
児玉ゼミナール卒業文集

2007 年度 & 2008 年度版

児玉ゼミナール
法政大学経営学部

目次

はじめに (2007 年度卒業生)	7
小澤 楨規	9
活動内容 3 年生	10
コンピュータリテラシー学習	10
夏期 (サブゼミ)	10
夏期 (ゼミ合宿)	10
簿記学習プログラム作成	11
活動内容 4 年生	12
夏期 (サブゼミ)	12
簿記学習プログラム概要	12
デザインパターンへ	13
なぜデザインパターンなのか?	14
神谷 僚	17
3 年次の研究の方向性	18
ロングテール現象とは?	18
Flash との出会い	19
Flash を研究しようと思った経緯	20
理想とする商用素材	20
2006 年の後期	20
Java	21
Java のエディション (J2SE, J2EE, J2ME)	21
Java の配布形態 (JRE, SDK)	22
オブジェクト指向言語	22
UML	22
4 年次の研究方向	23
Applet とは?	24
CGI の弱点	24
PHP の弱点	25
JSP & サーブレットの特徴	25
JSP & サーブレットの弱点	27
卒業発表のテーマ	28
最後に	30
黒崎 将史	31
はじめに	32
歴史	32
パソコン通信	32

インターネットの広がり	32
テキストサイト以前	33
テキストサイト	34
個人ニュースサイト	35
テキストサイト以降	35
Flash	36
ウェブログ	38
坂田 匠	41
背景	42
Java	42
特徴及び概要	42
現在	42
サーバサイドでの利用	43
サーバサイドの Java 技術	43
JSP/Servlet の動作原理	44
HTML を用いた静的なページとの違い	46
Web アプリケーションに Java を使用することのメリット	46
JSP/Servlet を使用した Web アプリケーション開発の問題	46
フレームワーク	47
フレームワークとは(ライブラリとの違い)	47
開発にフレームワークを使用すること利点	48
サーバサイド Java で使用される代表的なフレームワーク	48
Apache Struts (Struts)	48
Spring Framework	54
サーバサイド Java の抱える課題と今後の利用	56
課題	56
今後予測されるサーバサイドでの Java 利用	56
松崎 有樹	59
建物の建設	60
衣服の制作	60
スクリプトの利用	61
商品の販売	62
スカルプテクスチャ	62
Second Life の現状	63
Second Life の強みとは	63
これからのロードマップ	64
最後に	64
松原 正和	67

メタバース(Metaverse)	68
メタバースについて	68
MMORPG と MORPG について	68
MMORPG について	69
MMORPG の特徴	70
三次元(3D)仮想空間について	70
独自の時間軸	70
人間関係	71
経済システム	71
Second Life(SL).....	72
Second Life とは?.....	72
Second Life の統計データ	73
SL のユーザとアバタ人口について	73
SL に参入企業の傾向について	75
SL に対しての一般利用者からのニーズ.....	76
セカンドライフを知っているのか?.....	76
セカンドライフをこれから始める?.....	77
セカンドライフへの興味・関心	79
統計データからみえる事	79
法政大学とセカンドライフ	81
法政大学とセカンドライフ.....	81
企画(コンテンツ)について	81
将来へ	82
はじめに (2008 年度卒業生)	83
児玉 育也	85
Blender にいたるまで.....	86
はじめに	87
Blender を振り返る	87
Blender における 3DCG 制作の基本操作.....	88
Blender によるアニメーションの導入	89
Blender で動画作品を作る	89
まとめ.....	90
田村 俊一	95
はじめに	96
SEO 対策について.....	96
SEO 対策とは	96
SEO 対策のメリット・デメリット.....	97

検索エンジンについて	100
検索エンジン市場とアルゴリズム	100
Google について	102
ブログを利用した SEO 対策の実証実験	105
ブログの SEO 的優位性	105
ブログを利用した SEO 対策の効果実証実験	107
結果と考察	116
結果	116
さくいん	121

はじめに（2007 年度卒業生）

法政大学経営学部児玉ゼミナールは設立以来4年が経ちました。そこで本年度卒業生を中心にゼミナール活動の成果を後輩に残すために文集を作成し、内外に発表することにしました。これまで、年度末に各年度の活動内容をプレゼンテーションという形で発表してきましたが、なかなか活動自体を外部に発表していくことはできませんでした。本年度はプレゼンテーションに加え文集という形で残そうという試みです。

これまでの卒業生も本年度の卒業生に負けなくらい充実した内容の研究を行い卒業して行った学生は多くいます。新しいゲームを作成しインターネット上に公開した学生、アルバイト先のお客さんにアンケートをとり、マーケティングの観点から分析を行った学生、数値データを解析する手法の一つとして時系列データを解析した学生、コンピュータグラフィックスを学び様々な3Dグラフィックスを作成した学生など挙げればキリがありませんが、このような文集として残すことができなかつたのが残念です。今後、ゼミナールの続く限り卒業文集を作成し研究発表を行っていきたいと思っております。

平成20年3月春
児玉靖司

小澤 慎規

経営学部 経営戦略学科 4年J組 04F0911

テーマ “デザインパターンへ”

活動内容3年生

コンピュータリテラシー学習

入ゼミ当初は、研究ネタ探しの前にコンピュータに関する知識をつけるため、大学院で使われる、コンピュータリテラシーの教科書（英語版）の翻訳からスタートしました。約3ヶ月ほどでした。やり終えたとき、特に興味を持った分野はプログラミング言語でした。いくつもあるプログラミング言語の中から私が興味を引かれたのがオブジェクト指向プログラミング言語である Java 言語でした。オブジェクト指向というのは、定義を読むだけではさっぱりでしたが、“要は、ソースコードを再利用してプログラミングを楽にして開発効率を上げましょう”ということだという印象を受けました。これが、私がこの言語を学習しようと思った決定的な理由です。本格的に学習すること6ヶ月が経過してある程度、簡単なアプリケーションを作れるようになりました。私なりに何か役に立つ、意味のあるアプリケーションを実際に作ってみたいと思うようになりました。そこで私は、前々から仕事というのは、やはりどれだけ数をこなしたか？が、理解への近道であると実際に学習して感じていましたが、参考書などでは、問題が不足気味で、新しく問題集を買おうとすると結構な額になってしまい手軽に問題に対して金をかけずに解けるものがあれば便利であると感じていて、簿記の学習ソフトのようなものを作りたいと考えていました。そして、簿記の学習ソフトのようなプログラムを作ることに挑戦しました。ソフト作成についての前に夏期中の活動についてまとめたいと思います。

夏期(サブゼミ)

夏休み中は、サブゼミを数回行いました。基本情報処理試験に出てくるような基本的なことを学ぶ勉強会のようなものでした。一日中で、しかも資料作成が意外と大変でした。どうすれば、スムーズに相手が理解してくれるか悩みました。サブゼミを行った感想としては、非常につらかったです。しかし、考えよりも実際に物事を行うということは大変であるということを実感できたという意味では、非常に有意義なサブゼミだったと思いました。

夏期(ゼミ合宿)

入ゼミして初めての合宿になります。研究テーマは、・・・、正直まだ決まってませんでした。私がパソコンに興味を持ったきっかけである、P2P についてとりあえず調べて発表しようと思いました。そして、初めて人前で、パワーポイントを使ったプレゼンテーションを行いました。反省点としては、やはり見た目にはあまり出て来てなかったと自分では思いますが、緊張していたということだと思います。質問されてもたまたま人の質問内容が理解できないことがありました。それから、こう発表したら、この質問が返ってくるだろうという質問に対する準備も全くできてませんでした。初めてのプレゼンテーションは、あまり出来が

よくなかったと思います。しかし、反省点は見つかったので今のプレゼンテーションに生かされていると思います。

簿記学習プログラム作成

夏期中の活動についての内容が中間に入りましたがいよいよ、本題である、簿記学習プログラムについての話に移りたいと思います。私は、バイト先や、ネットなどからダウンロードしたソフトを使う際、やはりわかりづらい、見づらい、不便だ、というような見栄えがよいもの（GUI）に、こんな感想を持つ場面が多々ありました。例としては、バイト先の出勤退勤システムです。キーボードからの選択でした。正直画面に、ボタンをつけとけば、誤操作しないですし、オーナーもそれを修正するという微妙に手間のかかる作業をしなくてすむのではないかと思いました。自分自身で作るソフトは、とにかくこのような欠点がなく、見やすく使いやすいものを作ろうと思いました。

コンピュータ上で実際にプログラムを実現する際に、一番大切なことは、使う側が使いやすいと感じることだと思います。GUI を作る時特に気を使ったという点は、**視覚的に見やすい、誤操作をしない（誤操作をすることができない）**の2点でした。三年時の活動は、この GUI を実現する作業だけでプログラム言語の学習に時間の大半を割いたため、終了してしまいました。

学習する勘定科目のボタンを押してください **終了する**



1. まず、通常のダイアログボックスでなく、Undecorated ダイアログボックスを使いバーを取り除き、フルスクリーンにすることで誤操作を防ぎました。
2. GUI は、とにかくどんな方でも見やすいように文字やボタンの大きさや色を背景色とは、違う色にし、大きくしました。

これらにより上記で述べた事を実現しました。

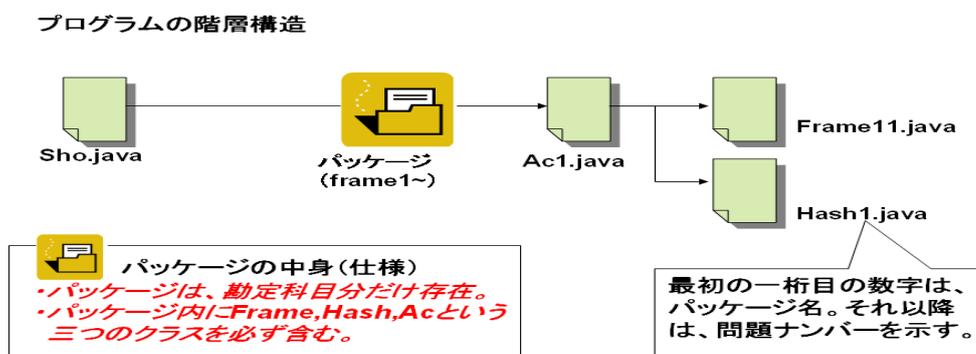
活動内容4年生

夏期(サブゼミ)

2回目にして最後?のサブゼミです。今回は、パソコンの自作(ゼミ専用機)をすることになりました。私もそのころ自宅でファイル共有用の掲示板のためのサーバをノートパソコンで立てました。しかしそのときはハードの知識がまったくなく、または、意識していなくノートパソコンを一つだめにしてしまったという体験をしていたので、パソコンのスペックというものに興味を持ち始めていましたのでちょうど良い機会だと思いました。これを機にパソコンの自作に興味をもち、また、要求を満たすようなものが、自作できるようになることができればと思います。体験した感想としては、1つのパソコンを複数の人たちで作るものではないな(ほとんど経験者の方に任せっきりでした)ということでした。しかし、流れを見ることができたので、おそらく、今度は自分自身だけで作れるのではないかと思います。意外と組み立てるパーツは少なかったです。ただ、パーツの相性というのがあって、指定されたパーツを使っても動かなかったり不具合が発生したりして数万円が一瞬で飛ぶことがあるのが恐ろしいと感じました。

簿記学習プログラム概要

3年生の時は、主にGUIの仕様をどのようにするかということを中心に行い、4年生の時は、実際に簿記学習プログラムのソースコード作成が活動の中心になりました。ここでもやはり、実際にソースコードを見る方々のことを考慮して、つまり、ソースコードを**把握しやすく、書き換えたり**するのが容易であるようにするべきだと思い、この2点に気を使いながら作業を進めてきました。実際のソースコードの全体図(階層構造)は以下になりました(付録1参照)。



ここでは、勘定科目ボタンの選択によりパッケージ1にアクセスしたと仮定してプログラムの全体の流れを説明したいと思います。mainクラスは、Sho.javaであり、このクラスの役割は、2つであり、1つ目はパッケージ内のHash1.javaのインスタンスを生成してHash1.java内のhash1メソッドを呼び出すことです。hash1メソッドは、Frameクラスにより作られ格納された問題(Frame11~)たちをランダムに1つ呼出すメソッドです。2つ目はランダムに取得された問題(Frame)を表示することです。mainクラスで行うことはこれだけです。他のクラスでは、役割は1つずつしか持っていません。ここで言いたいことは、役割ごとにクラスを作ったということです。それにより、**階層構造の把握のしやすさとプログラムの容易な書き換え**を実現しました。いったい何をするプログラムなのかここまでの説明だけでは、わかり辛かったと思います。とにかくこのプログラムは、勘定科目の書かれたボタンを任意で押してもらい、その勘定科目の仕訳の問題がランダムに呼出され、それを解くということを延々と繰り返す、シンプルなプログラムです。全部で100問ほど作ることができました。これを大学内のデジタルコンテンツ2008というイベントに出品してみました。入賞は残念ながらすることができませんでした。これを機に、このプログラムの仕様全体について考えてみることにしました。これから規模が拡大していった時(バージョンアップなど)本当に修正がしやすく拡張性などがあるのだろうかということです。

デザインパターンへ

私は、まず、何を基準に上記の事を考えればよいのか悩みました。正直、ただ実現したい機能を知っている知識だけを使って無理矢理このプログラムを作ってしまったなという感じはしていたので、専門家などの方々はどのようにプログラム開発をするのか?自分とどこがプログラム開発をするとき違うのか?と疑問に持ち、とにかく彼らの開発の仕方を基準に考えたいとはその時漠然と想着っていました。そこで、出会ったのがデザインパターンです。デザインパターンは23個のプログラム開発でよく使う、例えるならビルを建てるための建築方法のようなものです。私は、“Java言語で学ぶデザインパターン入門”という本を読み、この本を基準に上記で述べたことを考えようと思いました。

一通り2ヶ月ほどかけて完璧に理解したとはいえませんが、この本を読み終えた感想としては、同じ機能のメソッドやクラスはなるべく作らない、インターフェースや抽象クラスによる継承、委譲を利用して再利用ができるようなプログラミングをする方が好ましいということでした。私はこれらがデザインパターンの重要な部分だと思いました。これらを基に簿記学習プログラムのソースコード全体を見直してみることにしました。2つほど欠点が出てきました。第一にhashメソッドの機能の重複した記述、第二にそれぞれのクラスは機能ごとに分けて作ったので、完全に独立している状態であるがあまりに独立しすぎていて規模が小さいうちはいいが、大きくなるにつれてFrameクラス、Hashクラス、Acクラス(抽象クラス)をいちいち増やさなくてはならない、つまりソースコードの修正と全体把握がどんどん困難になっていくということでした。これら二つの欠点をとにかく私が本を読んで重要だと思った事柄に当てはめて修正しようと思います(付録3、4参照)。

修正前

呼び出しの流れ

mainクラスであるSho.javaで、問題呼び出す際、Hash1 からHash7まで任意のクラスを生成し、hash1からhash7までのメソッドを使用する。

ソースコードの欠点

- ・Hash1～7クラスにそれぞれ同じ機能のメソッドが、違うメソッド名で記述されていた。
ex)hash1();~hash7();
- ・問題呼び出し方法を新しく作る際にHash1からHash7の各クラス内に問題呼び出し方法を記述した(ここでは、ランダムに問題呼び出す) hash1からhash7メソッドを作ってしまったため、各クラスの修正が必要。規模が大きくなるほど修正と全体のソースコードの把握が困難になる。

修正後(ブリッジパターン)

呼び出しの流れ

mainクラスであるSho.javaで、問題呼び出す際、Callクラスを、引数にHash1~7クラスを渡して生成。問題の呼び出し方法ごとのメソッドを使用。例としてここでは、ランダムに問題呼び出すためのメソッドはhashメソッドである。

ソースコードの改善点

- ・同機能のメソッドの統一。
- ・実装階層と機能階層に明確にクラス分けを行った(付録3参照)。

↓

ブリッジパターンの使用によるソースコードの修正により二つの欠点を解決しました。

第一の修正点は Hash クラス (抽象クラス) を Hash1 から Hash7 のスーパークラスにし、Hash1 から Hash7 では、hash1 から hash7 メソッドを消去し、新たに make メソッドをおき、make メソッドで、ハッシュテーブル (問題のデータベース) のクローンを返すようにしたことです。これにより Call クラス内でクローンを使い、問題の呼出し方法を記述したメソッドを柔軟に作ることができ、無駄なメソッドの記述を避けることができます。この修正により上記で記述した第一のソースコードの欠点である“**hash メソッドの機能の重複した記述**”を解決しました。第二の修正点は、機能階層と実装階層の二つの階層に分け、委譲を利用した緩やかな二つの階層の結びつきにしたことです。実装階層では、Hash1 から Hash7 は、完全に Frame クラスで作られた問題を蓄えるデータベースとしての役割を確立し、またデータベースとして必要なメソッドだけを記述しました。例としてデータベースのクローン (コピー) を作るためのメソッド make です。機能階層では、実際に make メソッドを使い、Call クラス内で、問題の呼出し方法を記述したメソッド (ランダムに問題呼び出すためのメソッド hash) を記述しました。これにより、メソッドの機能修正等を行う際 Call クラスをベースに (継承して) 新たにメソッドを追加するだけで済み、またクラスの関係図全体把握、拡張の見通しもずっと良くなり、第二のソースコードの欠点である“**ソースコードの修正と全体把握がどんどん困難になっていく**”を解決することができました。

なぜデザインパターンなのか?

