

[dvipdfm]graphicx

平成 24 年度

卒業論文

# **Mashup** による **e-learning** コンテンツ検索表示

佐賀大学 理工学部知能情報システム学科

08233014

甲斐 遼馬

指導教官：新井 康平 教授

学科長：渡邊 義明 教授

Year 2012

Graduation Thesis

# The develop of e-learning contents viewer by “ Mashup ”

Saga University Department of Information Science, Faculty of Science  
and Engineering, Saga University

08233014

Ryoma KAI

Supervisor : Professor Kohei ARAI

Chief of Department : Professor Yoshiaki WATANABE

## Mashup による e-learning コンテンツ検索表示

近年、iOS や Android などのタブレット端末・スマートフォン端末の普及が進み、それらの教育目的での利用価値も俄然高く評価されてきている。しかし、教育方面での利用のための GUI は未だに未熟、またはなかなか考慮されにくいのが現状であり、タブレット端末での効果的な学習を支援するサービスを作ることは非常に意義深いことであると考え。本研究では、タブレット端末における e-learning 検索アプリを、mashup と呼ばれる開発手法を用いて柔軟に開発した後、検証を行ったものである。

キーワード： mashup、Android、e-learning

## **Graduation Thesis Overview Year 2012**

# **The develop of e-learning contents viewer by “ Mashup ”**

Tablet and Smartphone (e.g. iOS and Android ) devices today has a fairly, those app's value is appreciated in educational field(e.g. e-learning). However, those app's GUI is inexperienced and not considered carefully. Therefore, it is meaningful to build educational support service. This reseach is development and veriticcation of e-learning search app with mashup in Android.

**Keywords:** Mashup,Android,e-learning

# 目次

第 1 章	序論	1
1.1	背景	1
1.2	本文書の構成	1
第 2 章	Mashup と WebAPI	3
2.1	Mashup とは	3
2.2	WebAPI とは	3
		3
		3
2.2.3	Youtube Data API	4
2.2.4	Product Advertising API	4
第 3 章	検索エンジンの精度向上	5
3.1	先行研究	5
3.1.1	e-learning コンテンツにおけるドキュメントサーチの最適化	5
3.2	サブキーワードの選出	5
3.2.1	選出過程	5
3.2.2	選出結果	5
第 4 章	WWW 視覚化	6
4.1	開発事例	6
4.1.1	納豆ビュー	6
4.1.2	Flowser.com	6
4.2	検索エンジンにおける WWW 視覚化	8
4.2.1	Helix model	8
4.2.2	Star model	8
4.2.3	Star-Helix model	8
4.2.4	Star-Slide model	9
4.3	採用手法	9
第 5 章	開発手法	10
5.1	概観	10
5.2	Away3D	10
5.2.1	Away3D とは	10
5.2.2	Android 上への移植	10
第 6 章	Android アプリ「LEDOXEA」	11
6.1	使用方法	11
6.2	特徴	11
6.2.1	移植性	11
6.2.2	スペック性能への非依存性	11
6.2.3	フリック操作による直感的操作性	11
6.2.4	検索エンジンの同時検索、WWW 視覚化	11

第 7 章 アンケートによる評価と考察	12
7.1 評価結果 . . . . .	12
7.2 考察 . . . . .	12
第 8 章 結論	13
謝辞	14
参考文献	15
付 録 A プログラム	16
A.1 main.as(ActionScript) . . . . .	16
A.2 app.xml(XML) . . . . .	16

# 第1章 序論

本章では、本研究の背景、それを踏まえた上での研究の目標・目的、そして文書の構成について述べる。

## 1.1 背景

2007年、iOS、Android OSの両オペレーティング・システムを搭載したタブレット端末やスマートフォン端末が発表された。これら端末はスペック的にそれほど高くものではないものの、タッチパネルの搭載による直感的な操作、携帯性の高さ、Wi-fi接続によるインターネット接続が可能といった多数のメリットを兼ね備えており、欧米を中心に今日まで爆発的に普及してきている。<sup>1</sup>

