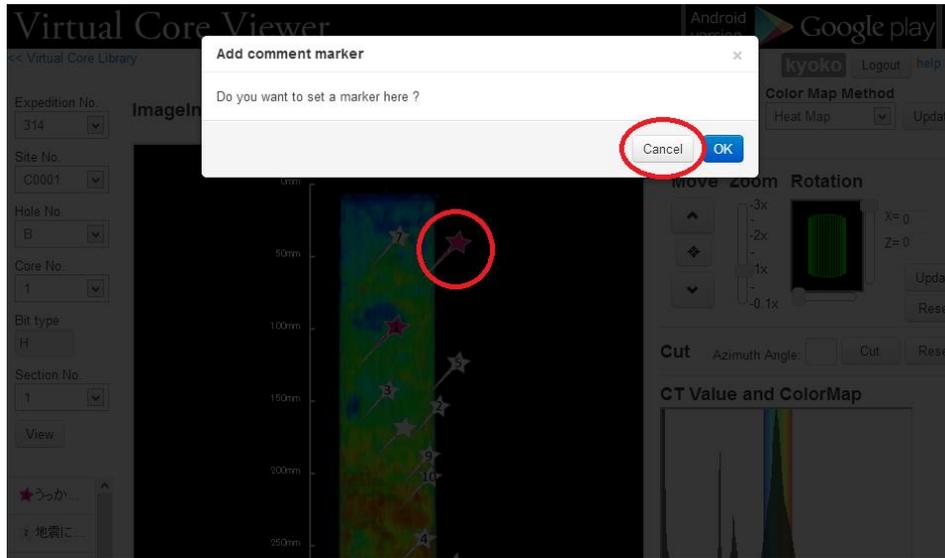
関数名

```
$("#cpb2").live("click", function(){...});
```

処理内容

メモを添付する時、ピンマーカーの位置確選定する後のダイアログ状態から、メモの添付を中止する。

表示画像



Cancel ボタンでのメモ添付を取り消すイメージ図

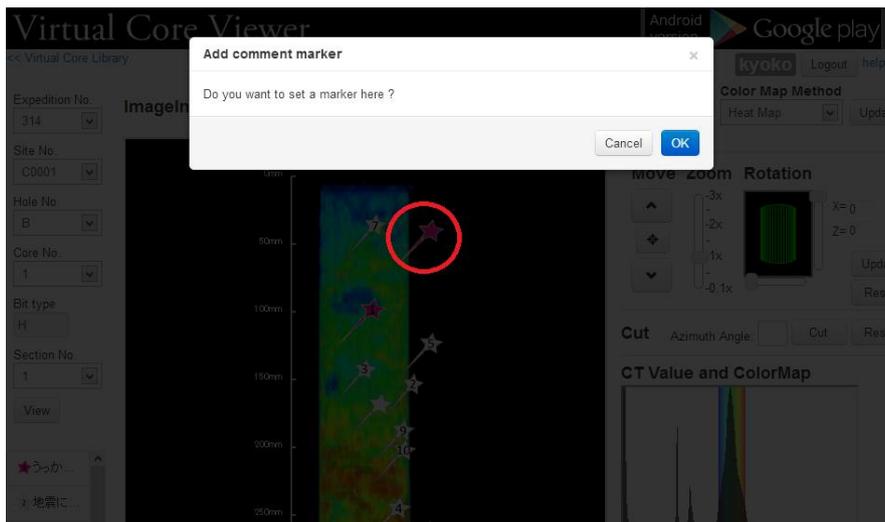
関数名

```
$("#pc").live("click", function(){...});
```

処理内容

メモを添付する時、ピンマーカーの位置確選定した後、確認する前のダイアログ状態から、メモの添付を中止する。(ESC ボタンで)