今回は、デザインパターンにより問題を解決しました。しかしデザインパターンを持ちいらずとも、効率的なプログラムの書き方や解決法はいくらでもあります。ではなぜデザインパターンでプログラムを書いたのか？ソースコードの詳細な説明をする上で、大まかな構造を説明する際、開発者同士の意思疎通のための共通の語彙をデザインパターンが与えてくれるという点です。開発が大規模になれば、すべてを一人で担当することが不可能となります。無限の時間が与えられていれば、大規模なソースコードの理解と開発をすることは、一人または少人数で可能かもしれませんが現実には不可能です。共通の語彙を与えられることにより開発についての複数の人々の間で交わされるコミュニケーションの解釈の誤差を埋めてくれます。それにより開発者同士の誤解から生じる開発の遅延、人件費の増加等を減少させることができます。企業でシステム開発の需要が逼迫する現在ではデザインパターンなる共通の語彙を用いてシステム開発することが納期、人件費の削減に繋がり需要逼迫による受注見送りの減少、売上や利益率の上昇に貢献することになると思われま

神谷 僚

経営学部 経営学科 04F0107 番

ゼミ内での2年間の活動内容

3年次の研究の方向性

3年次の合宿の際、Flashをテーマとして発表した。まず初めになぜ、私が情報系のゼミに入ろうと思ったかについて、次にFlashを研究しようと思った経緯について述べる。これからの日本社会は二つのことが出来ないと社会で通じないと新聞で見たからである。その2点とは英語と情報技術である。私は、英語に関しては中学校から学んでいるが情報技術に関しては大学に入ってから情報関連の授業を履修して初めて興味を持った。私は次のロジックで情報の勉強の仕方を考えた。私は情報のことについて勉強したいと思っていた。情報の授業を履修しようと思った。しかし、私は経営学部にも所属しているので、情報のことについて勉強しても経営学部の意味がないと考えた。

■ では経営と情報の繋がりとは？

→私はWebに魅了される。

以上のロジックでまずWebに関心を持った。私は具体的に次に挙げる部分に関心を寄せた。

- 時間、場所の物理的制約がない
- 知りたい情報をすぐに検索で引き出せるところ
- 一方的ではない（コミュニケーションが取れるところ）

経営学部にも所属している観点から

- 広告を好きにだけ流すことができる点
- 消費者が好きなときに選択できる点
 - 買い手市場の現在の経済と合致
 - 希少なものでも欲しいものが手に入る
(ロングテールの法則)
- 企業と消費者がコミュニケーションできる点
 - 商品改善に繋がる

ロングテール現象とは？

インターネット上での現象は、生起頻度の低い要素の合計が全体に対して無視できない割合を占めるという法則である。少数の上位で全体の大半を占めるという、いわゆる「20:80の法則」に対するアンチテーゼで、ネット上での人々の行動の特徴を表す理論として注目されている。米WIRED誌の編集長だったChris Anderson(クリス・アンダーソン)氏が2004年10月に発表した「the Long Tail」という記事の中で提唱した法則である。

マーケティングなどの分野では伝統的に「上位2割の合計が全体の8割を占める」という経験則が知られてきた。これは「2割の顧客から8割の売り上げ」「2割の社員が8割働く」などの形でよく知られているが、オンラインショップの売り上げなどを調べると、少しずつしか売れない商品の売り上げをすべて合計すると無視できない割合になり、逆に売り上げ上位の少数の製品を合計しても大した割合にはならないことが分かった。この現象をグラフに

描画し、縦軸に生起頻度(オンラインショップで言えば販売数量)をとり、横軸に頻度の高い順に要素を並べると、頻度の低い要素(あまり売れていない商品)の部分が横に長く伸びていることが分かる。これを動物の尻尾に見立て、「ロングテール」(long tail:長い尻尾)と名付けられた。物品の販売だけでなく、オンライン広告やWebページのアクセスなどでも見られる現象と言われている。

一方Webのデメリットは

- 企業の情報が漏れる可能性がある。
- 顧客の情報漏洩で信用が失われる可能性がある。
- ウィルスなどによりデータが破壊される可能性がある

→完璧には防げないがモラル向上や対策ソフトで 防止はできる

続いてWebの利用者数について考える。2003年にはWebの利用者数は7730万になりWebが一般化されている。またブロードバンドも進みWebが高速化し水道、電気、ガスなどと同様にインフラもなっている。更に2004年にはWebの広告費が1814億円になりラジオの広告費1795億円を上回る。この結果、企業としては完全に無視できない存在になった。

Flash との出会い

Flash との出会いについて記す。企業のWebのページを閲覧するとFlashが組み込まれていることが多く興味を持ったからである。企業のWebのページにFlashが組み込まれていることが多い理由は次に理由が考えられる。Webページでの滞在時間を1秒でも延ばしたいからである。Webサイトに入ってから出て行くまでに見ているページ数は平均3.75ページで、1ページを見るのに使っている時間は日本の場合平均35秒である。

続いてFlashの特徴について説明する。ベクター画像が規格の中心で、それにスクリプトで制御することによりマウスの動きに合わせてアニメーションしたり、音を鳴らしたりなど、インタラクティブなサイトを作成するのに向いている。アニメーション、ゲーム、ウェブサイトのナビゲーション、音楽再生などのコンテンツを作るためのソフトウェアである。再生環境への依存度が低く、ベクター画像であるためウインドウサイズを変えても画質が劣化しないという特徴がある。マクロメディア社が開発した動画を扱うための規格及びそれを制作する同社のソフトウェア群の名称である。略称はFlash、フラッシュである。旧称はMacromedia Flashで、マクロメディア社は2005年4月にアドビシステムズ社に買収された。拡張子はSWF, FLVである。

次にFlashのメリット・デメリットを次に挙げる。まずメリットはビットマップ画像ではなくて、ベクター画像であることから、ファイルサイズを軽くできることである。拡大縮小をしても、線がガタつかないこと、動きを持たせることが出来るため、表現力が増すことと自由なインターフェースを構築できることである。一方、Flashのデメリットは検索エンジンに引っかからないし、悪魔でブラウザのプラグインである。TOPページでSKIPボタンがない場合ビジターにとっては悪い印象を与えることである。

Flash を研究しよう思った経緯

まず、Flash のデメリットから述べる。Flash のアニメーションは企業の一方向的な考えになる可能性があるため良くない。これは相手先のブラウザに Flash プレーヤがインストールされていなく見られない、仕事時に見た際音が自動で出て迷惑をかけてしまう、ページが重くなるので閲覧者のフラストレーションが溜まってしまうなどの可能性である。またウィルスソフトを入れていると Flash が遮断される可能性があるため良くない。ではなぜそれでも Flash にこだわるのか。それは消費者に選択できる場所があるからである。具体的に説明すると消費者に合わせたものを作ることができる点、消費者が楽しむことができる点、マウスなどで簡単に操作ができる点がある。その利点を生かし私はユニバーサルデザインの考えで Flash を研究しようと考えた。ユニバーサルデザインの定義とはさまざまな人たちが、いつでも、どこでも、わけへだてなく安心して使える製品を生み出すこと、これが未来をめざす製品作りの基本になる（米国ノースカロライナ州立大学ロン・メイス博士らの呼びかけ）今までは「作る人たちの考え」でモノを生み出してきた。しかし、「ユニバーサルデザイン」の考え方によると、それは「使う人たちの身になってモノを作る」に変わる。使う人の年齢、性別、能力（のうりょく）、経験（けいけん）などの違いに関係なく、同じように使いこなすことができる製品を作っていこうということ。しかも見た目にも美しい、親しみあるデザインに仕上げる。これが「ユニバーサルデザイン」コクヨより引用

http://www.kokuyo.co.jp/eco_ud/ud/aboutud/index.html

では Flash で何を作るのか？使う人の年齢、性別、能力、経験などの違いに関係なく、同じように使いこなすことができるもの。具体的にマウスだけまたは限りなく簡単な様に扱えるものを作る。誰にでも見やすいように文字を大きくする。

理想とする商用素材

私の定義する、商用素材料金がかかるという意味でなく配布などの制限があるということである。Web で動く Flash ではなく HDD（ハードディスクドライブ）に残して動く Flash の素材である。また 商品の説明書を作りたいと考えた。マウスをあてると説明がでてくる。見やすいように文字が大きくなる。

例：車、テレビなどあらゆる商品の詳細を作りたい。

2006年の後期

3年生で共同のテーマで研究することになった。その共同テーマはプログラミング言語を学びながら簿記の学習できるコンテンツを作ることである。なぜ簿記学習なのかというと私たちのゼミはコンピュータを研究するゼミであるが、あくまでも経営学部なので簿記学習のコンテンツにすることに決定した。プログラミング言語は Java、JavaServlet, JSP (Java

Servlet Pages), JavaScript を勉強する 3 組に分かれた。私は Java の勉強 (クライアントの端末の Web ブラウザで動く JavaApplet) 組に属していた。以上の理由から私は Java の特徴について述べる。

Java

- 1、Java は、1995 年頃に Sun Microsystems 社(サンマイクロシステムズ)によって発表されたプログラミング言語である。
- 2、オブジェクト指向プログラミング (詳細は後述) が可能である。
- 3、Java の語源はコーヒーの Java (ジャワ) である。ただし、プログラミング言語の時は「ジャバ」と表記することが多い。英語の発音は「ジャヴァ」である。何故、コーヒーの名前になったかには諸説あり、本当の理由は謎にまつまれている。
- 4、プログラミング言語は、プログラムを逐次解析しながら実行する インタプリタ型言語と、あらかじめマシン語コードに変換しておく コンパイル型言語 に大別される。Java は基本的にはコンパイル型言語であるが、CPU に依存したマシンコードではなく、CPU に依存しない中間コード にコンパイルするのが特徴である。
- 5、CPU に依存したマシンコードは CPU が実行するが、Java のような CPU に依存しない中間コードは、Java の実行環境に含まれるソフトウェアが読み込み、CPU 依存のマシンコードに変換しながら実行する。このソフトウェアを Java VM (Virtual Machine : 仮想マシン) と呼ぶ。
- 6、Write Once, Run Anywhere 「一度書いたら、どこでも動く」。Java では CPU や OS に依存しないプログラミングが可能で、一度書いたプログラムは、Windows や UNIX 系 OS や携帯電話など、様々なプラットフォームで動かすことが可能と言われている。ただし、Java VM のバージョンなどに強く左右され、まだまだ思うようには動かないのが現実である。
- 7、メモリリークの可能性が少ない メモリリークのミスが少なくなるように設計され、確保されたメモリは、誰からも参照されなくなると自動的に開放される (ガベージコレクション)。

Java のエディション (J2SE, J2EE, J2ME)

Java 2 には複数のエディションがある。個人向けデスクトップなどで使用されるのが J2SE、企業向けサーバなどで使用されるのが J2EE であり、組み込み系で使用されるのが J2ME である。

J2SE(Java 2 Platform, Standard Edition)

→ 標準エディション

J2EE(Java 2 Platform, Enterprise Edition)

→ 企業向けエディション (Servlet, JSP, EJB ...)

J2ME(Java 2 Platform, Micro Edition)

→ 組み込み向けエディション

Java の配布形態 (JRE, SDK)

J2SE は、下記の 2つの形態で配布されている。Java のソフトウェアを動かすには JRE が、開発するには SDK が必要である。SDK は Java 1.1 までは JDK (Java Development Kit) と呼ばれていた。

JRE(Java Runtime Environment)

→ Java の実行環境のみ。

SDK(Software Development Kit)

→ Java の開発環境、JRE を含む。

オブジェクト指向言語

ソフトウェアの設計や開発において、操作手順よりも操作対象に重点を置く考え方である。関連するデータの集合と、それに対する手続き(メソッド)を「オブジェクト」と呼ばれる一つのまとまりとして管理し、その組み合わせによってソフトウェアを構築する。

すでに存在するオブジェクトについては、利用に際してその内部構造や動作原理の詳細を知る必要はなく、外部からメッセージを送れば機能するため、特に大規模なソフトウェア開発において有効な考え方であるとされている。データやその集合を現実世界の「モノ」になぞらえた考え方であることから、「オブジェクト」指向と呼ばれる。

個々の操作対象に対して固有の操作方法を設定することで、その内部動作の詳細を覆い隠し、利用しやすくしようとする考え方と言える。この考え方を応用したプログラミング技法が、オブジェクト指向プログラミング(OOP)である。

UML

UML は Unified Modeling Language の略で、オブジェクト指向分析、設計においてシステムをモデル化する際の記法(図法)を規定した言語(ビジュアル・ランゲージ)である。UML は現在のソフトウェア開発において事実上の標準となっており、これを学ぶことは開発者として必須要件になってきた。では、何のために UML で図を書くのか? 以下の理由が考えられる。

レベル 1: プログラムを書く前に、図で自分の考えを整理する。

レベル 2: チーム開発において、図でコミュニケーションする。

レベル 3: ユーザや顧客と仕様を検討する。

それぞれのレベルでの使い方を簡単に説明する。

■ レベル1

プログラミングを始めてしばらくすると、動くプログラムを書くことがシステム開発だと誤解することがある。完全な間違いではないが、仮に一人で書いていたとしても、プログラムの規模が3,000行を超えたあたりで壁にぶち当たるはずで行き当たりばったりでコーディングしていたのでは見通しがたたず、つぎはぎだらけのプログラムになってしまう。これを乗り越えるには、自分の考えを整理することが必要になる。プログラムだけを見ていたのでは、どうしても視点が小さくなりがちで、このようなとき、プログラムをUMLで図解してしてみる。UMLはプログラムの構造や動きを図でとらえることができるため、見通しよく全体を目で捉えることができる。これが、UMLの利用法レベル1である。

■ レベル2

さらに、プログラムの規模が30,000行を超えると一人で開発するのが難しくなる。チームを組んで開発することになるが、その際には分担を決めたり、システム全体の構造（アーキテクチャ）を決めたりする必要がある。職業としてシステムを開発する、いわゆる「プロ」の現場では、これが通常の姿であろう。

その際に、設計を検討するための図としてUMLを使うことができる。数人でシステムの設計を討議することがよくある。ホワイトボードなどを使って絵を描く際にもUMLは役立ち全員がUMLの語彙を持っていれば、簡潔かつビジュアルに設計の方針を示すことができる。あるいは、UMLの図をきれいに効率良く描くことができる市販のCASEツールがいくつか存在する。これらを使ってもかまわない。これが、UMLの利用法レベル2である。

■ レベル3

最後に、これがもっとも大切だが、システム開発とは「プログラムを正しく書く」ことではなく、「正しいプログラムを書く」ことである。プログラムが正しいかどうかを決めるのは、そのシステムのユーザやそのシステムに対価を支払う人、すなわち顧客である。だから、システム開発では開発者がユーザや顧客と対話をして、どういうシステムを作るのかを合意する必要がある。この場面でも、UMLは活躍し、ユーザや顧客は一般的にシステムを利用する機能に興味があり、プログラムの内部構造には興味はない。UMLではシステムがどういう機能を持つのかをも表現することができる。UMLで機能を記述し、それをもってユーザや顧客と話し合い、システムの仕様を決める。これが、UMLの利用法レベル3である。

4年次の研究方向

4年生になり私はクライアントの端末のWebブラウザで動くJavaApplet（以後Applet）からサーバ側サイトで動くJavaプログラムのサーブレット（Servlet）、JSP（Java Server Pages）に興味を持った。私がAppletからサーブレット、JSPに興味を移行した理由を以下に挙げる。

- 1、同学年である坂田匠さんの発表を聞いていてサーバ側の Java プログラムに興味をもったからである。
- 2、特に JSP は HTML 形式で Java プログラムを書け、作成できる点である。
- 3、Applet より動的なプログラムが作れると考えたからである。

3点目の動的なプログラムである「動的」の根拠が曖昧であると発表後に指摘されたので夏合宿の際に発表した。その発表の内容は以下の通りである。

Applet とは？

次に Applet について説明する。Applet には次のような特徴がある。

- 1、クライアントサイドプログラムでありブラウザのウィンドウに埋め込まれて実行される Java プログラムのことである。
 - 2、コンパイルした Java クラスファイルを埋め込む。
 - 3、Web ブラウザのみで利用することができるネットゲームやチャットを開発できる。
- 1点目に挙げてある様に Applet はクライアントサイドプログラムである。しかし次のような問題点がある。

1、情報の加工ができない

データの見せ方は動的ではあるがデータ自体は悪魔でも静的なものである。

2、ロジックの複雑さがページの重さに比例する

クライアントが必要なデータを取捨選択できなということ。構造的にデータ処理を想定して設計されていない。

3、クライアント環境に依存する

クライアント環境のデバイスに応じて異なるプログラミングを余儀なくされる。適用範囲も局所化されてしまう。

4、プログラムやデータが露出してしまふ

ではクライアントサイドプログラムの反対側にあるサーバーサイドプログラミングにはどのような種類、特徴があるのかを紹介する。代表的なサーバーサイドプログラミングには CGI、PHP、JSP&サーブレットが挙げられる。なぜ、CGI、PHP、JSP&サーブレットの中から私は CGI、PHP、JSP&サーブレットを選んだのか。理由は CGI、PHP には以下の様な弱点があり JSP&サーブレットには以下のような強みがあるからである。では CGI、PHP の弱点、JSP&サーブレットの強みについて述べる。

CGI の弱点

1、サーバーリソースの消費が激しい

CGI ではユーザのリクエスト毎に新たなプログラム実行のプロセスを生成する必要がある。

2、Web プログラミングに特化した仕組みではない

CGI に対応する Perl や C 言語はプログラミングする際に常に Web 特有のプロトコルを意識しなければならない

PHP の弱点

1、Java 言語ほど可搬性がない

UNIX系OSの受け継いでいること

2、パフォーマンスの問題

PHPはインタプリタ（逐次翻訳）型の言語のため動的なコードが要求される度に解析・コンパイルされる必要がある

JSP&サーブレットの特徴

1、Java 言語ベースの技術

JSP→HTMLベース・プレゼンテーション主体

HTMLの中に断片的なJavaプログラムを埋め込む形態

サーブレット→クラスベースロジック主体

Javaプログラムをベースに処理の一環として

HTMLを出力する。図1参照

下記の特徴は6ページ目に記したJavaの特徴と同意である。

2、コンパイル言語である

JSP&サーブレットでは処理に先立ち、コンパイル処理（一括翻訳）しコンパイル後のバイトコードはメモリ上に常駐するので、2度目以降の処理にあつたてはプログラムの開始に伴うオーバーヘッドもほとんどなく、高いパフォーマンスを望める（図2参照）。

3、豊富なクラスライブラリーを利用可能

JSP&サーブレットでは2500を超えるJava標準API（※注1）を利用することができ、Java自体がネットワークを強く意識した言語であることから標準API的にもそれに見合ったラインナップが揃っている。

※注1 Application Program Interface の略。アプリケーションから利用できる、オペレーティングシステムやプログラミング言語で用意されたライブラリなどの機能の入り口となるもの

4、プラットフォームを選ばない

March 8, 2009

UNIX系OSでもWindowsOSでも、実行エンジン（コンテナ）さえ対応すればどこでも動作することができる。

5、各種データベースへのアクセスが容易

フリーのMySQLや商用のOracle、汎用的なデータベースインターフェースであるODBCなど主要なデータベース製品へのアクセス手段を標準に備えている。

図 1 JSP&サーブレットのソースプログラムの違い

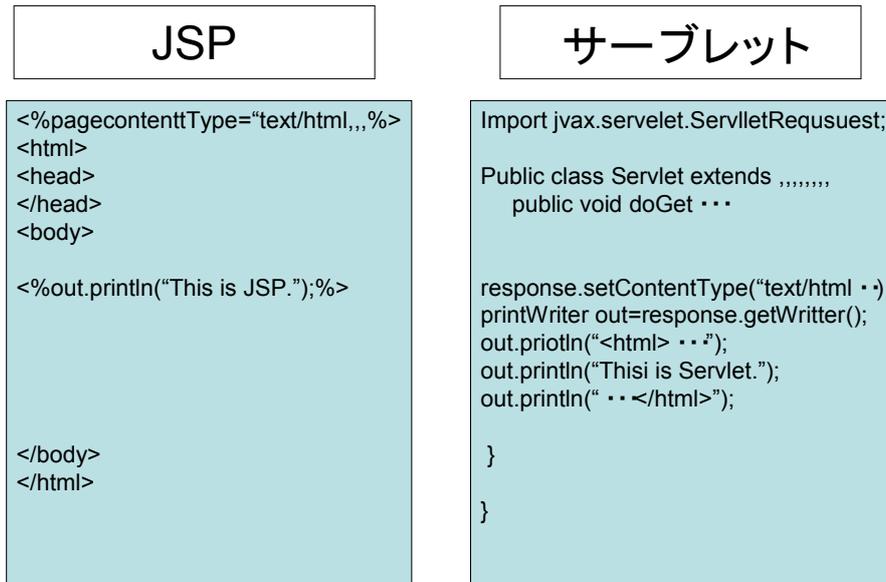
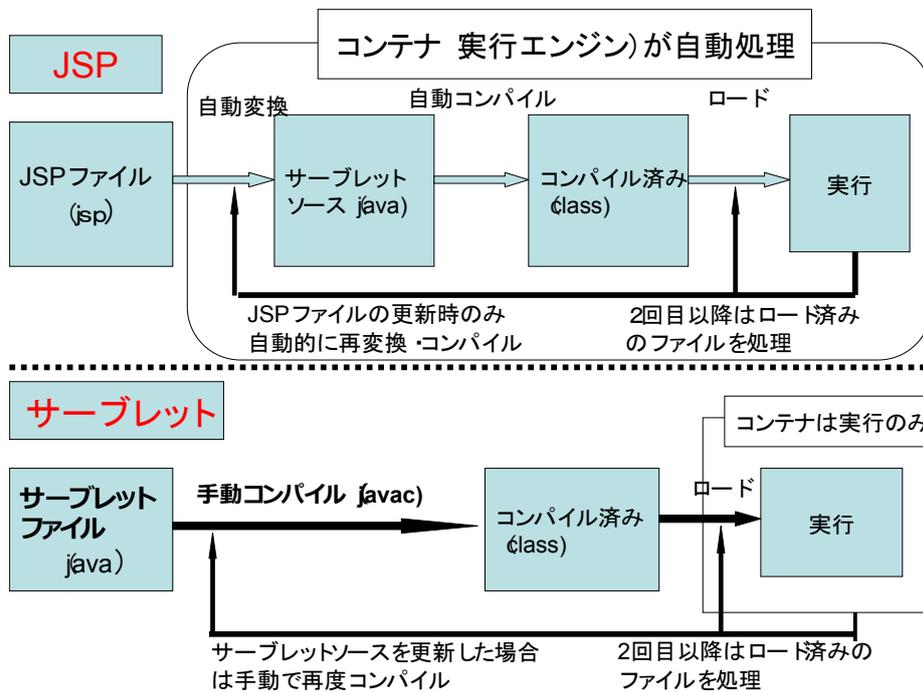


図 2 JSP&サーブレット実行手順



JSP&サーブレットの弱点

JSP&サーブレットの弱点をそれぞれ説明する。まずJSPの弱点だがJSPはHTML形式でプログラムを記述するため膨大で複雑なプログラムを記述する際パフォーマンスが落ちてしまう。