一方、e-learningとは、パーソナルコンピュータなどの情報機器を用いて行う学習のことである。1990年代後半からのPCの普及と共に様々な分野で用いられるようになり、現在ではe-learningのコンテンツ共有を目的とした規格[3]や大学設置基準に基づく文部科学省告示の中にe-learningに関する項目が記述される[5]など、制度や規格も整備されたものとなっている。

だがe-learningコンテンツを提供するサイトは、その多くがタブレット端末、スマートフォン端末が発表されるより前に製作されたものである。現状、スマートフォンやタブレットからe-learningコンテンツに対してダイレクトにアクセスするためには、まずはうまくコンテンツだけがヒットするような副次的な検索キーワードを考えて、PC用のサイトから小さなボタンをタップし、コンテンツをダウンロードし、更にテキストや動画で別々の検索エンジンを使わなければならない、といったようにかなりの労力を要する。<sup>2</sup>

当研究では、こういった問題点を改良するため、スマートフォン端末・タブレット端末上でe-learningコンテンツを簡単に検索し、自在かつ直感的に閲覧、ダウンロードできるアプリの開発を行う。

## 1.2 本文書の構成

この第1章では、本論文を書くに至った背景とその構成を説明している。第2章では、検索エンジンの構成に使用したWebAPIと、それらを統合した手法について説明する。第3章でキーワードを用いた検索結果の精度向上手法について説明する。第4章では、WWW視覚化という目線での先行研究や開発事例、そして解決方法についての案を提

---

<sup>1</sup>株式会社シード・プランニングの行った2012年7月の市場調査[1]によると、日本でのスマートフォン普及率は40%前後と先進国の中ではやや低調である。元々高品質な携帯電話が普及しており、プラットフォームが盤石であったことが要因であると考えられている。

<sup>2</sup>iOSについては、e-learning用にユーザーインターフェースが最適化されたiTunesU[4]が存在するが、これはiTunesStore内にあるコンテンツのみを対象としており、WWW上に存在するコンテンツをすべて検索対象とすることはできない。

起する。第5章では、アプリ開発のための言語や手法についての詳細を概説する。第6章では、開発したアプリの利用方法と、その特徴について説明する。第7章では、そのアプリについての評価を行い、考察を述べる。第8章では、本研究のまとめを行い、今後の課題を列挙する。



## 第2章 Mashup と WebAPI

本章では、表題となっている Mashup と呼ばれる開発手法に加え、WebAPI と呼ばれるタイプの API について解説する。

### 2.1 Mashup とは

Mashup とは、2 つ以上の WebAPI を組み合わせて 1 つの Web サービスやアプリケーションを構成する手法のことである。元来、利用価値の高い Web サービスを作るためには、独自に検索エンジンや結果応答用のサーバを構築する必要があり、目的の Web サービスを作るために多大な努力をする必要があった。しかし、Mashup では、既存の Web サービスを組み合わせるにより、短期間で価値の高い Web サービスを製作することができる。

### 2.2 WebAPI とは

WebAPI とは、インターネットを介して利用することのできるアプリケーション・プログラミング・インターフェイス (API) のことである。殆どの WebAPI が一般的な URL の形式を取っており、HTTP による POST メソッドを用いて、パラメータを付加した URL を使用してアクセスしてデータを取得する。返ってくるデータは XML、JSON のどちらかが一般的である。今回用いる WebAPI は、以下の 4 つである。

#### 2.2.1 Yahoo!検索 Web API-ウェブ検索 API<sup>1</sup>

開発	ヤフー株式会社
URL	<a href="http://search.yahooapis.jp/PremiumWebSearchService/V1/webSearch">http://search.yahooapis.jp/PremiumWebSearchService/V1/webSearch</a>
機能	Web 上に公開されているページを検索する

#### 2.2.2 Yahoo!検索 Web API-画像検索 API<sup>2</sup>

開発	ヤフー株式会社
URL	<a href="http://search.yahooapis.jp/PremiumImageSearchService/V1/imageSearch">http://search.yahooapis.jp/PremiumImageSearchService/V1/imageSearch</a>
機能	Web 上に公開されている画像を検索する

<sup>1</sup>ウェブ検索 API は、2013 年 3 月頃を目処に API のリクエスト URL が変更される予定であり、これはそれまで公開されていたアップグレード版ウェブ検索 API を使用している。