表示画像



ESC ボタンでのメモ添付を取り消すイメージ図

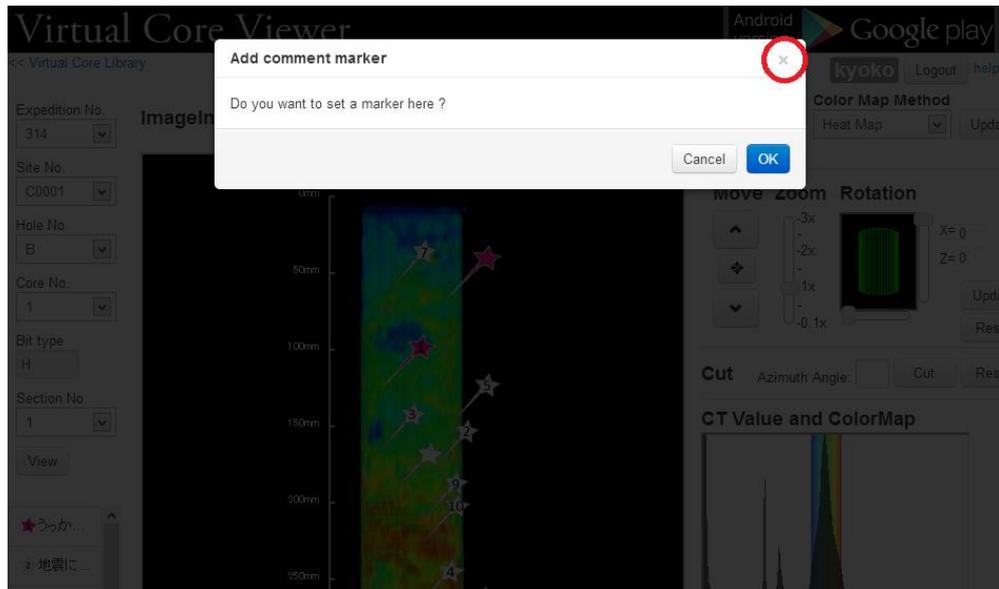
関数名

```
$("#Cancel").live("click", function(){...});
```

処理内容

メモを添付する時、各ダイアログのキャンセルボタンを押した後、メモの添付を中止する。

表示画像



×ボタンでのメモ添付を取り消すイメージ図

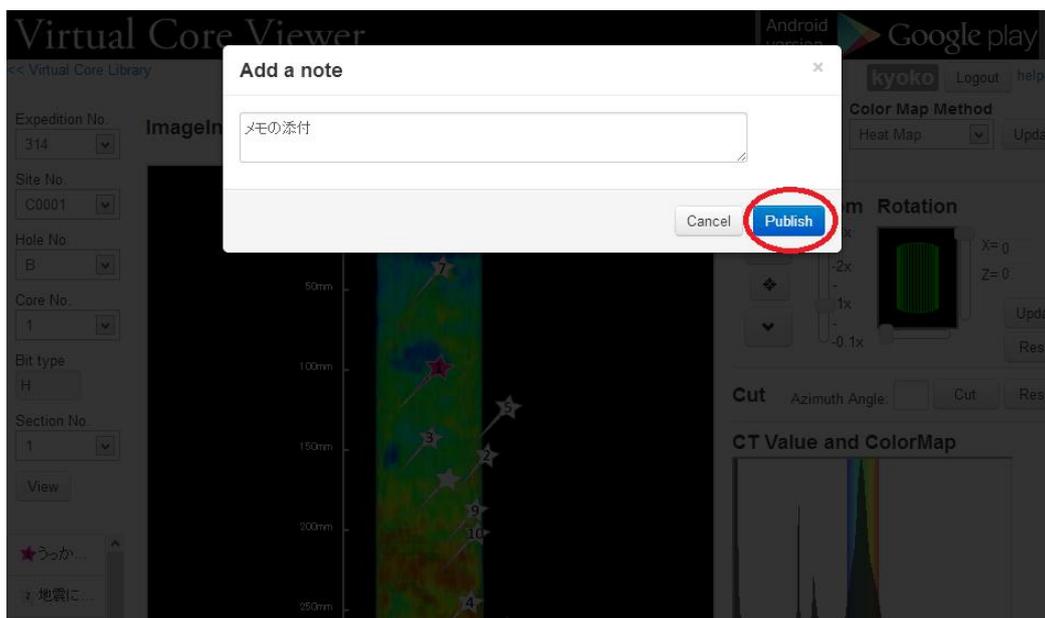
関数名

```
$("#Publish").live("click", function(){...});
```

処理内容

メモの添付を確認した後、サーバにメモの座標や内容などを送信し、DBに保存する。

表示画像



メモ添付の確認画面