次にメールアドレスの簡単な構造を以下の図4に述べる。

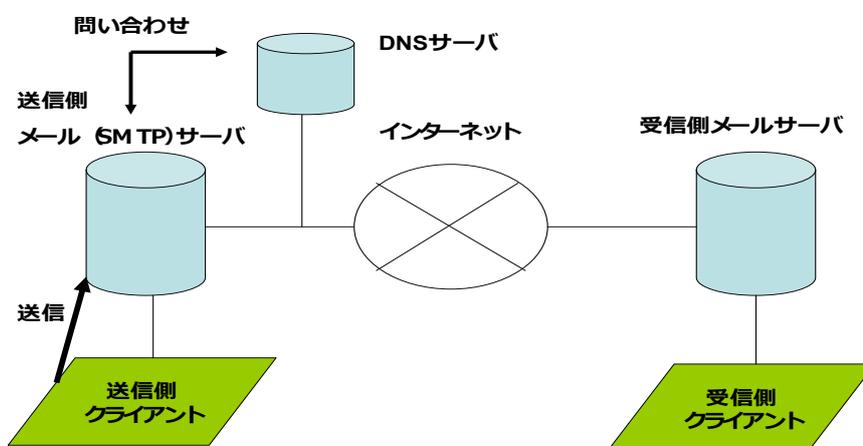
図4 メールアドレスの構造



図5 簡単なメールの送受信の仕組み

@マーク以前はアカウント名で任意の文字列で構成され、@以降はドメイン名（領域）を表す。続いて簡単なメールの送受信の仕組みを以下の図5に表す。

- 1、まず送信側クライアントが送信側のメールサーバ（以下 SMTP サーバ）に送信する。
- 2、次に SMTP サーバは送信側クライアントから送信された送信先のメールアドレス部分のドメインを DNS（Domain Name Systems）サーバに相手先の IP アドレスを問い合わせる。
- 3、SMTP サーバは DNS サーバで問い合わせた送信先のメールアドレスと照合し IP アドレスを照合し送信する。



- 4、SMTP サーバはから送信された電子メッセージはインターネットを経由し相手の受信側のメールサーバ（以下 POP サーバ）に送信される。

5、受信側クライアントはPOPサーバにメッセージが来ているか問い合わせ、ダウンロードする。

SMTP (Simple Mail Transfer Protocol) とはインターネットやイントラネットで電子メールを送信するためのプロトコル

POP (Post Office Protocol) とはインターネットやイントラネット上で、電子メールするためのプロトコル。

最後に

メールは便利ではある反面、他人のSMTPサーバを経由して簡単に送信することができるため危険性も高い。その危険性を防ぐためにSMTPサーバとユーザとの間でユーザアカウントとパスワードの認証を行い、認証された場合のみメールの送信を許可するSMTP Authenticationを設定し、関係の無い第三者が自由に、電子メール送信に用いる事が可能なメールサーバである第三者中継を禁止する設定を行うことが必要である。

何事にもメリット・デメリットがあるが、メールは特に危険性が高いことが分かった。しかし利便性も高いのでメールがなくなることはこれからはないと考えられる。私は、いつでも、どこでも、誰とでもコミュニケーションがとれる時代(ユビキタス)が近い将来、来ることを期待し更に人間同士がコミュニケーションをとりやすいメール以上の技術の革新を期待している。

参考文献・参考サイト

今日からつかえる JSP&サーブレット サンプル集 JSP2.0+サーブレット 2.4 対応版
秀和システム 山田 祥寛 (著)

IT用語辞典 e-Words <http://e-words.jp/>

wikipedia <http://ja.wikipedia.org/wiki>

黒崎 将史

経営学部 市場経営学科

無題

はじめに

近年 PC の普及率は 70% を超え、各世帯で PC が 2 つ以上あることも珍しくなくなってきた。それにつれインターネット普及率も 50% を超え、インターネットは新しい技術ではなくもはや当たり前のものとなってきている。利用するだけでなく個人が情報を発信するという行為も活発になってきている。そこで日本にインターネットが導入されてから現在までのいわゆる個人サイトに焦点を当て考察していく。

歴史

パソコン通信

インターネットが登場する以前に「パソコン通信」というサービスがあった。大きな違いはパソコン通信には中心があるということだろう。ホストといわれるコンピュータに掲示板、メールやゲームなどのデータがあり、利用者はホストに接続することによってそのネットワークに参加できる。

パソコン通信は戦後、日本電信電話公社が民営化されると共に公衆電気通信法が法改正された。電気通信事業法により通信の自由化がなされ、端子がモジュラージャックにされた。これによりモデムをつなぐことも可能になりパソコン通信普及に一役買った。

しかし大手商用ホスト（ニフティサーブやアスキーネットなど）とは別に個人が運営する草の根 BBS 増えるが通信回線が一本のところも少なくなき、人の少ない深夜が中心となり、複数の人が一遍に接続すると回線がパンクしてしまうなど、まだまだ個人には贅沢なことだった。それゆえ遠くのホストにかけると電話代がかかり物理的に近いホストと接続し地域に根ざしたコミュニティが形成されていった。初期のころは実験や研究的な場としてだったが、法改正後に登場した商用パソコン通信により BBS の内容も技術一辺倒だったものから、知らない人とのコミュニケーションや同じ趣味の集まりなどそれぞれの特色を帯びていくようになり、いままで「全員ほぼ知り合い」から「全員ほぼ他人」へと変化していく。

こうして利用者数を拡大させ盛り上がりを見せたパソコン通信だったが、90 年代中盤にもなると商用パソコン通信各社がインターネットへの対応を始め、インターネットの爆発的なブームによりパソコン通信は徐々に影を潜め、2006 年最後に残ったニフティが全サービスを終了させその歴史に幕を閉じた。

インターネットの広がり

インターネットは 1969 年にアメリカの情報処理研究所から生まれた、大学、研究所間をつなぐ「ARPANET」が起源といえる。後に USENET や CSNET が生まれるが、ARPANET のプロトコル NCP から TCP/IP に切り替わり CSNET と相互接続したことにより最初の大規模ネットワークが誕生しインターネットの産声が上がった。

一方日本でもそのネットワークに繋がっていたが、日本インターネットの礎ともいえるのが村井純が作った「JUNET」である。これにより各大学、企業と徐々に数を増やしていきUSENETやCSNETに接続するところもあられ国際的にも広がっていった。そして松下やアスキー、ソニーなどが商用プロバイトを作る「WIDEプロジェクト」がたちあがりTCP/IPなどの統一のネットワークを作る、そしてパソコン通信業者が移行し始めるなど日本にもその根が徐々に広がっていった。

日本でのインターネットが誕生したが、それでは個人サイトはいったいどうなのだろう？ インターネットが誕生したとはいえ、まだプロバイダ料金は高く個人の趣味で利用できることは難しかった。しかしその中にも例外はいた。大学生である。大学では当時の最先端の設備があり回線も太くインフラは整っていた。そこに余剰スペースを学生に分け個人サイト文化が始まった。といっても最初のうちは手探りであり自己紹介程度のものだった。

しかしそこに変化が現れる。インターネットが登場する以前、自分の考えを伝える手段として同人誌、ミニコミといったような紙媒体が主流だった。そこにDTPが合わさりパソコン上で作られた記事をインターネットで公表する人が出始めた。いわゆる「e-zine」の始まりである。ここに表現の場を模索していた大学生が飛びついた。その代表が九州工業大の広告研究会の「Japan Edge」である。そのコンテンツとして研究と関係なく個人的なものでテクノ系の紹介やライブ情報、町で見かけたオシャレな人や投稿レビューなど、今考えるとすごい充実していた。まだ産声を上げたばかりで何をしていたかわからなかった人たちにこのサイトに人々が集まって行ったのもうなずける話である。それに加え早くから訪問用のゲストブックを設置したいたことにも注目したい。これは自動登録リンク集で、閲覧者はここを起点として面白いサイトへ飛ぶ。そこで個人サイトのコミュニティが形成されていった（もちろんメーリングリストの存在は忘れてはいけない）。もちろん「Japan Edge」以外にも注目すべきサイトはある。伊藤が開設した「富ヶ谷」である。このサイトは有名人にスペースを分け与えたりしていたが、重要なのは「Netsurf Team」というメーリングリストである。ここに掲載されることが「Japan Edge」とともに面白いサイトの条件だとされていた。その後アングラ系日記系など数々の個性的なサイトが生まれた。その活動は外にも広がりイベントを開いたり学園祭に参加したりとしていた。しかし、管理人が大学生だったこともあり卒業と共に「e-zine」は収縮していった。その後「SHIFT」など有力なサイトは出てきたが今までのようなコンテンツ志向ではなくデザイナー文化などの方向へ吸収されていった。

テキストサイト以前

ウェブ日記言っても確かな定義があるわけではなく、日付と内容をセットで更新する事位である。しかし一番重要なのは続けることである。このことは発信する側、受信する側両方に影響を与えた。まず発信する側だが、読者に飽きさせぬよう日々新しいものを求め感受性が増していく、新しい視点が増えるなど日々の影響に影響を与え新しい自分を発見することもある。一方読者のほうはたまたま訪問した日記に共感したり興味を引かれそれに没頭して

いく。そしてそれに満足せずもっと似た別の日記を探し始め自分のアンテナが広がっていく。こちらも今まで自分のなかったものにふれ新しい自分に出会うこともある。そんなわけで E-zine 以降に個人の日記とした文化が花開いていった。もっとも日記を書くということはコンテンツに困って始めた面もあるのだが…。

しかしなぜこのような形式になっていったのか。その起源をたどっていくと「週間黒田」にたどり着く。その内容は新聞やゲームの話題、日々の雑多な話題にコメントをつけるというものだった。いわゆる情報サイトに日記をつけたようなものだった。そのことは当時先見的でありこの世代に多大な影響を与えた。

しかし文化までに成長させた中心はほかにある「日記リンクス」である。基本的には日記サイトのリンク集だが CGI を使い自分で登録できるといった点が人気だった。徐々に数を増やしていき、そこにランキングが加わることにより外から人がやい内部で競いあい発展していった。またこの時代「ハイパーダイアリー」というものも登場した。これは HTML の日記の文中に他日記のリンクを直接張り相互強く結びついていく、一種のコミュニティに育っていった。ただこのような事は元々行われていたが現象として一気に広まっていった。その後言葉自体はなくなっていったが形式は今でも生き残っている。

さてウェブ日記自体も 95 年に発売された Windows95 によりインターネットが注目されパソコン通信各所もインターネットに移行し始めたため多大な初心者が参入してきて多く流れが変わっていった。そのなかで「アンテナ」というものが出始めた。複数のサイトの更新時刻を自動でチェックするリンク集である。このころにウェブ日記とは違う「テキストサイト」と後に呼ばれるサイト、そしてそれと強く結びつくランキングサイト「Read ME」がうまれる。

テキストサイト

さて多くの新規が入ってことで混沌としたネット社会だが、日記サイトも変わってきている。96 年以降の個人サイトはい以前のものとは流れが途切れている。では今までのものどどのように変わって言ったのか。まず多くの初心者が加わったことである程度の輪郭が崩れ、その内容も大学での身内向けから広く世間一般相手向けが多くなっていった。つまり身内で共感できる批評なり日記から不特定多数向けの形は日記と呼ばれるが内容は別のものへと変容していった。そこの受け皿としてできたランキングリンク集が「Read ME」である。当初は何でもありの形式をとっていたこともあり以前のようなサイト同士の関係性というもののは薄れていき、リンク集のコミュニティとしての独自性というものはなくなってしまったが、なんでもありという新しいコミュニティが形成されていった。そこにパソコン通信からやってきたものの中で、ショートストーリーやエッセイといった短い文章、いわゆる「雑文」という概念が持ち込まれた。その特徴としてきちんとした起承転結や高い文章力である。これに刺激されたものたちが今までの日記から内容や構成を意識した「雑文」を取り入れていくようになっていく。このようにして広がりを持っていた雑文はそれぞれ同じようなコミュ

ニティを作り分離していったが、その礎を作った README のことは忘れてはいけない。その後は社会現象を起こしたエヴァンゲリオンを考察する旨としたサイトや俗に渋谷系といわれるジャンルの盛り上がりの影響などサブカル的な娯楽文章サイトが登場していき、雑文もその個々の特色を強めていった。その後「shift」がオタク系とおしゃれ系をつなぎ合わせ、今のようなデザインはおしゃれだが内容はオタクといったフォーマットが確立される。そのなかでオタクと親和性が高いゲーム批評で語り継がれている

「FunnyGasmer's Heaven」の「背景黒＋文章中央揃え＋強調文字サイズ大」といったようなフォントいじりは後に数々のサイトに引きづかれており個々を期限と見る人も多い。

個人ニュースサイト

インターネットが始まって初期の頃からあるジャンルとして日本のブログの元祖といわれる「ニュースサイト」がある。海外の情報や企業の情報紹介するサイトのことである。そのなかで個人ニュースサイトというジャンルが出てきた。これは個人といっても複数で管理している、他サイトにリンクを張るだけで独自のソースが無いなどその実態をつかみにくい。

その出発点は商用ニュースサイト（新聞各社など）が取り上げないネタを扱うものだった。例えばネット初期の mac ユーザによるバージョンアップやバグの報告である。

ただしこのようなものは「窓の杜」のように個人から企業へスポンサーを変え、商用サイトでも情報が充実してくると個人サイトはその質や内容にかなわずリンクを張るだけに留まるようになっていき、体験レポートやお勧めにとどまるようになっていった。またリンクを張るという対象がネット上の面白サイトや事件にも広がっていき、それを日々繰り返す内に出来事や話題が時系列に網羅されるニュースサイトに変化していった。その後徐々にアングラ系ならアングラゲームならゲームとすみわけられていったが、基本的には取り上げる内容はひとつもジャンルのとらわれず幅広くなっていく。つまり気になったものを片っ端から紹介する総合型とひとつのジャンルに絞った専門型に分かれていくということになる。

テキストサイト以降

テキストサイトは管理人同士のオフ会やイベントでコミュニティを強化していった中、新たに登場する数も数え切れないほどとなっていた。そのなかで「日記系」というものがある。いままで何かしら特徴があったサイトが多かったがコンテンツが日記中心という素朴な感じで一般の人がやっている印象を受ける。その中で、日常で起こった信じられないことや扱うネタが自虐的な自虐系などの～形などが生まれ多様化していった。

そんな中衝撃を与えたサイトが登場する。「侍魂」である。これはのちにフォントいじり形と呼ばれるジャンルを生み出すきっかけとなったサイトである。もともとそのようなテクニックは各所で見られていたが、異常な人気振りをみせ（管理人本人も異常だったと供述する）普段テキストサイトなど覗かないような人にも火をつけ一大ムーブメントを巻き起こした。ニュースサイト系で突如現れたものとして「ちゅ12歳」があげられる・このサイトは

ちゅというバーチャルネットアイドルを作り出し、二次元の架空のアイドルが作りだされた。このアイドルと扱う内容のマッチがあいまってその後多くの類似サイトを生み出した。

また01年は個人ニュースブームだった。そのせいかさまざまなジャンルが増えては消えていったが異文化交流やニュースサイトにリンクを張る孫ニュースサイトが誕生し、もはや一括りにできなくなっていき混沌としていった。そこで新たに文章以外のコンテンツでもぬきんでたものが登場した。海外からの画像中心の「Pya!」といった画像中心の収集サイトが出始めてきた。

さてここでニュースサイトをもう一度捉えなおすと全体の傾向として個人ニュースサイトをコミュニティと捉える動きが出てきた。何かを発信しようとするよりも面白そうなものに触れたい参加したいといったコミュニケーション重視のユーザが増加していった。コレはニュースサイトだけではなくほかのジャンルも同様だった。これからサイト単体で完結するものから他サイトと繋がることで成り立つニュースコミュニティが成立していった。

そこで「侍魂」以降は自分たちの居場所の再確認が行われた。テキストサイト論や批評サイトが流行したのもそういった動きからであろう。

また2ちゃんねるネットウォッチ板での日記系サイトへの陰口スレットが拡大しサイト同士の口論に発展することもあり、サイト内容よりも管理人の性格やバトルなど人間関係にも注目するものの見方が生まれるのもこの時期である。

しかしすべてのサイトが「侍魂」に影響を受けたわけではなく独自のスタイルを貫いていったサイトも多い。その中で注目を集めたのが絵日記サイトである。テキストに画像を添える時とイラスト中心の漫画的表現のときなどがある。画像を使う方法はいままでにあまりなかった、それは画像のダウンロードにストレスを感じなくなった通信環境の向上やテキストサイトの表現のひとつとして進化していったともいえ表現の多様化に花を添えた。

Flash

Flashの作り手として今は、プロやセミプロが活動の一環として手がける場合、サイト製作会社やプログラマが新しい技術として取り入れる場合、最後にそれ以外の個人が新しい動画作品として活用した場合である。

2番目の例は音の出るメニューとしてだけでアクセシビリティもユーザビリティも無視した使い方である作品というには程遠い。注目すべきは最初と最後の事例だが近年盛り上がりを見せているのは個人の方である。そもそもFlashが注目されたのは良くも悪くも一発芸的なものであったからだ。最近はその以上に水準の高いプロにも引けを取らない作品が数多く生まれているのはいうまでもないことだろう。

さてFlash以前にはどのような受け皿があったのだろう。それまではShockWaveやQuickTimeなどの作品が主流だったがファイルサイズなどで不満があった。その次の主流はGifアニメである。Gifもはじめは味気ないものにアクセントを与える程度のアニメ機能を使ったアイコン画像素材として出回った。さてFlashに繋がる作品はGifで生まれたのだ

ろうか。その点で重要なサイトは「いかそうめんファンクラブオフィシャルホームページ」である。この「世紀末Gifアニメーションシアター」で公開された「ピカプー」というアニメが大反響を呼んだ。コレ以前にも同じようなサイトは存在していたかもしれないが、多くのサイトに転載されたり偽者が誕生したりというインパクトは過去にはなかった。この流行はピカプーみたいなアニメを「作りたい」「もっと見たい」という需要と供給を生み出すほどの勢いを増した。それはやがて初期Gifアニメの中心となってくる投稿掲示板サイト日本Gifアニメ連合へと繋がっていく。さらにその輪を広げていった。

しかしFlashにつながる前にひとつの足がかりがあった。それは90年代初頭のマルチメディアブームである。コレはCD-ROMを媒体としたもので「Macromedia Director」で製作されていた。Directorは動画アニメーション編集の代表的なソフトであった。クリエイターはやがてCDからWWWに移そうとしていたが公開に問題があった。まず当時のブラウザ(MosaicやNetscape)ではDirectorで作った動画を表示することができずいったんダウンロードをしてから再生ソフトを別途立ち上げ投げればならず大変不便だった。そこでMacromediaが開発したのがShockWaveである。これによりブラウザ上でも表示できるようになった。しかしそれですべてが解決したわけではなく容量の大きさの問題が残っていた。このころはまだ回線が細く定額料金制などなかった時代だったので、もしコストをかけて落とす動画がつまらなかったらという意識が働いて、なかなか利用者が増えなかった。そこでインターネット動画のシェアを増やしたいMacromediaは「Future Splash Animation」を持っていたFutureWaveSoftwareをソフトごと買収した。容量が小さくすむアニメーション作成ソフトが手に入り「Macromedia Flash」の名で販売した。これがFlashの始まりである。アクションスクリプトの強化やMP3への対応など機能が充実したFlash4あたりでウェブと相性のいいアニメーション作成ツールとしての地位を不動のものとする。そして今後インタラクティブやモバイル方面にも力を入れていくことが予想される。

さて日本最古のFlashコミュニティはmad-cDesignの笠井氏が作ったflash-m1である。初心者から上級者まで日本中の作者が集まった一大コミュニティとなった。03年にサーバー移転で発展的に解消したが日本Flash文化に果たした役割は大きいものだった。