<sup>2</sup>画像検索 API は、2013 年 3 月頃を目処に API のリクエスト URL が変更される予定であり、これはそれまで公開されていたアップグレード版画像検索 API を使用している。

### 2.2.3 Youtube Data API

開発	Google Inc.
URL	<a href="http://gdata.youtube.com/feeds/api/videos">http://gdata.youtube.com/feeds/api/videos</a>
機能	Youtube の機能（動画の検索、アップロード、再生リストの作成など）を利用する

### 2.2.4 Product Advertising API

開発	Amazon.com, Inc.
URL	<a href="http://ecs.amazonaws.jp/onca/xml">http://ecs.amazonaws.jp/onca/xml</a>
機能	Amazon の商品情報や関連コンテンツを検索する

## 第3章 検索エンジンの精度向上

この章では、キーワードを用いた検索結果の精度向上手法について説明する。

### 3.1 先行研究

当研究室にて行われた研究として、以下の研究がある。

#### 3.1.1 e-learning コンテンツにおけるドキュメントサーチの最適化

Yahoo!Search BOSS API を用いて、有用な Web ページへのヒット率を向上させる実験が行われた。その結果、検索を行う主なキーワードとは別に、HTML ページ中に埋め込まれた `<meta>` タグの `keyword` 属性のパラメータとして特に多いものをサブキーワードとして検索を行う方法にて、有用な Web サイトへのヒット率が向上することが確認された。

### 3.2 サブキーワードの選出

上記研究結果から、e-learning コンテンツへのアクセス精度を高めるため、サブキーワードを選出する方法を考える。

#### 3.2.1 選出過程

サブキーワードを選出する方法に上記の先行研究結果を用いることを試みたのだが、HTML をキャッシュせず、検索を行うたびに毎回数十ページへアクセスを行っていたため、タブレット端末やスマートフォン端末など、通信が不安定になる可能性が高い端末でこれを用いることは難しいと判断した。よって今回は、予めヒット率が向上すると考えられるキーワードを適当に予測、検証し、1～2個程度のサブキーワードを選出した。

#### 3.2.2 選出結果

- 基礎
- 講座

以上の2個をサブキーワードとする。

## 第4章 WWW視覚化

この章では、WWW 視覚化という目線での先行研究や開発事例、そして解決方法についての案を提起する。

## 4.1 開発事例

この分野における開発事例や先行研究は多数存在しているが、中でも 3 次元 CG による視覚化を実現している UNIX ソフトウェア「納豆ビュー」と、mashup による検索エンジンの WWW 視覚化を実現している Web サービス「Flowser.com」について解説する。