このあとFlashがトピックとなる時2ちゃんねるのFlash板とかかわってくるが多くなる。そのくらい盛り上がりを見せた。最初のほうは慣れ親しんだAAキャラクターが動くことに感動を覚えていたが、海外からの「h a t t e n」というものが始めてから流れが変わった。これは海外の曲をFlash化したもので変な音楽と奇妙な映像があいまって何かよくわからないけど「面白いFlash」として瞬く間に広まっていった。このあとも2ちゃんねるに新たな海外のFlashが発掘されていった。その後日本人製作者も出始めこちらも盛り上がりを見せた。あるとき「吉野家コピペ」で有名なポエ山氏のサイトにおいてある「quino」というFlashをついでに見た人たちが今までのお笑いのものと違うシリアスで壮大なアニメーションがFlashでもできるという可能性に気づいた。それからいわゆる感動系Flashが出回り始めFlash-m1的な文化と2チャンネル的なぶんかが混ざり合い。新たなシーンへと移っていった。その後2ちゃんねる寝るないで騒動がおきたとき支援でFLASHが作られるようになり新たな才能が生まれていった。

02年は新たなジャンルが生まれいろいろな試行錯誤が生まれその盛り上がりは年末の「第一回紅白FLASH合戦」へと結合していった。ここで無名の新人が一気に上り詰めたことから、最近では登竜門的な認識になっている。これとは別に数々のイベントが起こりまた作者にファンがつくなどもはやひとつの文化としてflashはウェブに定着していった。いままでテキスト中心だったものから映像や音楽を絡めた新しい表現方法ができていったのだった。

ウェブログ

02年11月から日本でウェブログという言葉が急速に普及していったが、その始まりは海外サイトの「robot wisdom」のジョンバーガー氏が97年12月に始めたものである。99年に同種のコンテンツを持つコミュニティが形成された後無料ホスティングサービス「BLOGGER」を契機に広まっていった。

9.11同時多発テロが起きたときには大手マスコミが伝えきれない個人の声を発信する場として大いに注目を浴びた。日本にこの文化が輸入されたのは00年ごろだがもちろん似たようなものはすでにあっただしそれは海外でも同じことではある。ただ言葉が作られたことをきっかけに新しいコミュニティが形成されていったのは事実であった。

昔から似たようなものがあつたとすればウェブログの定義とは何なのだろうか。それはウェブ上で記事を出版するのに適したサイト運営スタイルで一日一回位更新しほかのサイトへのリンクも多い、とこの程度のものでいい。今日本ではやっているRSSやトラクバック云々はただのツールであり本質は表してはいない。しかし使っているうちに進化をしていくのがツールであり今ではウェブログの概念に吸収され同軸で考えても問題ないのかもしれない。今まで多くの個人サイトが生まれてきた、その中にはウェブログと似たコミュニティは数多く存在した。それとはどのような違いがあるのだろうか。そうした疑問は過去新たなコミュニティができるたびに湧き上がってきたものだ。しかし明確な答えは出てこなかった、それもしかたのないことである。なぜなら外見上はまったく変わらないからである。先ほど言った定義も以前のどのコミュにも当てはまることだ、むしろ今までそのような境界線があまりなかったのだから、いまさらそれが一つ増えたところで大してかわらない。ではどこなのだろうか、それは文化的思想とコミュニティである。

「低コスト流通と不特定多数の情報発信」「文章によるエンターテインメント」「ネット時代の情報共有」など同じような敬意式だがスタイルとしてはこのような違いが出てくる。

日本では伊藤穰一とm.e.s.hの面々が創世記の立役者となった。その新しい形式だが今までの個人サイトと変わらないスタイルに反発は起きたがじょじょにその便利さに飛びつき今まではテキストサイトや個人ニュースサイトとくらわれていたかもしれないサイトがウェブログ封に鞍替えしていった。このようにして徐々に面白さが広まっていったときはてなダイアリーが日本発の穂ホスティングサービスを行い徐々に無料レンタルサービスが増えていき潜在的にやりたいと思っていたユーザの底上げしていった。有名人のウェブログとい

うことで話題を集めたサービスもあった。このように企業が飛びついていったのはいままでなかなか儲けぐちが見つからなかったところに久々見えた光明だったのかもしれない。そんなに期待はこめなかったにせよ、しかし結果としてハードルが下がりいままでの何十倍のユーザ獲得という結果に繋がった。また構造的に見直すいい機会となった。

坂田 匠

経営学部経営学科

サーバサイドにおける Java の利用に関して

背景

昨今、ネットインフラの拡充にともない様々な Web コンテンツが生まれ、利用されている。そんな中開発現場では生産性の向上を狙い実装段階において、各種プログラム言語に合わせたフレームワークが頻繁に使用されるようになってきている。様々な言語が存在するなか、現在でも Web アプリケーションの開発言語として人気のある「Java」に焦点をあてて研究を行った。

Java

特徴及び概要

創成期

サンマイクロシステムズによって開発された言語であり、95年正式発表されに96年リリースされた。現在もサンマイクロシステムズによって開発が行われており、Web アプリケーションを中心に汎用的な言語として使用されている。

c/c++の代替言語として、組み込みシステムの技術を開発することを目的に当初開発が行われていたが、後に開発者たちはウェブ (World Wide Web) の世界に主眼を置くという方針変更した。

開発される言語は c++ の弱点を補うため重要な機能として「セキュリティ、分散コンピューティング、マルチスレッド、非プラットフォーム依存」を持たせることを目標とした。95年デモにおいて、Java と Java アプレットの技術により、ウェブページ内でアニメーションの表示やマウスによるインタラクティブな操作が可能であることをアピールした。その後、Netscape での動作が保障されることが発表されたことにより爆発的に普及した。

現在

クライアントサイドの Web アプリケーションとしての利用

Java アプレット、ブラウザ (ウェブのクライアント側) がウェブページ内でアニメーションの表示やマウスによるインタラクティブな操作を可能とする技術がある。最近では Flash や Ajax にとって代わられつつある。

サーバサイドの Web アプリケーションとしての利用

JavaEE、サーブレット、JSP などの技術が存在し、JVM の改良が進むにつれ徐々に使用されるようになってきている。これらの理由としては、(1) CGI のプロセスよりも軽量なスレッド単位で稼働するため負担が少ないということ。(2) ASP はサーバが高価、MS Windows NT

Server が必要となるため中小規模のネットワークインフラ構築に向かないことが挙げられる。そのため、Web アプリケーションの中でサーブレットの稼働は着実に増えてきている。

デスクトップ環境 (スタンドアローン)での利用

当初より、アプリケーションのために JavaSE パッケージが配布されているが、Java の普及後もあまり使われてはこなかった。理由として(1) 他の言語によるアプリケーションに比べて大量のメモリを使用する。また、起動に時間がかかる。(2) 当初、無償提供されていた開発環境の品質が悪かった。(3) ユーザはアプリケーションが動くバージョンの JRE をインストールする必要があったためである。だが近年、仮想マシン・コンパイラの改良、開発環境の強化、JavaWebStart による容易なインストールによりソフトウェアが広まりつつある。ソフトウェアの例としては、NetBeans、Eclipse といった (IDE: Integrated Development Environment)、Lotus Symphony (IBM)、Sun StarSuite、OpenOfficeorg などのオフィスソフト、V2C(2ch ブラウザ)などがある。

組み込みシステムとしての利用

JavaME というリソースが制限されたデバイス向けのパッケージがあり、携帯電話、PDA、テレビなどで利用されている。

サーバサイドでの利用

サーバサイドの Java 技術

JavaEE

JavaSE に大規模なシステム開発に必要とされる API、ライブラリを組み合わせたパッケージである。主に、Web アプリケーションなどのサーバサイド分野における開発で使用されている。

Java Servlet (ジャバ サーブレット)

Java を用いて、ウェブページのための HTML 文書などを動的に生成するサーバ上で動くプログラム、またはその仕様のことである。単に「サーブレット」と呼ばれることが多い。通常は Web アプリケーションのビジネスロジックを提供することに使用されるため、ファイルや DB へのアクセスに多用されている。Java EE のコンポーネント (一機能という位置づけになっている) である。Servlet は事前にプログラムと URL を対応付けており、ブラウザ側からの要求 (GET, POST 等) を受けることで実行される。HTTP 要求からデータを読み込み、HTTP 応答に書き出す方法でブラウザとの対話を行う。

Java Server Pages (JSP)

HTML 内に Java のコードを埋め込んでおき、Web サーバで動的に Web ページを生成してクライアントに返す技術のこと。JSP はサーブレットを自動生成して動作している。厳密に言えばサーブレットと JSP は違う技術だが、これらは組み合わせて使うのが一般的のため、JSP もサーブレットの一部として扱われることが多い。最初の読み取り時にサーバ側でコンパイルが行われ、HTML を含む Servlet に変換され、Servlet として動作する。

Web アプリケーションサーバ

Web アプリケーションを動かす際に必要となるサーバのこと。Web サーバに、プログラムを動作させるための環境(API、ライブラリ)を追加したものであり、Java Servlet、JSP はこの上で動作する。Java 用の Web アプリケーションサーバとしては、Tomcat、JBoss、GlassFish 等がある。これらの Web アプリケーションサーバは JavaEE の仕様を満たすよう設計されている。

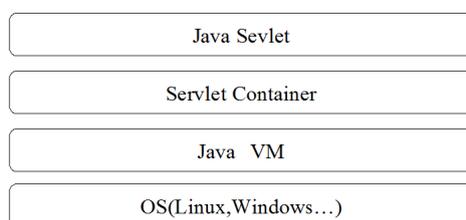
JSP/Servlet の動作原理

Servlet

Servlet は、OS 上で動く JavaVM 上で Servlet Container (Web アプリケーション) を介して動作を行う (図上、01. bmp)。

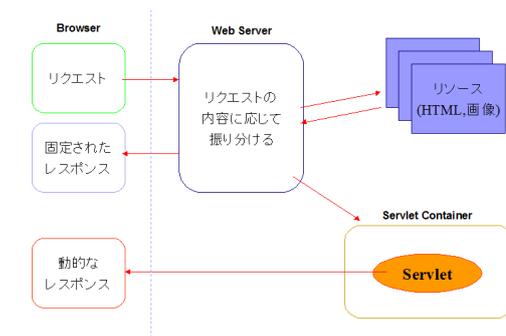


サーブレット動作原理(1)



ブラウザからのリクエストに対してサーバが HTML と Servlet との振り分けを行う。Servlet が呼び出された場合、Servlet コンテナが Servlet をコンパイルし、実行結果をブラウザに表示させるのである (図下、02. bmp)。

サーブレット動作原理(2)



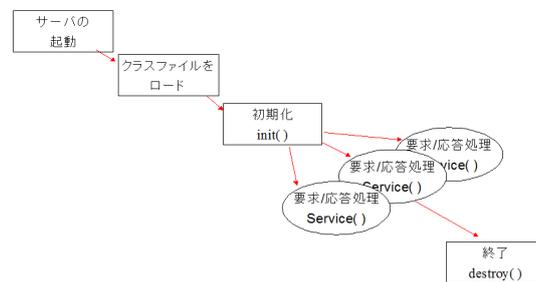
08/01/30

経営学部経営学科 坂田 匠

6

具体的には、クライアントからリクエストがあった場合、サーブレットコンテナは、クラスパスをサーチして、該当するサーブレットのクラスファイルからインスタンスを生成し、サーブレットコンテナにロードを行う。そして、init メソッドで初期化を行い、必要な処理を行うためにメソッド(通常は、doGet もしくは doPost)を呼び出す。メソッド実行後、サーブレットはコンテナに常駐し、以後、同じサーブレットに対してリクエストがあっても、メモリへのロードや初期化処理などを行うことなく、メソッドを呼出して処理を繰り返し実行する。管理ツールから何らかの指示があった場合 destroy メソッドの呼出しとサーブレットコンテナからサーブレットをアンロードを行う(図、03. bmp)。

サーブレット動作原理(3)



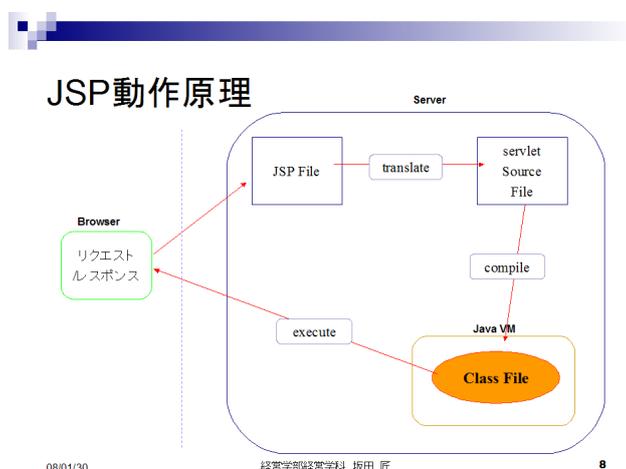
08/01/30

経営学部経営学科 坂田 匠

7

最初の処理にはコンパイルの過程が入るので、処理に時間がかかるが、2 回目以降のアクセスではその処理がないためすばやいレスポンスを行うことができる。

ブラウザからリクエストを受けた場合、JSP ファイルを一旦 Servlet に変換し、コンパイルを行う。そして、できたクラスファイルを JavaVM 上で実行し、ブラウザ画面に結果を表示させる(図、04bmp)。



HTML を用いた静的なページとの違い

静的なページは、常に表示されるページが変わらないため、クライアントサイドからの入力に応答させることができない。そのため HTML だけでは、BBS やブログのようなアプリケーションは構築できない。HTML と異なりステートレスでないため、ページを跨いで情報のやりとりが可能である。また、クッキーでなく HttpSession という機能を使い、サーバ側でセッション管理を行う。そのため、クライアントサイドでのウイルスやハード紛失による情報流出の心配がない。

Web アプリケーションに Java を使用することのメリット

開発に Java を使用することができるため、オブジェクト指向の利点を活かせることももちろんのこと、豊富にあるライブラリ・API 等を利用することが可能である。また、Java というメジャーな言語を使用することで教育のためのコスト(人件費)を低く抑えられる。Web 上での動作を考慮して開発された言語のため、C などの言語に比べライブラリ、ドキュメントが豊富にあり生産性が高い。Perl 等の CGI がプロセスとして一つのプログラムを動かすのに対し Java はスレッド単位で動作するため、軽量でありサーバへの負担が少ない。PHP と異なりビジネスロジックを別のファイルとして用意するため小規模な Web アプリケーション開発での生産性を犠牲とするものの、保守性や拡張性の向上が期待される。

JSP/Servlet を使用した Web アプリケーション開発の問題

MVC の枠組みが用意されていない

プログラマ自身が MVC モデルを意識して開発を行わないと、MVC の処理が各ファイルに混在することとなり、保守性や拡張性に問題をもたらす可能性がある。

データの管理機能がない

ブラウザから送信された情報を保持し管理する機構が JSP/Servlet には標準で用意されていない。そのため、実際のアプリケーションでデータ管理を行うためには各セッションに情報をもたせるなどをする必要がある。また、こういった動的ページの場合リロードのたびに初期状態に戻ってしまうようなことがあり、現在の状態を保持させるコンテキストの作成にはそれなりの手間を必要とする。

GUI 構築のための機能が貧弱

GUI とビジネスロジックを明確に分離されておらず、状態保持やデータ管理のための機構がない。また、GUI がコンポーネントかされていないため再利用が困難である。アプリケーション開発において同じことの繰り返が多い。

Web アプリケーションは同じような処理をいくつも作成することが多く、表示形式の違いがあるたびに基本的なスタイルに変更点を加えた同じようなコンテキストをいくつも作成することになる。このような定型的な部分を汎用化する仕組みが JSP/Servlet にはあまり用意されていない。あるとしても、カスタムタグの使用、細分化したファイルのインクルード程度である。

JavaEE の問題(旧 EJB の抱える生産性の低さ)

サーバサイド Java の標準仕様である JavaEE の中に Enterprise JavaBeans (EJB) というパッケージがある。これは、Java Beans をサーバサイドで実現したものであり、ビジネスロジックをモデル化するための技術である。以前の EJB、以前はテストの困難さなどの問題点があり、非常に生産性の低いものであった。これらの問題に対する解決策として、数多くのフレームワークが開発され、様々な現場において利用されている。

フレームワーク

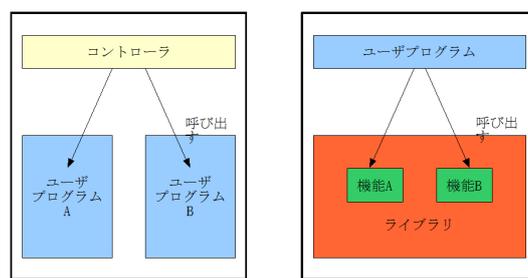
フレームワークとは(ライブラリとの違い)

同様の処理が複数回出現する場合、サブルーチン(メソッド、関数)として呼出すようにする。こうすることでプログラム作業の減少、バグの減少につながる。ある目的に沿って組み合わせたサブルーチン群(プログラム群)のことを、「ライブラリ」と呼ぶ。Windows では、

ユーザインタフェース用に user32.dll, comctl32.dll (Windows API) というライブラリが提供されている。同様に、MacOS X にはユーザインタフェース用 API として、Cocoa があり、Linux では Gnome の gtk+, KDE の qt などがある。

フレームワークもライブラリと同様に、汎用的に使用されるサブルーチンを組み合わせることで作られている。フレームワーク自体はその大半が完成された半製品であり、ホットスポットとよばれるユーザが操作可能な場所を設定することでアプリケーション開発を容易に行うことができる。ライブラリが開発者の作ったユーザプログラムによって呼び出されるのに対して、フレームワークではフレームワーク内のプログラムであるコントローラが全体制御を行いユーザプログラムを呼出している (図、05bmp)。

フレームワーク
-フレームワークとライブラリ-



08/01/31

経営学部経営学科 坂田 匠

4

開発にフレームワークを使用すること利点

フレームワークは必要な機能まとめて提供するものであり、開発ではアプリケーションの雛型として利用されている。これを利用することにより、独自に必要なとされる部分だけを開発すればすむため開発効率の向上が見込める。ある一定の制約下で開発が行われるため、開発者によって極端に異なるものができにくく、品質が一定に保たれる。そのため、GUI アプリケーション、Game、Web など統一したデザインを必要とするアプリケーションにおいて多用されることが多い。

サーバサイド Java で使用される代表的なフレームワーク

Apache Struts (Struts)

特徴及び概要

Jakarta プロジェクトによって開発が行われている、Java Servlet/JSP を用いた Web アプリケーション開発用のオープンソースフレームワークの 1 種である。

- Jakarta プロジェクト: オープンソースの Web サーバソフトなどを開発している Apache プロジェクトのサブプロジェクトであり、Java 技術に関連するソフトウェアを開発するプロジェクトである。MVC(Model View Controller)の考え方にしたがって設計されており、GUI とロジックを自然に分離した形で書ける点に特徴を持つ。
- MVC: アプリケーション開発において、プログラム内部のデータを model・view・controller の 3 つの部分に分割して設計・実装するという技法、又はそのような構造のこと。
- Java による Web アプリケーション開発ではデファクトスタンダードのフレームワーク (1. x 系) であり、開発現場において広く使われてきた。

Struts を使用することのメリット

開発の効率化

フレームワークの制約内で開発を行う必要があるため、開発者によって大きく異なるということがなくなる。

MVC モデルに沿った構造

MVC モデルではそれぞれの結合が弱いので、作業の分担が容易となる。また、ロジック部分と UI 部分が分かれているため、JSP/Servlet のみの開発に比べ画面デザインの変更に強い。多くの現場で使用されてきたフレームワーク (Struts 1. x 系はサーバサイドアプリケーションフレームワークにおけるデファクトスタンダード) であるため、他のフレームワークに比べドキュメントが多く、導入しやすい。

構造 (Struts の主要なコンポーネント)

JSP ファイル

Struts でも GUI 部分は JSP を使用し作成する。ただし、通常とは異なる Struts 専用のカスタムタグを使用する。カスタムタグを使用することで、煩雑な UI 開発の軽減を図っている。

Action Servlet

クライアントが JSP フォームなどのインタフェースを通して送信してきたリクエストを最初に受け取るコンポーネントである。リクエストに応じて Action Servlet が適切なビジネス・ロジックを呼出し、その結果を JSP に投げることでクライアントに応答する。Action Servlet は Struts 自身に用意されているためプログラマが作成する必要はない。

ActionMapping

Action Servlet は、ActionMapping にしたがって、要求に特定のモジュールをディスパッチ(割り当てる)する。ActionMapping は要求された URL パターンに対して、クラスの完全限定名をマッピングしており、ビジネスロジックをカプセル化した Action Class へのマッピングと、応答の JSP へのマッピングを行います。ActionMapping を介することで、Struts はビューとコントローラは独立性を保っている。

Action Class

MVC モデルにおける Controller を担当しており、struts-config.xml で指定したクラスにおいて execute メソッドを呼び出すことで、JSP/Servlet における doPost, doGet メソッドを実行する。

ActionFormBean

ActionFormBean は Struts コンフィギュレーションファイル(Struts の設定ファイル)の form-bean 要素から呼出し、実行時に ActionForm オブジェクトを生成する。

ActionForm

表示用 JSP から送信されたフォームの情報を管理する。この値は外部から読み書きすることができ、次に JSP を表示する必要が生じた際には、自動的にコンテキスト・コレクションに格納された値を元にページが構築される。

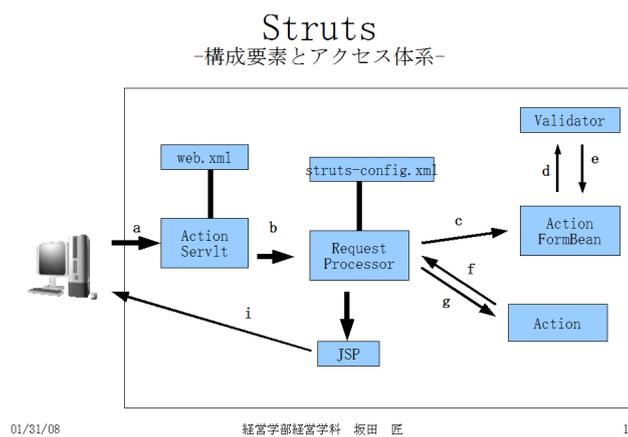
動作原理

- a. ブラウザのリクエストを受け付ける。web.xml を参考に、RequestProcessor に処理を引き継ぐ
- b. 引き渡されたリクエストが、どのアクションクラスに対応するか判断し、そのアクションクラスに対応する ActionFormBean を生成する。アクションクラスに対応させる判断は、struts-config.xml を参照します。
- c. ActionFormBean を参照し、リクエストのフォーム値を ActionFormBean へ代入。これ以降、フォーム値を直接取得する(getParameter()する等)ことはなく、ActionFormBean のインスタンスを参照する。
- d. ActionFormBean を作成し、リクエストのフォームインスタンスを作成するとき、Validator により値の検証を行う(オプション)
- e. Validator からの検証結果を得ます。もし入力値にミスがあり検証が失敗した場合、以降の処理に移らずエラーのページへ遷移する(オプション)
- f. ActionFormBean を引数として、Action クラスの Execute メソッドを実行する。

- g. ActionMapping から遷移先を取得。
- h. 取得した結果の指示を元に、遷移作業を行う。
- い。
- i. 遷移結果を表示。

注 i:d, e はオプションであり、開発者がアプリケーション内に組み込まない限りは使用されない。

注 ii:これは、Struts1.x 系の動作原理である。Struts2 では ActionForm がなくなり JSP を呼出したアクションから直接値をとるよう仕様が変更されている。これは、軽量言語の生産性の高さに対抗するためである(図、06bmp)。



JSP/Servlet を用いた開発との違い

JSP/Servlet には以下の問題がある。

- a. 入力情報チェックや繰り返し処理機能などを追加するのが、Struts に比べて面倒である。
- b. MVC モデルに沿って開発されておらず、JSP ファイルにロジックが埋め込まれている。そのため、更改作業が非常に面倒なものとなる。また、開発者以外の人間が保守を行う際に全体の把握に時間がかかる。そのため、上記 a, b のような問題が発生しやすい以下のようなアプリケーション開発の現場では、Struts 導入することで生産性を高めることができる。
- a. 一定度の大きさを持つもの。
- b. 入力フォームなどユーザから情報を受け取るもの。

Java Server Faces (JSF)

特徴及び概要

Java ベースの Web アプリケーションフレームワークの一種である。Java Community Process (JSR-127) で仕様検討が行われ、現在の JavaEE 最新バージョンである JavaEE5 には

標準機能として組み込まれている。UI(ユーザインタフェース)開発支援に力が注がれており、デザインツールを用いたビジュアルな開発が可能である。

-デザインツール: Sun Java Studio Creator, Web Tool Platform など

JSF を使用することのメリット

JSF は開発に力が注がれており、JSP/Servlet のもつ UI 弱点を克服することができる。JSP/Servlet の Web アプリケーションの場合、UI とビジネスロジックが混在しており、保守性や拡張性に問題がある。また、UI の状態保持やデータ管理の機構がない。それに対し、JSF は GUI のデザインとロジック処理を JSP、Backing Bean のそれぞれに担当させることによって切り離している。そのため、プログラマが開発することに関して、それらを意識する必要がない。また、JSP/Servlet では UI がコンポーネント化されていないため、UI の再利用やツールを用いた開発が困難なのに対し、JSF では UI を「クラス」「カスタムタグ」といったコンポーネントとして扱っているため、ツールベンダにとっては、UI デザインツールを提供しやすくなる。UI ツールを用いることで Microsoft の VisualStudio のように、部品を D&D(ドラッグ&ドロップ)による開発が可能となる。

-コンポーネント化(ソフトウェアの部品化): コードを機能単位で分離すること。このようにすることで、保守性(コードがみやすい)や生産性(部品単位での再利用が行いやすい)が高まる。

構造(JSF の主要なコンポーネント)

JSP ファイル

JSF でも UI は JSP ファイルを用いて構築する。ただし、JSF 専用のカスタムタグを使用して作成を行う。

Faces Servlet

アプリケーション全体のコントロールを行う Servlet である。これは JSF 本体に用意されているため、開発者が独自に用意する必要はない。

Backing Bean クラス

表示用 JSP の背後でロジック処理、データ管理を行うための Bean クラスである。ここで、クライアントサイドから送信されたフォームの情報の管理、各種コンポーネントを操作した際のイベント処理を行う。Struts のアクションクラスと異なり、JSP と 1:1 での対応をしていないため、各コンポーネントやイベントごとに Bean を個別に割り当てる柔軟な処理が可能だ。また、フォーム情報とイベント処理を別のファイルに分けて管理するという事もできる。

動作原理

a. Faces サーブレットにリクエストを送信

クライアントは Faces Servlet に対してリクエストを送信する。JSP ファイルへの直接要求はしない。

b. UI コンポーネント (GUI 情報のオブジェクト) の構築、JSF 内部での処理である Faces サーブレットは、JSF に用意されている「UI コンポーネント」と呼ばれるオブジェクト (GUI 情報を管理するオブジェクト) を用意する。これを使用することでクライアントからの情報を保存できる。

c. 値の検証、型変換

コンポーネントに設定された値の検証やキャストなどの処理 (コンバータ、バリデータ) を行う。これで UI コンポーネントに GUI 情報の構築することが完了する。

d. Bean への受け渡し

Bean のインスタンスに対し、UI コンポーネントから必要な情報を受け渡す。これにより、Bean 側で UI コンポーネント操作する準備が整う。

e. イベントの処理

イベントリスナーによって処理される。通常、リスナーにより呼び出されるメソッドは BackingBean に用意してある。インスタンス内のメソッドで処理することで、各種のビジネスロジック処理を行う。

f. 送信用 UI コンポーネントの作成

ロジック処理を行い GUI を作成する。JSF カスタムタグをもとにして UI コンポーネントの再構築を行う。

g. GUI のレンダリング

UI コンポーネントのインスタンスから HTML タグのソースコードを生成する処理を受け持つ「レンダラ」によって HTML コードが生成される。

h. クライアントへの送信

結果をクライアントサイドへ送信しブラウザに表示させる。

JSF と Struts の違い

• Web アプリケーションフレームワークとしての方向性は異なるものの (Struts が MVC アーキテクチャに基づいた Web アプリケーション開発を目指すのに対し、JSF は UI 開発に重点をおいておりフォーム等がコンポーネント化されている)、双方ともに Web アプリケーションに必要な機能はほぼ充足している。また、開発ツールに関しても、双方ともに EclipsePlugin やサードパーティによる Integrated Development Environment (IDE : 統合開発環境) が存在しており、大きな差異はない。

-Struts の IDE : StrutsIDE, Exadel Studio

-JSF の IDE : JSF Tool, NetBeans, WebSphere Studio Application Developer など

一番の違いは、Struts がデファクトスタンダードなフレームワークなのに対し、JSF は JSR で仕様策定されている標準的な Web アプリケーションであり、JavaEE5 の API 仕様に含まれている。そのため、ベンダが JavaEE5 準拠の製品を開発する際には、JSF の機能を実装しなければならない。そのため将来的に、Java による Web アプリケーション開発の場において、JSF ベースのフレームワークが Struts に変わるもしくは、Struts と組み合わせて使用されていく可能性が高いといえる。

Spring Framework

特徴及び概要

Java ベースの Web アプリケーションフレームワークで、Spring Framework.org によってオープンソース による開発が行われている。依存の注入 (DI) という手法を導入する軽量コンテナ (フレームワーク) であり、扱いにくかった従来の J2EE (Enterprise JavaBeans 2. x) に代わるフレームワークとして普及した。

「プレゼンテーション・レイヤー」「ビジネス・レイヤー」「データアクセス・レイヤー」の 3 つを疎結合 (動的な関連付け) することで、拡張性や保守性の向上が得られる。

歴史的背景 (旧 EJB の抱えていた問題)

従来の EJB (1. x, 2. x) には技術的な課題が存在しており、多くの開発者の指示を得ることができずにいた。

- 記述する上で EJB 独自の様々なルール (指定されたインタフェースの実装など) を守る必要がある。また、コンテナにデプロイするまで誤りを検知することができない。
- メソッドを動作させるには EJB コンテナの持つコンテキスト (実行時情報) が必要なため、EJB コンポーネント上でしか動作させることができない。そのため、JavaSE 環境で JUnit などのテストプログラムを利用することが困難である。
- エンティティ Bean (永続化クラス) の設計に制約が課されるため、オブジェクトモデルを自由に設計できない。また、JavaEE サーバ上で動作させるためには、サーバに合わせた定義ファイルを用意する必要があるため、JavaEE 仕様のポータビリティ (可搬性) が失われている。
- SQL パフォーマンスのチューニングができない。
- デプロイ記述子 (Servlet と URL の関係付け等を行う xml ファイル) が複雑である。
 - ・そのため、アンチ EJB のムーブメントが湧き上がり、「POJO (Plain Old Java Object)」への回帰を基本思想とする「Light Weight Java」が叫ばれるようになった。
- POJO: 古き良き Java のオブジェクト、EJB に対する批判の意を込めた造語である。
- Light Weight Java: EJB を使わないアーキティチャ

・POJO ベースのクラスは、特定のフレームワークやコンテナが提供するクラス・インタフェースの継承・実装を行わないため、クラス設計に EJB のような制約が無い。そのため、クラスを動作させるために必要なコンテキストなど(依存性)、フレームワークを使うことで動的に注入する必要がある。

- 依存性:オブジェクトが成立するために必要な要件

このようなフレームワークを「軽量コンテナ(Light Weight なコンテナ)」と呼び、Spring Framework は軽量コンテナの 1 種である。

Spring Framework の 2 つの重要な機能

DI:Dependency Injection(依存性注入)

- DI とは、あるクラスに対する依存性を外部から注入するデザインパターンであり、以前は IoC (Inversion of Control) と呼ばれていた。

- DI を用いることで、“プログラムを動作させる上で必要となるモノ”との依存関係を弱くするため、クラスの単体テストを行いやすくなる。

- DI を汎用的に利用するためのフレームワークを DI コンテナと呼び、モノの定義ファイルを元にオブジェクトの生成(or 取得)・インジェクションを行う。

・AOP:Aspect Oriented Programming(アスペクト指向プログラミング)

- AOP(アスペクト指向プログラミング)は OOP(オブジェクト指向)と組み合わせて使われ、OOP を補完する役割を担っている。

- AOP を利用することでソースコードを変更すること無く処理を追加することができるようになる。そのため、ログ出力のような“複数のクラスで共有したい機能”を一ヶ所にまとめることが可能となる。

- AOP を利用することのメリットとしては OOP の継承と異なり、横方向に機能追加するため各クラスを横断するような機能追加ができ、各クラスにソースコードを埋め込む必要が無いということ。また、ソースコードに後から処理を付け加えることが可能なため、コーディング 規約からはみ出てしまった誤りを正すことができるなどが挙げられる。

既存フレームワークに与えた影響

POJO ベースフレームワークの流行

以前の EJB ではとりわけ、エンティティ Bean と呼ばれる永続化クラスに批判の大半が集中しており、いかにしてこの部分の生産性を上げるかが急務であった。そのため、Hibernate や Java Server Faces、Seasar2 など多くのフレームワークで、容易にテストが行えかつ設計の上で制約の少ない POJO が取り入れられた。

EJB の改良

軽量コンテナの人気の高まる一方、EJB は危機に瀕していた。そのため、現在の EJB3.0 の仕様検討の際に、Hibernate の作者であるギャビン・キングを招き軽量 Java の設計思想を

取り入れた。結果として、3.0 になって DI コンテナ、AOP フレームワークの機能を取り入れ、Spring Framework の持つ優位性は少なくなった。

サーバサイド Java の抱える課題と今後の利用

課題

軽量言語に対する生産性の低さ

以前より言われてきたことだが、Ruby、Python と言った軽量言語に比べ Java による開発は生産性が低い。これは、Java が例外処理、変数の型等の処理を厳密に定める必要があるため実装の段階が煩雑になってしまうためである。また、たしかに、実行速度の点においては Java に大きな欠点がある。だが、開発現場において生産性に対する需要が高いためさほど問題とは言えない。

Java の習得の難しさ

Java 習得過程において、どうしてもオブジェクト指向や継承と言った Java の持つ大きな特徴に時間が割かれる傾向が強いのも問題である。このために他の言語に比べて習得に時間がかかりやすい。

今後予測されるサーバサイドでの Java 利用

大規模開発での利用

サーバサイドアプリケーションにおいて、軽量言語が Java に対し優れているかと言えばそうではない。たしかに、小規模で生産性が重視されるような場合は軽量言語のほうが圧倒的に優位と言えるが、大規模な開発現場ではそうともいえない。技術的な面で言えば処理速度の速さや各種ツールの豊富さで優位な面がある。また、開発人員の確保に関しては Java 技術者のほうが容易に行うことができる。なぜならば、Java 開発者の多くが Web アプリケーションの開発者であるためである。これは以前の Java 仮想マシンが、スタンドアローンアプリケーションに使用すると起動までの時間が長く非常に扱いにくかったため、Java を用いた開発は Web アプリケーション専用となっていた時期があったためである。そのため、大規模な開発においては Java が使われていく可能性は高いと言える。

軽量言語との組み合わせ

現在、「Ruby on Struts」、「Ajax + Java Server Faces」など軽量言語+Java という開発手法が注目を浴びている。これは、Java を用いることで世の中にある JavaAPI、ライブラリ、フレームワークを使用できるという点、軽量言語を使用することで生産性を高めること

ができるという点、JavaVM上で処理を行うことで実行速度を高められる点でメリットがある。そのため、軽量言語のライブラリ及びAPIが現在のJavaほどに豊富になり開発の主軸に移るまではこのような使い方はどうぶんされると考えられる。

引用：

『はじめての Java フレームワーク』 著作：岡田賢治、出版：秀和システム

『Eclipse 3 ではじめる Java フレームワーク入門』 著作：掌田津耶乃、出版：秀和システム

『はじめての JSP&サーブレットプログラミング 改定第3版』 著作：アイティーブースト、出版：秀和システム

『Spring 2.0 入門 Java・オープンソース・Web 開発自由自在』 著作：長谷川裕一／麻野耕一／伊藤清人／岩永寿来／大野渉、出版：技術評論社

『マスタリング JavaEE 開發生産性を高める Java フレームワークの標準仕様』 著作：斉藤賢哉、出版：翔泳社

松崎 有樹

法学部

Second Life SIM 開発を振り返って

この約半年の間、Second Life に私たちは実験的な事をいくつか行ってきた。またその中においてわかった事、困った事が数多くあった。それらをここにまとめ、後代のメンバーへの引き継ぎの一つとしたい。

建物の建設

2D はともかく 3D のツールは使った事はない、そんな人は多いのではないか。しかし Second Life ではいとも簡単に建物が建てられてしまう。まずは「建造」ボタンを押してみよう。このツールでは 3D 空間上で手に取るように加工が



図 1 立方体

できてしまう。立方体をクリックしてしてから画面をクリックすれば立方体ができたとわかるだろう（図 1）。いくつか矢印が見えるかと思うがこれを引っ張ったりする事で形を変更する事ができる。普通の状態では矢印を引っ張るとオブジェクトを動かす事ができる。コントロールキーを押すと回転、コントロール+シフトキーで拡大縮小ができる。あとは板や柱をつくり組み合わせるだけだ（図 2）。標準で用意されている形だけでは少し物足りないという人はオブジェクトの項目でひねったり、一部削ったり、空洞化したりということができる。空に浮かぶ島（図 2-2）では多用しているので参考にしていきたい。

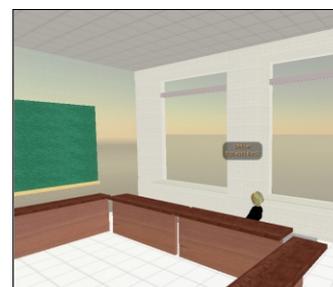


図 2 完成した建物の例

衣服の制作

Second Life では自分のシャツぐらいは簡単につくる事ができる。公式サイトでテンプレートが公開されているので、それにざざっと色を塗って取り込めばあとは服として解釈してくれるので、むしろアップロードの方が手間に感じるかもしれない。

具体的にはテンプレートをまずこちらからダウンロードする。

<http://secondlife.com/community/templates.php>



図 2-2 完成した建物の例

そしてそれを展開して Photoshop あるいは GIMP などの編集ソフトを使って色を塗る。テンプレートは多重レイヤーとなっていてそのほとんどがガイドのラインなので、表示非表示を繰り返すとよいだろう。うまくできたならば早速アップロードをする。このアップロードなのだが Second Life では一々仮想通貨が必要になる。アルバイト等をすればすぐ溜まるかと思うので用意しておこう。ファイルから画像のアップロードを選べば、ファイルの選択画面に移るはずだ。ここでのポイントだがその次の画面で服を実際に着た場合のプレビューが見れる。ここでチェックしておけば無駄にお金を使わなくて済む訳だ。



図3 容姿編集モード

さて無事アップロードが完了するとそのデータはテクスチャとしてユーザのフォルダの中に入っているはずだ。あとは容姿の画面でこのテクスチャをシャツのデータとして選べばいいだけだ(図3)。無論この方法では3Dに関することは考えていないので立体的な服をつくる事ができない。よってどんな服のデザインもウェットスーツのように体にピッタリ密着してしまう。スカート等のひらひらしたものもつくる事はできない。この辺りが次の課題だろう。

スクリプトの利用

建物はできたが、もう少し面白い仕掛けを施したいと思う事もあるだろう。いくつかスクリプトを試してみたのでその結果をまとめてみたい。

まず、スクリプトの利用の仕方について。オブジェクトを編集画面にしてからコンテンツー新しいスクリプトを選択する。



図4 新しいスクリプト

するとスクリプトを記述できる画面に移る(図4)。ここでスクリプトを書けば良いのだが、スクリプトに関する知識がなくても Web 上に公開されているサンプルのコードを貼付けるだけでも十分活用できる。



図5 操縦の難しい飛行機

法政の学生スペースを見るといくつかの建物が見えるかと思うが、これらに使用されているスクリプトを挙げてみたい。

- ・オブジェクトの上に字を表示するスクリプト (商品等)

- ・クリックするとメッセージを表示するスクリプト（マスコット）
- ・クリックすると扉が開くスクリプト（空母の後ろ）
- ・テクスチャを回転させるスクリプト（噴水・お風呂等）
- ・自分で運転する乗り物（車・飛行機）
- ・遊覧飛行等勝手に動く乗り物（飛行船）

スクリプトを理解してその上で一から自分でコードを書くという事は残念ながらできなかった。そのため特に乗り物関係に関してはうまく制御できておらず電車等もつくるのが困難だった。そこが課題となる（図5）。