#### 4.1.1 納豆ビュー

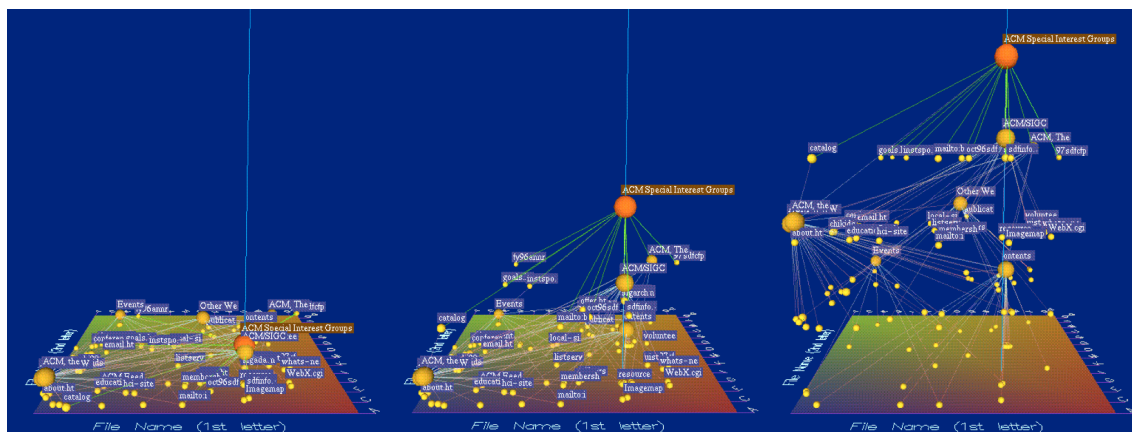


図 4.1: 納豆ビュー

UNIX の X-Window System、Mesa、GLUT を用いており、3 次元 CG グラフ上に展開したノードをリンクに見立て、リンク・被リンクにある関係がエッジで表示される。xy 平面には一意的な座標が与えられ、z 軸方向にはノードを摘んで持ち上げる、つまりユーザによる操作が可能となっている。これにより、複雑なネットワークをユーザーの意志によってわかりやすく可視化できるようになっている。

#### 4.1.2 Flowser.com

Product Advertising API を用いた Web サービスであり、Mashup の作例でもある。検索ボックスに入力したキーワードから Amazon の複数ジャンルの商品を一気に検索・閲覧することができ、通常のサイトを通じた検索では得られなかった情報を届けることを可能にした。



## 4.2 検索エンジンにおける WWW 視覚化

以上2つの開発事例を見たところで、改めて、今回の研究で解決したい問題を整理する。

- タブレットやスマートフォン上から検索結果への直感的な操作  
直感的に結果を閲覧するためには、納豆ビューのようにユーザー自身によって結果をわかりやすく可視化できるようにする工夫が必要になる。また、タブレットやスマートフォンでの閲覧を前提とするため、画面サイズやボタンサイズなど、操作性に対する配慮も必要となる。
- 複数のコンテンツ (検索エンジン) を同時に検索  
e-learning コンテンツは、Web ページだけでなく、動画、画像、PDF などの文書ファイルのように、多数の形式に分かれていることが考えられる。であれば、Flowser.com のように、複数結果をそれぞれ分離して見やすく表示する必要がある。
- e-learning コンテンツへの導線  
コンテンツを検索して終了、ではなく、検索したコンテンツへはダイレクトにアクセス可能にする。ページであればブラウザでの表示を行い、動画であればタップと同時に動画サイトやアプリへ遷移し、再生を開始する必要がある。

以上の条件から、4つの表示モデルを考えた。

### 4.2.1 Helix model

3次元螺旋の円周上に検索結果のコンテンツを配置し、螺旋階段を降りるように検索結果の下位コンテンツへと閲覧していくモデルである。この方法の優れた点として、螺旋階段の中心線からコンテンツを見た時、コンテンツ同士の下位と上位が判断しやすく、ユーザがどのような方向にスワイプしても検索結果を辿れる、つまり上下左右方向への持ち替えが容易であるといった利点がある。しかし、複数コンテンツへの対応を考えた際、螺旋一つでは画面上に対しての情報量に無駄が多く、複数コンテンツへ対応しきれない可能性が高いと考えられる。

### 4.2.2 Star model

複数コンテンツへの配置を最優先に考え、Flowser.com のような円形状のコンテンツ展開を考えたモデルである。コンテンツ毎にそれぞれのノードに分類され、そこから3次元上にランダムにノードが生えており、このノードまでのエッジの長さが検索結果の上位と下位を表している。しかし、3次元上に無作為にノードが存在しているため、コンテンツの一覧性には著しく欠けており、視点を定め辛いことが考えられる。

### 4.2.3 Star-Helix model

Helix model と Star model、両者のメリットをうまく併せて設計したモデルである。螺旋はコンテンツジャンルの数だけ存在し、円形状に展開した始点から同方向に対して一様に伸び

ている。しかし、複数の螺旋を同時に見れる視点の位置が一位に決まらないため、ユーザーが混乱する可能性が高い。

#### **4.2.4 Star-Slide model**

Star model のコンテンツ分離性を活かし、そこから垂直、同方向にコンテンツを配置したモデルである。このモデルを外周から見ると、ちょうど画面上にすべてのコンテンツが入るようになっており、非常に視認性に優れている。加えて、スワイプによって移動する方向も一位であることから、ユーザーが混乱しにくい。また、最前方に配置するコンテンツは丸型ハンガーラックのように回転するようになっており、別のコンテンツをメインに見たい、という時には横方向へのスワイプで自由に切り替えられるようになっている。