商品の販売

意外に思うかもしれないが、商品の販売自体にはスクリプトは必要ない。オブジェクトのプロパティをちょっといじるだけでできてしまうのだ。編集モードからコンテンツを選択。コンテンツとはオブジェクトの中にデータを入れる事で、スクリプトの他に物を入れる事ができる。つまり箱というオブジェクトの中に洋服のテクスチャを入れてみたりするわけだ。後は一般から販売用項目で料金や小勇健について設定すればいい（図6）。

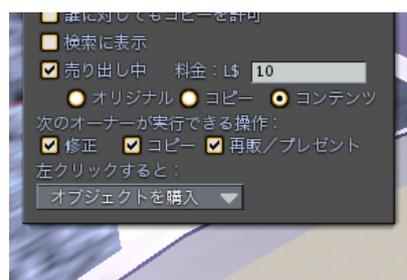


図6 販売用項目

スカルプテクスチャ

Second Life では積み木のように直法体や円錐などの基本的なオブジェクトを並べたり、簡単な加工を加えるだけでも十分世界を創造できるが、ちょっと凝った形を作りたいと思う事もあるだろう。そのためにスカルプテクスチャ

という方法がある。IT 研究センターで導入済みの Maya の他、Shade では Second Life 対応を表明し、よりスマートにデザインできるようになりつつあるが、私たちは使い慣れていた Blender を使った方法を使用した。



図7 Blender

Blender とはフリーの 3D 編集ソフトである (図 7)。フリーとはいえプロフェッショナルなユーザにも耐えられるほど高機能なツールだが、インターフェースにクセがあるのが主な特徴だ。

話が込み入ってしまうので簡単にまとめるがオブジェクトの 3D 情報を 2D のテクスチャに焼き込み、それを取り込むとその形になるのだ。これは形だけのデータなので別途絵柄となるテクスチャが必要になる。(図 8) は球体の上部をつねったような形だ。これはブロックの組み合わせでは難しい形だ。他の形も試してみたが、横から何か突き出ているような形は難しいようだ。これは円柱上に 3D のデータを書き込む事からの起因しているのだと思う。この点は次期からは新しいソフトウェアが導入されると思うので、このあたりも解決するだろう。

Second Life の現状

法政大学では Second Life という世界最大の仮想空間での取り組みを今まに行っている最中である。私もその中の一人として、短期間ではあったが参加したのでその事をまとめ、これからの展望としてまとめていきたい。

まず、Second Life の現状であるが、残念ながら日本においては未だ厳しい状況にあるといえる。実際にログインしてみると 30 代ほどの社会人が多く、大学生はほとんど見かけない。これは Second Life の発展において問題であり、また法政大学が Second Life への参入にあたってぜひとも解決したい問題である。

原因はいくつか考えられるが、私はその一つとしてビジネスを前面に押し出した PR があると思う。つまりマネーを中心に考え、実世界にどれほど影響があるかを訴えるものである。これは社会人の目を振り向かせるのに役立ったと思うが、大学生には今ひとつピントがずれているのである。もう一つは Second Life に何を求めてログインするのかについての見解の違いである。Second Life は決してアトラクションを楽しむ事がメインではない。とある企業が Second Life のブースを立ち上げたという話を聞いてアカウントを作成し、そのブースへは行ってみたがそれが何なんだと思い、それ以来ログインしていない。こんな話はよく聞く。

Second Life の強みとは



図 8 液体風キャラクター

Second Life の強みとは何か。それは様々な既存の Web サービスを統合した上で、単一のコミュニティが生きるところである。例えば動画配信であれば、Youtube・Podcast・ニコニコ動画などのサービスが既に存在し、ショッピングは Amazon・ヤフオク・楽天、企業紹介であれば各企業ホームページがある。しかし、これらはコミュニティの存在を前提とはしておらず、ニコニコ動画や、mixi についてもコミュニティは確かに存在するが既存の Web サービスの統合まではできていないように思う。

つまり先ほどの例を挙げてみれば、ブースに会社の同僚、大学の友達と一緒にいく、あるいはそのブースがそもそも自分の興味に密着した土地だったら評価が違ったのではないかという事である。私はこれが法政大学でぜひできないかと考えている。大学の休講情報や授業配信はあった方が良い。しかしこれはメインではない。上述した通り Second Life の立ち上げの時間等を考えても Web ブラウザで行った方が遥かに効率的だからだ。しかし、友人やサークルの先輩と掲示板を見に行ったり、授業を受ける事ができたら面白いだろう。

問題はどうやってコミュニティを育てるかである。バラバラの個人から一からコミュニティを組み上げるのは不可能に近いと思う。そこで現実世界の既存の団体を利用する事が考えられる。幸い法政大学にはクラス・ゼミ・サークル等多くの団体が存在している。特にサークルは長年の優秀な実績によりかなり個性的なブースが期待できる。また、近年の校舎の改装に伴いたまり場不足は慢性的な状態にあり、広報次第では Second Life がたまり場の一つの形として定着する可能性は十分にあると思う。

これからのロードマップ

最後に私の考える Second Life の展望を述べたい。まず、私の案では制作担当を運営とクリエイターに分割する。これは多くの法大関係者の興味を引きつけるアイデアを定期的に更新するには少人数では非現実的であり、またコミュニティ育成の観点からより多くの作り手にゆだねるべきだと思うからである。運営側はルールを策定し、クリエイターの認定条件・土地の利用ルール・ルール違反者への処遇等を決める。それに基づいてクリエイターを募集し、いよいよ制作が開始する。ここで重要なのは大勢のクリエイターが IT 研究センターに入る訳にはいかないという事だ。そのため、それまでには法政大学の一般回線から Second Life へのログインができるようになっている事がとても重要である。

最後に

ここまでの道のりは決して簡単ではないと思う。期間も1～2年はかかるだろう。しかし、その先に多くの学生の支持によって裏打ちされた本当に賑やかで面白いSIMが待っているんじゃないかと期待している。

松原 正和

経営学部 経営学科 04F0124

「Second Life × Hosei Univ.」レポート

メタバース(Metaverse)

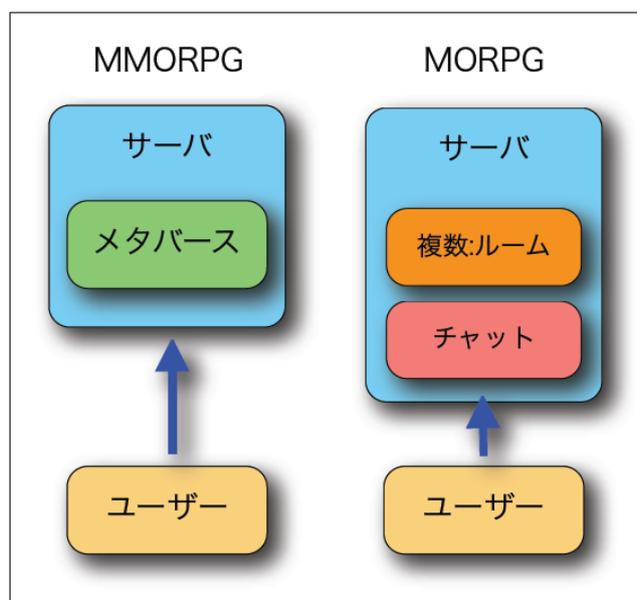
メタバースについて

メタバースとはインターネット上に存在する三次元仮想空間のことを指す。代表的なメタバースとしてリンデン社が運営する「Second Life(セカンドライフ)」や、和製セカンドライフとして開発が進んでいる株式会社ココア (Co-Core) の「meet-me」が存在する。このメタバースは今後、ビジネス・教育などの場での多くの利用が期待されている。このメタバースという言葉は造語であり、SF作家のニール・スフィーブンスン(Neal Stephensp 1959～)の自著「スノウ・クラッシュ」の中で使用したメタ(meta)とユニバース(universe)の合成語が由来である。メタバースの代表的な物として前述したように、「Second Life(以下・SL)」が上げられる。このSLでは三次元仮想空間の特徴を最大限に活かした経済活動・コミュニケーション活動などが、24時間世界中のユーザによって行われている。SL内では敵となるモンスターや殺し合いなどはないものの、広義ではオンライン(ネット)ゲーム・MMORPGに分類される。

MMORPG と MORPG について

オンラインゲームには多種多様な物が存在するが、SLは中でもMMORPGに分類される。このMMORPGは「多人数同時参加型オンラインRPG」と訳される。このMMORPGと近い物としてMORPG(Massively Online Role-Playing Game)が存在し、「複数プレイヤー参加型オンラインRPG」と訳される。こちらもMMORPGと同様なオンラインゲームの一種でコンピュータRPGをベースとしたものを指す。

MMORPGとMORPGの違いについて述べると、MMORPGはサーバ内に存在するゲーム世界に対し不特定多数のプレイヤーが接続する



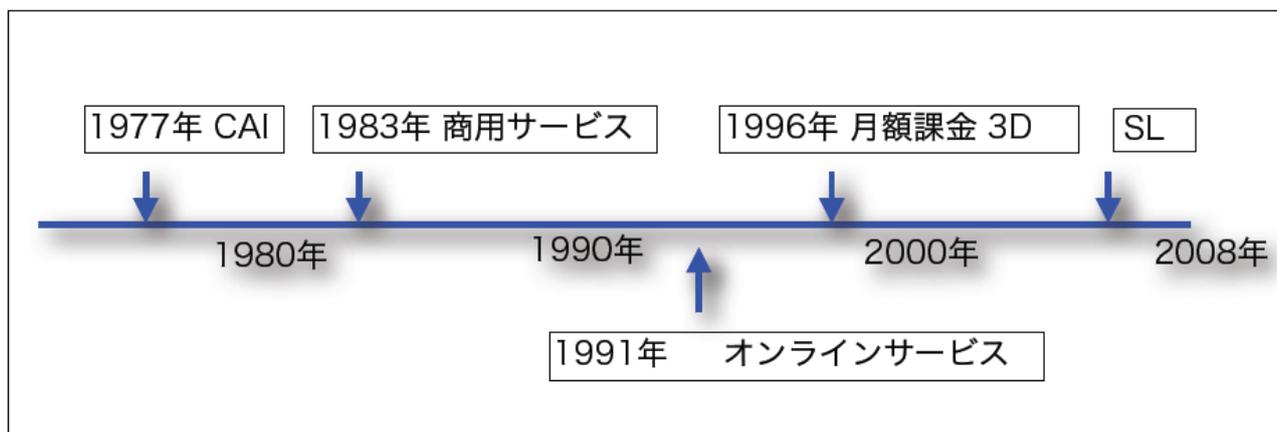
物に対して、MORPGは比較的少ない参加者が集まり、それぞれの独立したゲーム世界を作成してゲームをプレーするという物であり、ここで重要な事は、これらのゲームの参加者が本質的な違いではないという事である。例えて言うと、yahooなどのポータルサイトが運営するチャットと同じ様に、複数のゲーム部屋が用意されていて、参加者が利用したいゲーム部屋を選択してゲームに参加する物である。この様な部屋をロビー・ルーム制と呼ばれるゲーム開始前に参加者同士が集まり、そこで募ったメンバーと独立したゲーム世界でプレーするのである。

MMORPG について

MMORPGという概念は1970年代後半に生まれた。1977年にCAI (Computer-Assisted Instruction: コンピュータ支援教育) システムを応用し複数人数が同じ世界でプレイできるゲームが開発された。それ以来、MUD (Multi User Dungeon) と呼ばれ欧州中心に広がったテキストベースのシステムが多数作成されていく。ホストコンピュータ側にひとつの世界を構築し、プレーヤはパソコン通信の端末ソフトやTELNETターミナルソフトを介してサーバに接続し、コマンドを入力して自分のキャラクターを行動させたり、同じサーバに接続している他のプレーヤとのチャットを行う単純な物であった。

このMUDはイギリス・エセックス州のエセックス大学を始めとする教育機関で実験的に行われ、学内外問わず多く人に利用され人気が出た。1983年以降になると商業サービスとして提供する会社も現れ、この時点からテキストベースからグラフィックインターフェイスを追加する試みがおこなわれていき、参加者はキャラクター(アバタ)によるチャットシステムを経て、グラフィック処理を兼ね備えた最初のMMORPGである「Neverwinter Nights」が1991年にアメリカオンラインでサービスが開始された。この時点で今のMMORPGのシステムが形作られていった。

1990年代中盤になるとインターネットの普及により世界のプレーヤが一カ所のサーバに接続する事が可能になってくる。1996年には月額課金、3Dグラフィックスによる現代的なMMORPGとして3DO社のen:Meridian 59が開始され、1997年のウルティマオンラインによってこのジャンルが確立され一般的なものとなった。それ以降、インターネットの普及や端末の処理能力の向上により複雑なシステムや機能を持ったMMORPGがアメリカ・日本のゲーム業界から登場している。



MMORPG の特徴

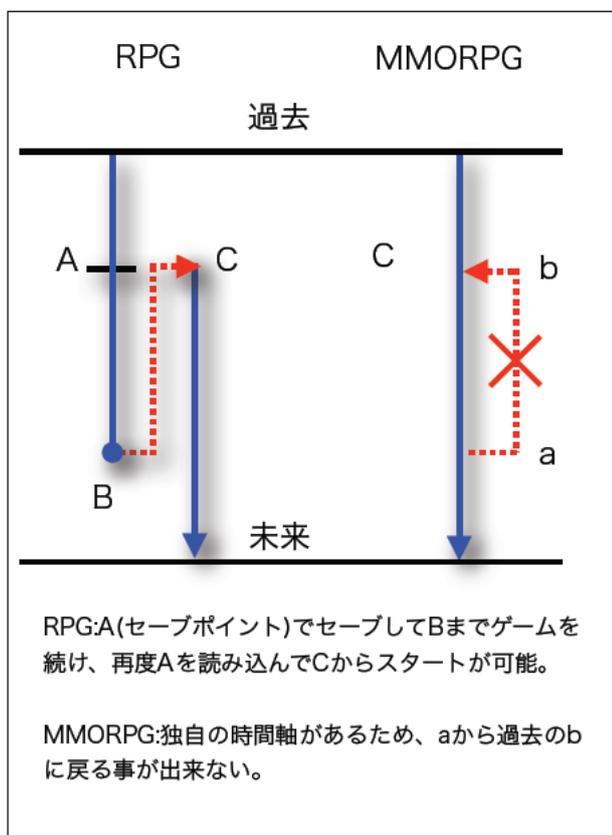
現代のSLを始めとするMMORPGの特徴としては、「三次元(3D)仮想空間・独自の時間軸・人間関係・経済システム」が上げられる。

三次元(3D)仮想空間について

MMORPGは、SLを始めとした三次元(3D)の物とそれ以外の簡易的な二次元(2D)の物に大別できる。2Dではレイヤの貼付けと透過処理によって画面を構成するのに対し、三次元ではポリゴンの生成とテクスチャを貼付けることで画面を作り出している。初期のMMORPGは、パソコンのグラフィック機能のうち、3Dレンダリングエンジンが未発達であったこともあり、ウルティマオンラインなどは2D方式を採用し、低解像度の物が主流であった。その後、パソコンのCPU・グラフィック機能の向上に伴い、2Dと3Dを組み合わせた物が登場し、その後完全に3D化したMMORPGが登場した。

操作性という観点からみると3Dになった事で利用者が空間を意識しやすくなる反面、3Dであるが故にキャラクタ(アバタ)の操作に関して充実したインターフェースを備えていないと、初めて利用する利用者からすると非常に利用しにくく、意欲をそいでしまうと同時に空間に酔ってしまい、車酔いと同じような感覚に陥る事もある。

独自の時間軸



我々、現実世界を生きる人間には1日24時間という時間が絶え間なく時を刻み続けているが、MMORPGも現実世界の時間と同様の独自の時間軸が流れている。その為、プレーヤ1人もゲームに接続をしていない状態でもゲームの世界が存在する限り、独自の時間が進み続けるのである。

これがMMORPGである場合は、ゲームを開始すると同時にゲームの世界が作れるので、ゲーム終了とともにその世界は消去される。そしてまたゲームが開始されると前回のゲームと同じゲーム世界でプレーする事ができるのである。一方、MMORPGの場合はMMORPGと違い、ゲームの時間やユーザの情報は常にサーバに記録されるので時間を意識したイベントなどが可能である。

この独自の時間軸が存在することで、時間をトリガとしたゲーム内のイベントを行う事が可能であるが、MMORPGは一般的なプレーステーション2などの1台端末完結型PGBと違い難しい問題が生じる。通常のRPGの場合は、ゲームを保存しておけばゲームの電源を切って再度、ゲームを開始する時に以前、保存したセーブデータを読み出してゲームを続行すればよいが、MMORPGの場合は常に時間が流れているのと、他のプレーヤが利用している事からゲームをセーブし、セーブデータからやり直すという事が出来ない。これは他のプレーヤの操作の矛盾を引き起こすからである。その為、一般的にはポイントの減点や何らかのペナルティをプレーヤは受けて生き返る方法がとられる。

人間関係

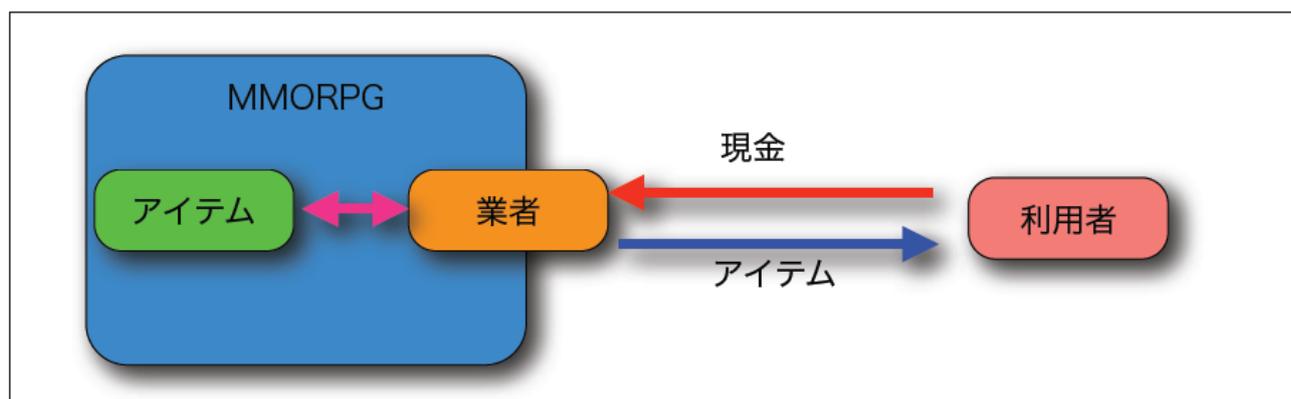
MMORPGの特徴として人間関係がゲーム内に存在することが上げられる。現実世界にも人間関係や組織がある様に、ゲーム内にも同様な関係や組織が形成される。これはチャットにキャラクタ(以下:アバタ)という存在が加わったことで一般のRPGと違い、アバタ同士がコミュニケーションを図り連携しながらゲームを進行したり行動したりする事が可能になり、連携を図る為にはプレーヤ同士の間で、英語・日本語などの現実世界の言語を用いて意思疎通を行う必要がある。その結果、集団で行動をしたり一定の思想や考えを持った集団や組織が生まれる。

また現実世界と違い、アバタという自分のキャラクタを操る事で初めて会った人に声をかける事も容易にで、現実世界の自分と違った一面を持つアバタになりきる事ができる。その為、男性なのに女性と偽って男性を騙したり、相手が女性と分かるとセクハラにあたる言葉を投げかけたり、アイテムの交換などという情報を持ちかけ詐欺行為を行う人もいるので、利用者のモラルが問われている一面もある。

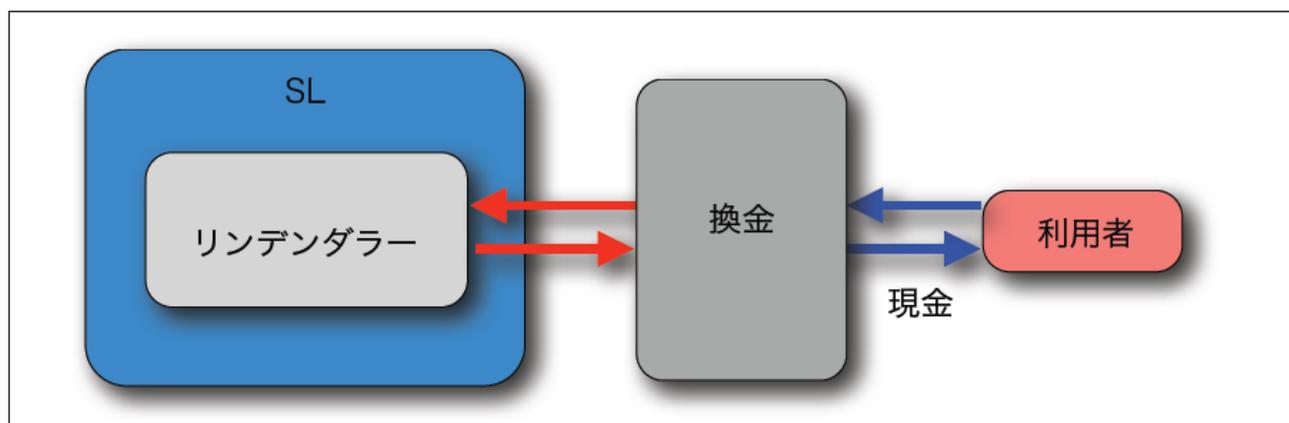
経済システム

MMORPGは規模の大小はあるものの経済システムが存在する。例えばSLでは「土地・アイテム」などSL内で物の売買が可能である。前述したように独自の時間軸が存在し、時間が常に流れている為に、ある一定期間にしか入手出来なかったアイテムを高額で販売を行っていたり、独自で需要の高い商品をMMORPG内で作り上げることで、経済活動が可能である。

この様な経済システムを功名に利用し現実世界でお金儲けを行う業者も存在する。ネットゲーム内でゲーム目的ではなく集団であるアイテムの入手やポイントを稼ぐ為に戦闘を繰り返すことでアイテムやポイントを入手し、それを現実世界で現実のお金で取引を行う業者も存在する。



SLでは現実世界のお金をSL内独自の通貨単位である”リンデンダラー”に変換して、SL内で利用することも可能である。またその逆に、SL内で得たリンデンダラーを現実の通貨に変換して現金として現実世界で換金する事も可能である。独自の経済システムが発達することで、MMORPG内での流通や雇用といった事も行われる。



Second Life(SL)

Second Life とは?