### **4.3 採用手法**

以上より、最も多くの問題を改善できた方法として、Star-Slide model の採用を決定した。

## 第5章 開発手法

本章では、本研究の背景、それを踏まえた上での研究の目標・目的、そして文書の構成について述べる。

### 5.1 概観

タブレットやスマートフォンアプリで使える言語としては、Java や Objective-C が挙げられる。しかし、これらの言語では3次元CGを扱う

### 5.2 Away3D

#### 5.2.1 Away3D とは

#### 5.2.2 Android 上への移植



## 第6章 Android アプリ「LEDOXEA」

本章では、本研究の背景、それを踏まえた上での研究の目標・目的、そして文書の構成について述べる。

### 6.1 使用方法

### 6.2 特徴

#### 6.2.1 移植性

#### 6.2.2 スペック性能への非依存性

#### 6.2.3 フリック操作による直感的操作性

#### 6.2.4 検索エンジンの同時検索、WWW 視覚化

## 第7章 アンケートによる評価と考察

本章では、本研究の背景、それを踏まえた上での研究の目標・目的、そして文書の構成について述べる。

### 7.1 評価結果

### 7.2 考察

## 第8章 結論

この章では、結論らしいことをかく。

## 謝辞

本研究を卒業論文として完成させることができたのは、担当して頂いた新井康平教授、Herman Tolle 博士研究員の熱心なご指導や、第 4 研究グループの皆様方に協力して頂いたおかげです。皆様へ心より感謝の気持ちと御礼を申し上げたく、謝辞に代えさせていただきます。

## 参考文献

- [1] 株式会社シートプランニング: 世界のスマートフォン普及予測  
<http://www.seedplanning.co.jp/press/2012/2012072601.html>, 2012年7月26日
- [2] Taro HogeYama, Jiro HogeYama: The Theory of Hoge, *The Proceedings of The Hoge Society*, 2008.
- [3] Advanced Distributed Learning(ADL): SCORM, <http://www.adlnet.gov/capabilities/scorm>, 2004.
- [4] Apple Inc.: iTunes U, <http://www.apple.com/jp/education/itunes-u/>, 2004.
- [5] 文部科学省: 平成十三年文部科学省告示第五十一号（大学設置基準第二十五条第二項の規定に基づく大学が履修させることができる授業等）,  
[http://www.mext.go.jp/b\\_menu/hakusho/nc/k20010330001/k20010330001.html](http://www.mext.go.jp/b_menu/hakusho/nc/k20010330001/k20010330001.html), 2001.

## 付 録A プログラム

実装した ActionScript のソースコード、並びに Android アプリを定義する xml ファイルを掲載する。実装は FlashCS6 を用いて、Android2.2 端末 (IS04) での動作を確認している。

### A.1 main.as(ActionScript)

未完成

ギリギリまで粘ったものを添付する予定です……

### A.2 app.xml(XML)

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<application xmlns="http://ns.adobe.com/air/application/3.2">
  <id>legdoxea</id>
  <versionNumber>1.0.0</versionNumber>
  <filename>Legdoxea</filename>
  <description>3D view e-learning searcher</description>
  <name>Legdoxea</name>
  <copyright></copyright>
  <initialWindow>
    <content>Legdoxea.swf</content>
    <systemChrome>standard</systemChrome>
    <transparent>false</transparent>
    <visible>true</visible>
    <fullScreen>true</fullScreen>
    <autoOrients>false</autoOrients>
    <aspectRatio>portrait</aspectRatio>
    <renderMode>direct</renderMode>
    <depthAndStencil>true</depthAndStencil>
  </initialWindow>
  <customUpdateUI>false</customUpdateUI>
  <allowBrowserInvocation>false</allowBrowserInvocation>
  <icon></icon>
  <android>
    <manifestAdditions><![CDATA[<manifest>
      <uses-permission android:name="android.permission.INTERNET"/>
    </manifest>]]></manifestAdditions>
  </android>
  <versionLabel></versionLabel>
  <supportedLanguages>en ja</supportedLanguages>
</application>
```