Second Life(SL)の特徴

セカンドライフとはリンデン・ラボ(Linden Lab)社が運営しているメタバースである。従来のオンラインゲームと違い、利用するユーザがSL内で自由に建物を作る事ができたり、経済活動を行う事が可能である。リンデン社はあくまでも仮想空間を提供し、世界中のユーザはその仮想空間内で自由に建物を建築したり、経済活動をおこなったりする事が可能なのである。よって従来のMMORPGと違い、敵やモンスターが登場しない分、イベントや企画は参加ユーザ次第なのである。

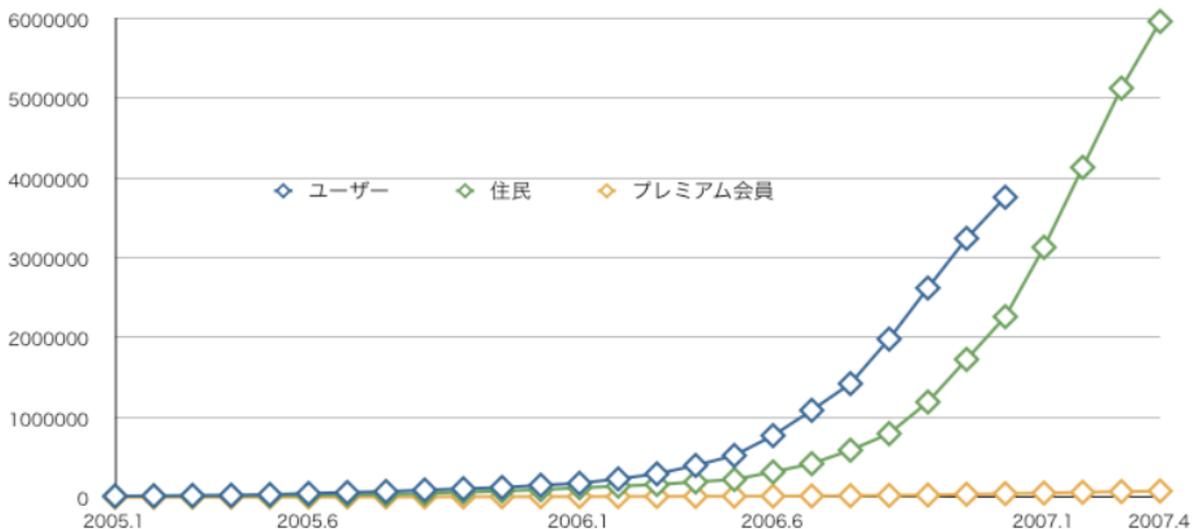
詳しくは後述するが、SL内では自分の土地を購入する事ができ、権限のある土地ではプリムという基本形の立体を組み合わせる事で、家やオリジナルな乗り物などが作れる。そしてMMORPGの特徴の1つでもある、独自の時間軸がSL内にも流れている為に、一度訪れた土地でも次の日に来ると別の建物が建っていたりとユーザを飽きさせない仕組みになっている。

またSLの最大の特徴としては、前述したMMORPGの特徴で述べた独自の通貨システムが導入されている。SL内での通貨単位は「リンデンドル(L\$)」である。土地やアバタ(自分の分身)が着る服などの売り買いの際にこの通貨を使用する。そして、このSLの仮想空間内ではしか有効でないリンデンドルを現実通貨に換金する事が双方向に可能である。この結果としてSL内で不動産や人気アイテムを売る事でリンデンドルを手に入れ、現実の通貨に変換してお金を得るユーザも出て来ている。実際のドルをリンデンドルに換金する際は相場の変動はあるが、「1ドル(\$)=270リンデンドル(L\$)」とが一般といわれている。日本円で換算すると「120円=1ドル=270リンデンドル」ということになる。つまりSL内で物を購入する際は表示金額の半分が日本円で換算の金額に相当する。このリンデンドルはSL内でアルバイトという形式でリンデンドルを稼ぐ事ができる。

Second Life の統計データ

SLのユーザとアバタ人口について

SLは2005年から運用をされた。日本語版SLがリリースされたのは2007年7月13日である。2007年4月の時点でリンデンラボ社から発表された統計データ(下図)では、登録ユーザ(無料)は376万人。その登録ユーザが所有するアバタは597万人。一方、プレミアム会員は8万人となっている。



March 8, 2009

アカウント種類	無料登録	追加オプション	プレミアム登録
費用	無料	\$9.95	月払い:\$9.95 3ヶ月:\$22.50 年払い:\$72.00
アバターのカスタマイズ	可	可	可
契約時ボーナス	L\$250	-	L\$1000
お小遣い	-	-	L\$300/週
アイテム作成	可	可	可
土地所有権	-	-	可

Age Range	Percentage
13-17 (Teen Grid)	1%
18-24	26.64%
25-34	38.51%
35-44	21.78%
45 +	12.06%

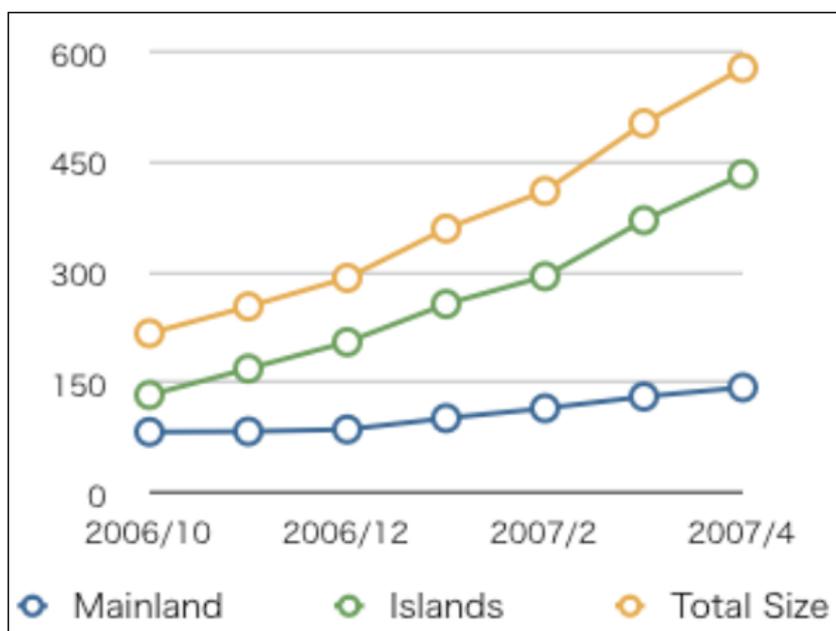
平均年齢
30

Teen Grid平均年齢
15

プレミアム会員と一般のユーザ(無料)との違いは、プレミアム会員は毎月9ドル(3ヶ月:22.5ドル、年払い:72ドル)の支払いをする。通常の無料ユーザは登録時にL\$250を支給されるが、プレミアム会員はL\$1000支給される。またお小遣いとして毎週L\$300が支給され、アイテムの作成、土地所有権などの権利も譲渡される。

一般登録ユーザとアバタの数が同数でないのは、1人のユーザが複数のアバタを所有しているからである。その理由としては

契約時のボーナスを稼ぐのと、複数のアバタを操る事で、SL内のアルバイトでのアルバイト料を稼ぐ為だと考えられる。またアバタの見た目や格好をその日の気分によって使い分けるというユーザも存在する。



登録アバタの年齢分布の通りある。分布として25~34歳の層が最も多く平均年齢は30歳である。TeenGridとはSL上で子供向けに開かれた専用の区画で行動できるアバタである。通常のSLで我々が行動できる区画はメイングリッドと呼ばれる場所である。TeenGridで行動出来るアバタの平均年齢は15歳である。この年齢はアバタの設定年齢であってユーザの実年齢ではない。

またアバタの性別比は男性が56.97%で女性が43.3%である。またアバタの人口が増加するに共がない、経済活動も活

発になりSL内の全体の島のサイズも拡大されている。(右図・単位:km²)2007年4月時点のSLの合計サイズは579.4km²である。これは東京ドーム(46.755km²)で換算すると東京ドーム12個に値する。実際の現実世界と違い容易に島を拡張する事が可能なので今後も増加すると考えられる。

SLにおいて日本国内からの利用者は、世界の中で大きな割合を占めていて、一ヶ月の間に1時間以上SLを利用したアクティブユーザ数は、日本が35,841人(6.94%)で世界では1位はアメリカの156,516人(30.32%)、2位はドイツの40,940人(7.93%)、3位はブラジルの36,999(7.17%)、4位は日本で5位はイギリスの35,012人(6.78%)という現状である。

SLに参入企業の傾向について



現在、SLは世界中の国のユーザによって利用されている。日本においても国内のメディアによってセカンドライフの紹介や企業の参入事例が多く我々の耳に入ってくる。シード・プランニングの調査によれば2007年8月時点では、参入日本企業は85社あり、うち43社は独自に、土地を購入しそこで店舗を展開を行っている。残りの42社は商品・ブランドのプロモーション活動を行っている。業種別に見るとネット関連サービス業が28%、広告業が16%、通信業・製造業・公的機関が各6%を占めている結果が分かった。

具体的には「三越・野村証券・ソフトバンクモバイル・HIS・NTTドコモ・mixi・テレビ東京」など有名企業などが参戦しているが、実際にその島に日

本時間の9時に行ってみたが1人もおらずその島の企業の人も1人もいなかった。個人の感覚ではメディアでSL参入は華やかに演出されているが、実際にログインをして行ってみると誰もおらず、拍子抜けをしてしまう。

ITmediaNewsの記事によると、2007年の初めごろから経済紙には「Second Lifeは世界で何百万人ものユーザが利用して大人気。大企業の参入が相次いでいる。仮想通貨を現金に換え

る仕組みがあり、次世代の新ビジネスが期待出来る」といったやや加熱報道が企業幹部を焦らせ相次いで日本企業の参入が行われたのである。

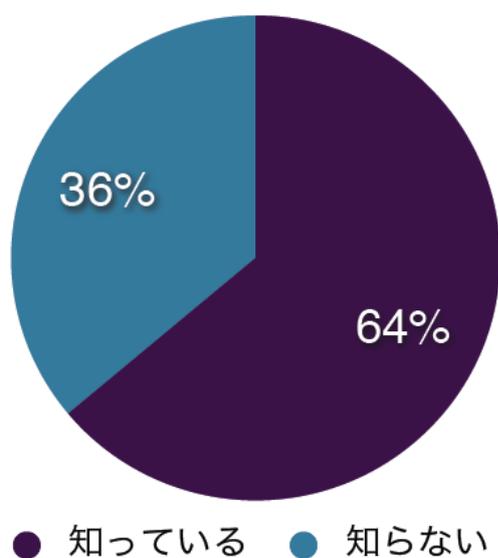
SL自体は欧米諸国では多くのユーザが利用しているが、アジア圏や日本だけで考えると前述したようにアクティブユーザ数すくなく、日本企業が参入しても参入に見合っただけの経済効果が出たかには疑問が残る。SL内の日本企業もサービスが日本語をメインとして展開されていたり、企業名の自体の海外からの認知度が低いこともあり欧米アクティブユーザからのアクセスもない状態が続いている。SLに参入する事での広告PRの側面が強く、今後は管理費のかかる一方で、誰も来ない専用SIM(島)をいつまでも維持し続けるか?が争点になると考えられる。

SL に対しての一般利用者からのニーズ

今後、日本企業や日本の教育期間がこのSLに進出するにあたり、利用者のニーズや傾向を的確に把握しておく必要がある。そこでブログ連動のオンラインリサーチなどを手がける株式会社アイシェアの「セカンドライフ」についての意識調査の結果を紹介すると同時に今後、どのように展開をして行くべきかを考えていきたい。調査期間:2007年7月/調査方法:CLUB BBQ会員限定Web調査/男女比:男51.3%女48.7%

セカンドライフを知っているのか?

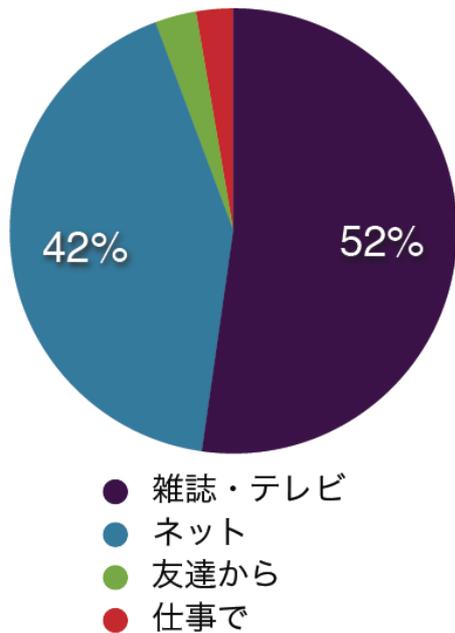
セカンドライフを知っている



セカンドライフを認識していた数は63.9%。その知ったきっかけは「雑誌・テレビ」が52.4%と過半数を越え、「ネット」で知ったという人が42.2%という結果であった。またごくわずかに「知人・友達」から聞いて知ったという人は2.4%と非常に少なく、利用者に利用の理由を調査したところ「話題になっていたから」が72.2%を占めていた。

セカンドライフは他のMMORPGと違いルールがないゲームだけに友人を直接誘っても目的がない為に、友人を誘いにくいという意識がある。またインターネットが普及した現代の日本でも有力なメディアは「雑誌・テレビ」と昔からある媒体が利用者に好まれているのも日本でセカンドライフの利用者率が低い原因となっている。

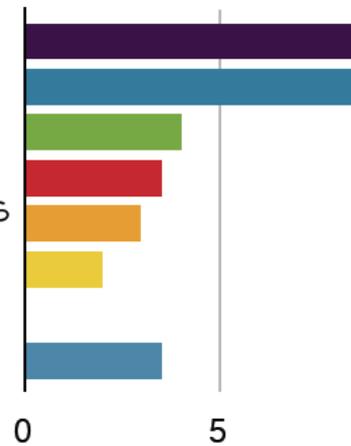
どこで知りましたか?



また利用者の多くが「話題となっているから」という動機付けで利用していることから、話題(ブーム)が過ぎたら利用しなくなる可能性が高い。よって利用者には「継続的な動機付け」が必要であると考えられる。

セカンドライフを利用

- 話題になっていたから
- 新しい事だから
- 誰もやっていなかったから
- 物作りが出来るから
- イベントに参加したかったから
- 友達などの誘いで
- 買い物ができるから
- その他

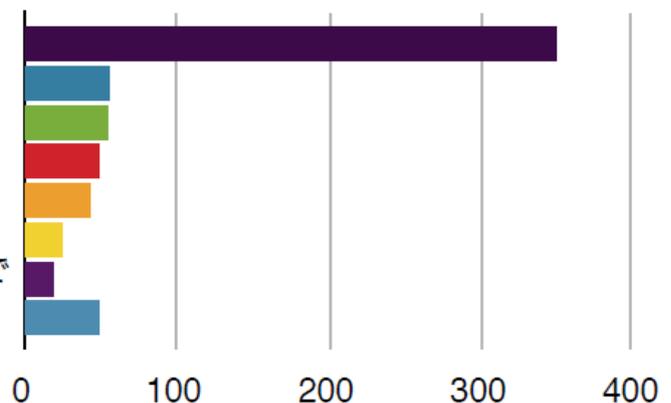


セカンドライフをこれから始める?

この調査の結果ではセカンドライフ利用経験者は3.8%。それ以外の人で始める予定は?という質問に対して現状では「興味がない(76.4%)」という意見が大多数を占める結果となった。その理由として「周りにセカンドライフをやっているが少ない」が20.3%を占め、「パソコンに高いスペックが必要そうだから」が17.4%、「始め方がわからない」が15.3%という結果になった。この結果から完全にセカンドライフに興味を持っていないのではなく、セカンドライフの情報がしっかりと伝わっていない、理解されていないという課題が見える。

セカンドライフを利用しない理由《複数回答》

- 特に興味がないので
- 周りにやっている人がいないので
- 高スペックなPCが必要なため
- 始め方がわからないので
- お金がかかるらしいので
- サイト・システムが英文だったので
- 操作方法が分からないため
- その他



利用しない理由について「特に興味がない」という理由が一番多い。これは逆に言えば興味を持てば「利用してみたい」という意見でもある。その為、利用したいと思わせるような動機付けが重要になってくる。また「周りにやっている人がいない」も同様に何か組織的にこのセカンドドライブを利用する様に促せば解決が出来る。次の「高スペックなPC」については必要な最低スペックについて下記にまとめた。

最低動作スペック

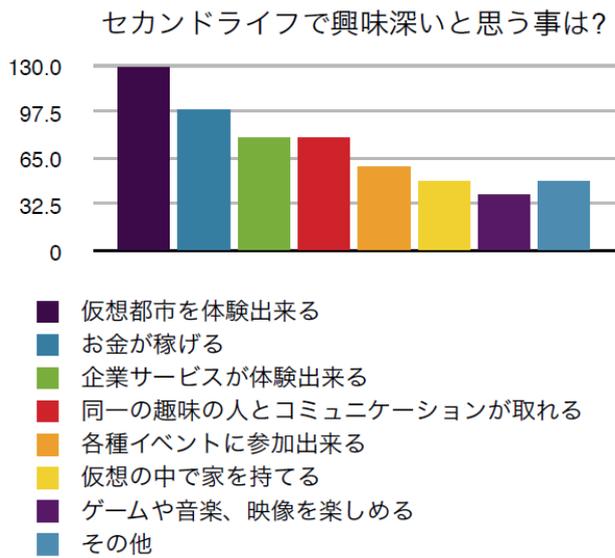
	Win	Mac
CPU	Pentium III 800MHz Athlon, それ以上	G4 1GHz それ以上
OS	Windows XP (Service Pack 2) Windows 2000 (Service Pack 4)	Mac OS X 10.3.8 それ以上
メモリー	256MB 以上	512MB もしくはそれ以上
グラフィック	nVidia Geforce 2 Geforce 4mx それ以上 ATI Radeon 8500 9250 それ以上	nVidia Geforce 2 Geforce 4mx それ以上 ATI Radeon 8500 9250 それ以上
接続速度	Cable もしくは DSL	Cable もしくは DSL

推奨動作スペック

	Win	Mac
CPU	Pentium 4 1.6GHz Athlon 2000+ それ以上	G4 1.25GHz それ以上
OS	Windows XP (Service Pack 2) Windows 2000 (Service Pack 4)	Mac OS X 10.4.3 それ以上
メモリー	512MB 以上	768MB それ以上
グラフィック	nVidia Geforce FX 5600 Geforce 6600 それ以上 ATI Radeon 9600 X600 それ以上	nVidia Geforce FX 5600 Geforce 6600 それ以上 ATI Radeon 9600 X600 それ以上
接続速度	Cable もしくは DSL	Cable もしくは DSL

セカンドドライブを使用するのに最低限のスペックで使用すると、カクカクと動くセカンドドライブに不満を多く持つことが推測される。セカンドドライブに接続するという事はできても快適なコミュニケーションなどは難しい環境にある。現在、市販に販売されているPCのスペックなら十分にセカンドドライブを使用する事ができる。十分なスペックのPCを持ったユーザが多くいるのに利用していないのは、「セカンドドライブ=特別なハイスペックPC」というイメージが強く整理された情報を的確に提供する必要がある。

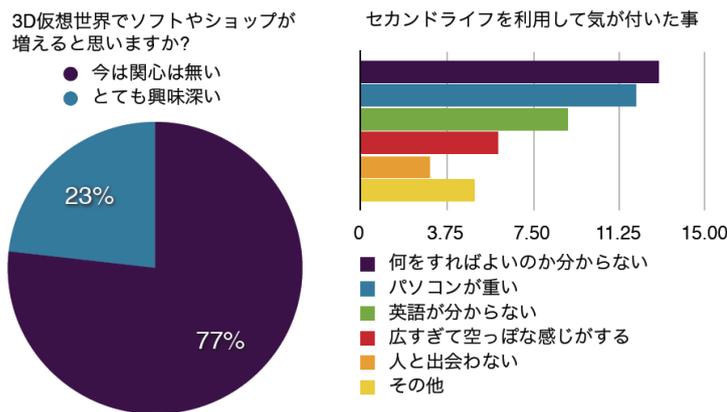
セカンドライフへの興味・関心



セカンドライフを実際に利用してみている興味深く思う事や感想についての調査では、多くの意見としては、1位が「仮想都市を体験出来る」で、2位が「お金が稼げる」、3位4位が「企業のサービスを体験出来る」「同一の趣味の人とコミュニケーションが楽しめる」がランクインしている。仮想空間ならでの仮想都市への観光や無料で体験できるサービスが人気が高い傾向にある。また「今後3D仮想空間を自由に行動出来るソフトやショップが増えてくると思いますが、どう思いますか?」という質問には「今は関心はない」が76.8%

で「とても興味深い」が23.2%という結果になった。

セカンドライフを利用している人に「体験してみて気が付いた事は何ですか?」という質問には「何をすればよいのか分からない」が1位になり2位に「パソコンが重い(動作不安定)」という意見が次に続き、3位に「英語がわからない」という意見がきた。前述したようにセカンドライフでは他のMMORPGと違いゲームの目的がない為、目的がない利用者の立場からすると1位の意見が高いのも理解出来る。



統計データからみえる事

結果、統計データから分かる事は下記の通りである

- ・セカンドライフの認知度は高い
- ・セカンドライフの情報はテレビ、雑誌からが多い
- ・利用目的は特になく、流行や興味があったから利用するという動機
- ・利用目的が明確でないため利用しないというマイナスな動機
- ・セカンドライフでは観光、体験、コミュニケーションに期待

March 8, 2009

という事が分かった。以上のことから今後のセカンドライフ内に参入・企画運営等を行う時の課題としては、利用者に対しての「明確な利用目的・動機」を与えるような工夫が必要であるという事と初回利用者をリピータにするような仕組みが必要である。

法政大学とセカンドライフ

法政大学とセカンドライフ

現在、法政大学でも一般企業と同様にセカンドライフ内に島を購入し「セカンド・キャンパス」の建築を進めている。このレポートでは技術的な事ではなく、「セカンド・キャンパス」の現実化に向けての戦略を述べていく。セカンド・キャンパスを建築するにあたり、前項の統計データの結果から「明確な利用目的・動機」を与える事が重要になる。ここでセカンド・キャンパスの利用者の設定としては、在校生・卒業生・父母・受験生をメインに考える。

利用者に利用目的を与える為には「利用せざるをえない必要性(強制)」「魅力ある豊富なコンテンツ」などが考えられる。セカンド・キャンパスに必要性を持たせる事でアクセスをしなければいけない状態を作り出し、そこで豊富な企画(コンテンツ)や情報を配信して行く事で、セカンド・キャンパスをただの観光名所にするのではなく、リピーターを増やすに繋がると考える。

では、豊富な企画(コンテンツ)とは何がよいのだろうか?考えられるコンテンツとして「大学紹介(オープンキャンパス)・各学部授業配信・ゼミ研究結果発表・六大学野球配信・入試センター・掲示板(憩いの場)」などがある。最終的には将来的に導入が検討されているSakaiとの連携したサービスを展開など考えられるが、ここではセカンドライフ単体だけのコンテンツについて考えて行きたい。

企画(コンテンツ)について

セカンドライフの特徴の1つとして仮想空間内に自由に建築物を作る事ができる点がある。セカンドライフ内であれば物理的な法則や建設費用などを気にする必要がない。その点を考えるといかに独創的で魅力的な建物や企画(コンテンツ)を展開していくかがポイントになる。

企画(コンテンツ)	内容
大学紹介	大学の各種情報・特徴・イベント情報
各学部授業配信	12学部(2008年1月現在)の特徴ある授業を配信
ゼミ研究結果発表	ゼミ単位で研究の成果を世界に配信
六大学野球配信	神宮球場に行かなくても観戦できるスタジアム
入試センター	仮想空間上で学生や職員が直接、サポートできる環境
掲示板	休講情報から入ゼミ情報まで幅広く掲示

利用者が魅力を感じ、動機付けをされ再度利用を促す為には常に変換を続ける企画が必要と考える。上記の内容、特に六大学野球配信などは六大学以外の大学には真似出来ない法政大学独自の個性を出す事が出来ると考える。実際にセカンドライフ上で動画を配信(上映)することは可能なので、いかに実際の野球の試合とタイムラグをなくして配信するかという

問題は残るが、セカンドライフ上での配信が現実化かすれば、アバタが装着する法政大学応援グッズなどを揃える事で、話題性や利用者の動機付けにも繋がる。この他にも野球以外で法政大学が他大学に模倣出来ないオリジナルなものとして「能楽」があげられる。能楽であれば海外からの注目も高く多くの訪問者を呼ぶ事が出来ると思う。

将来へ

三次元の仮想空間は今後、主流の技術になると考えられる。インターネットの普及により我々は家にいながらして、地球の裏の人とコミュニケーションを取っていたり、行った事のない場所の風景写真を瞬時に調べる事ができるようになった。近年、SNSやブログなど始めとした情報を引き出すだけでなく発信する時代になり、双方向の情報のコミュニケーションが行われる様になった。SNSやブログなど文字ベースのコミュニケーションであつたが三次元仮想空間の登場により文字だけでなく、聴覚、視覚によるコミュニケーションが可能になり利用者は「空間」を意識することができる。

セカンドライフはただのMMORPGという枠組みを越えた新しいコミュニケーションツールであると思う。特に通信教育などのe-ラーニングのカリキュラムを受けている人は、e-ラーニングに起こりうる利用者の「孤独感」をセカンドライフで緩和する事も出来るのではないだろうか。また三次元という事を利用すれば、不動産の物件を実際に内見しなくても仮想空間内に街並を作りモデリングされた物件を内見のサービスを提供するという事もでき、このような仮想空間を巧みに利用すればあらゆるビジネスチャンスをつかむ事が可能である。

現在、法政大学では「セカンドライフCG学生プロジェクト」を立ち上げ学生の視点からこのセカンドライフについて研究を進めている。技術面については多くの関連ホームページや参考文献が出ている。その中で重要なのは技術力ではなく企画力の方であると考え。特に統計データから分かる様に、セカンドライフについての認知度は高いのに対して利用者が少ないのは「利用目的がない」という事が最大の要因である。逆に言えば「利用目的を与える」ことが出来れば利用者は増えるのではないだろうか。また世界のどこからでもアクセス可能なセカンドライフだからこそ、この仮想空間を使用して他国の大学生や人々と研究について発表を行う事で国際交流も可能になると思う。

【参考文献】

Wikipedia	http://ja.wikipedia.org/wiki/メインページ
はてなダイアリー	http://d.hatena.ne.jp/
Second Life	http://secondlife.com/
アイシェアサービス	http://www.isharel.com/

はじめに（2008年度卒業生）

昨年度、はじめて卒業文集としてまとめましたが、多くの部数を刷ることができませんでした。本年度は、昨年度卒業文集とまとめて印刷し、より多く刷り広く発表していきたいと思います。本年度は、2人の卒業生がおります。さらに、論文として価値の高い文章となっていると思います。

本年度は、ゼミ生総勢3人というとてもコンパクトなゼミでした。しかし、各人が考え発表するという、とても充実したゼミを展開することができたと思います。

平成21年3月春
児玉靖司

児玉 育也

経営学部 経営戦略学科 05F1727

Blender for Personal Use

Blender にいたるまで

私は2年前、パソコン・インターネットを媒介とした色々な研究を行っているという兎玉ゼミ（以下、ゼミ）に興味を持ち試験を受けたところ、合格をもらったのでこちらのゼミに参加させてもらうことになった。

私がゼミではじめに研究したことは、「情報とは何か？」ということである。情報とは何か、それは私にとっていかにも定義しがたいものであった。なぜなら、そんなことを考えたことが一度もなかったからである。情報の定義を考えるために何か本を読もうと考えた私は、先ずインターネットで「情報」をキーワードに検索をかけるが、なかなか読みたいと思う本が見当たらなかった。そのため本屋に出かけることにした。探し回った結果みつけたものは日経文庫「関口和一（著）情報探索術」であった。私はこの書物から情報の定義を考えはじめたのである。『情報探索術』を読んだ結果得られた情報は以下のとおりである。

「情報」という言葉の語源は、森鷗外がドイツの軍事思想家であるクラウゼヴィッツの軍事戦略の古典「戦争論」を翻訳した際に、ドイツ語で「知らせ」を意味する「Nachricht」）という言葉を訳すために作った造語が最初だといわれています。しかし、実際には明治初期にフランスの軍事書物を翻訳する際の訳語に使われており、敵の状況を知らせるという意味の「敵情報知」という言葉がもともとの語源だったようです。「情報」はそれを短く略した言葉で、いずれにせよ軍事用語として登場したものでした。(1)日本で「情報」が今日のような一般的な「知らせ」という意味で使われ始めたのは大正時代に入ってからです。(2)ところが、日本では「情報」にはもう一つ別の意味があります。例えば「情報とは情けに報いると書く」という…(中略)…営業活動などで相手から重要な情報をもらうためには、相手のところに頻繁に通うなど、人の気持ちに訴えなければならないというわけです。つまり、この場合は、情報は「親しい人間にしか教えられない特別な事柄」を指します。(3)

(『情報探索術』より)

上記の内容には、情報の2通りの性質が述べられている。すなわち、情報には「ただ単に知らせ」というとらえ方と、「親しい間柄にしか伝わらない特別な事柄」という二つの側面があると考えることができるのである。

そして、この他にも情報についての研究と、それから発展してネット社会についての研究などを続けたのである。それらの研究の後、私はBlenderの着手を始めたのである。Blenderについて上記では一切触れていないので、これを研究しようと思った理由を述べていこう。

ゼミの先輩たちの中に Three Dimensions Computer Graphics（以下、3DCG）を研究している人たちがいた。彼らは3DCGの研究とその応用としてセカンドライフでの活用を研究していた。彼らの作品は私にとって見て楽しいものであり、またその作品の見事さ（これは個人の技術力が十分にあったからであるが）に、3DCGの制作は個人にとっても十分に手の届

くとても身近な存在になっているのかと驚嘆し、実際のところを自分自身でも知りたくなったのである。

はじめに

今、世の中には多くの CG (2DCG、3DCG 問わず) が取り入れられている。映画やテレビ、法人、個人のサイト、紙媒体の情報誌にも多くの CG が使われている。今、いたるところに CG は使われており、それを見ないことは現代ではほとんどないように感じる。しかし、3DCG の分野においては「ひと昔前までは 3D-CG の表現力があまりたかくなかったために、「タイトル」や「特殊効果」「報道用の説明アニメーション」に使われるくらい」(4)であった。「それが今日では技術が進歩し、映画として鑑賞に堪えるクオリティのものが作れるように」(5)なった。古くは『トイストーリー』から始まり、一番身近な例でいえば、『バイオハザード-ディジェネレーション』をあげることができる。『トイストーリー』が放映されたのは、95年11月であり、それから12年経過して今日にいたっている。この12年の間に、さまざまな 3DCG 制作ソフト (以下、3DCG ソフト) が個人向けにも販売されるようになった。「しかし、高いクオリティの作品を作るには、ツールにもそれだけの性能が必要です。十分な性能をもつアプリケーション・ソフトは個人的な趣味として買うには高価なものが多いのが現状」(6)と述べられているように事実、『Maya』、『3dsMax』、『Shade9』、『CINEMA4D』などの多機能かつ高性能なソフトが数万~十数万で個人向けに販売されている。しかし今日、Blender をはじめとするフリーソフトの台頭、低価格な 3DCG ソフトの台頭によって、私が金銭面で 3DCG 制作に携わること自体に抵抗をもつことはなかった。

Blender を振り返る

上記で述べた通り、金銭面からみて 3DCG ソフトに対する抵抗はなかったもので、私は Blender を用いて 3DCG の制作を始めたのである。以下では、過去に取り組んでいた Blender の活動を少し述べていく。

私が Blender で制作を始めるにあたり、用意したものは「Blender Beginner's Bible」(以下 BBB) といった本一冊と、「Blender.JP」といった一つのサイトだけであった。私はまず、BBB に載っているサンプル (ひとつめは人参、次に葉っぱ、そして魚をモデリングしていった。) を教本の指示通り作ってみることにした。どんな経緯にせよ、Blender で作品を作ることが手っ取り早くこのソフトになれると思ったからである。私は上記の手段で作品作りを始めたが、BBB の文章で理解に苦しむところが多々あった。なぜなら、私の使用していた Blender のバージョンと BBB で説明されているバージョンは異なっていたため、GUI (Graphical User Interface) を含めたさまざまな仕様が若干変更されていたからである。これは私が Blender の研究を始める前から口コミで言われていた Blender の特徴 (良い点も問題点も含む) である。つまり、独自性の強い GUI、解説書の少なさ、そしてバージョンアップの早さである。今現在この特徴は良い方向へ舵を切りつつあるように考えられる。

その理由として、まず GUI に大きな変化が生じたことを挙げる。これはすなわち、各コマンドのショートカットを一覧できるようになったことである。Blender には、50 個を超えるホットキーが存在しているためすべてを把握するのは正直大変である。したがって、これは Blender を始める人にとって非常に有益なサポートとなるだろう。次に解説書の少なさについて取り上げる。Blender Foundation の公式サイトで調べればわかるが、現在アメリカでは 10 冊以上の参考書が販売されている。ただし、これらの日本語訳版は一切発売されていない。日本語の参考書はアメリカで販売されている種類には及ばないが、上記で述べた『BLENDER Beginner's Bible』、他にも『はじめての Blender』、『3D キャラクターアニメーション Blender』、そして一番新しい出版物では『物理シミュレーション Blender』が 2008 年 12 月 19 日に発売されている。上記のように日本国内外問わず販売されている図書の過半数が、私が Blender に着手し始めた後に発売されている。このことは解説書が少ないという従来の Blender の問題点を見事に解決しているのである。

Blender における 3DCG 制作の基本操作

ここでは Blender を 3DCG ソフトとして使ってみようとする人に、私が経験したなかで比較的分かりやすいと思った操作で、ある程度知っておいたほうが良いと思うことを少々述べていこう。

はじめに述べるのは、「Curve」オブジェクトを用いたモデリングである。モデリングとは、ある物体の形状を作成することである。このモデリング方法は、指定した場所を中心に、線画に 360 度以内の回転を加え (X 軸、Y 軸を指定可能)、立体画に仕上げる手法である (※1)。これは、線画の操作が非常に容易であるため、初心者でも十分に使いやすい機能である。ただ、注意すべきポイントとして、線画で描いた形どおりに立体が仕上がるわけではないということを覚えておくべきである。

もう一つの手法は、下絵を基に、平面画をつくり、それから立体化、そして細部の調整を行っていくモデリングである。これは、「Curve」オブジェクトを用いたモデリングより、多少操作は難しくなるが、より柔軟に色々な形をつくることのできるというメリットがある (※2)。この手法を用いる場合は、より多くの下絵があるほうが、下絵により近づいたモデリングが可能となる。

このほかに、モデリングの手法ではないが、モデリングした物体(オブジェクト)をより現実的な物体に見せる方法としてマッピングといった手法などがあり、これらがまずは 3DCG 制作の基本となるのである。

Blender によるアニメーションの導入

Blender のアニメーション機能も試してみたので、そのことについても多少述べていきたい。ただひとつ注意点として、すべての 3DCG ソフトにアニメーション機能がついているわけではない。私が試しにつくったアニメーション、それは擬似モーション・キャプチャーと呼ぶようなものである。モーション・キャプチャーとは、本来は

関節部に加速度センサーを付けた被験者に動作を行なってもらい、被験者の動作をデジタルデータに変換してコンピュータに取り込むこと。取り込まれたデータを元に 3 次元グラフィックスのキャラクターの動作を構成していくことで、リアルな動きを持つ CG キャラクターを作成できる。(7)

(IT 用語辞典より)

といったものである。今回私は卓球のプレイ動画を下絵にし、その実際の卓球選手の動きを真似て 3DCG 上のキャラクター (rigging 済み) に反映したのである。(※3) この作業時、下絵となっている動画は FPS (Frame Per Second) 30 (一秒間に 30 枚の画像を表示する) で作業を行っていた (数値は変更可能)。したがって、フレームを適当に飛ばし、各フレームでキャラクターの動作を下絵の動画と合わせていけば擬似モーション・キャプチャーとなるのである。このときの注意点として、フレームを飛ばしすぎると、その間の動きはコンピュータが独自に最適と判断した動きをとるため、自分の予想していなかった動きをとる場合がある。そのため、フレームの飛ばしすぎに注意することと、毎回設定した通りに動いているかをチェックする必要がある。

Blender で動画作品を作る

アニメーションについては上記で述べたとおりである。以上の手段を用いて実際に一つの作品を作ってみたので、そのことについても多少述べていく。

今回は「三次元の動画空間を楽しむ」といったコンセプトで作品づくりを行った。オブジェクト (モデリング及びマッピング済み、Bmania 氏の H.P.)、自分で作業を行ったところは、カメラワーク・ライティング、及び特殊効果の付与と背景である。まず、私はパーティクル (粒子の動きをシミュレーションする機能) を使用した。そして次に背景を作成し、またオブジェクトに動作を加えない代わりに「Build」アニメーションといった特殊効果を付けた。そののち、カメラワークを行い、カメラの動きに合わせたライティング作業を行った。このとき、非常に苦労したことがある。それは、アニメーションのつなぎの部分である。私は動画を完成させるにあたり、Blender だけの使用では限界があるように感じたのである。

さて、Blender から少し話がそれるが、私は法政大学主催の 2007 年度第 3 回デジタルコンテンツ・コンテストの優秀作品を展示しているブースに立ち入ったことがある。そこで表彰されていた動画の作品は全て Maya を使用していたが、そのすべての作品にはもう一つの共通点があった。それは、3DCG ソフト一本で作られていたのではないということである。

作品のすべてに、他の音響用ソフトや、特殊効果を用いるために使用したと思われるソフトが使われていたのであった。これは当たり前のことかもしれないが、映像制作初心者の私には驚きであった。そして、私も自分の動画を完成させるにあたり、Blender 以外のソフトの必要性を知ることになったのである。そこで私が見つけたのは「Textaizer」といったフリーソフトである。これは文字だけで、一枚の静止面を作ることができるソフトである(※4)。また、Windows Movie Maker (以下 WMM) を用いてアニメーションのところどころに切り替え効果を付けたことによって、一つの動画作品を完成させたのである。

まとめ

第一章のまえがきの部分でも述べたとおりであるが、私は近くの人たちから多大な影響を受けて、3DCG をずいぶん身近に感じたものである。そこで私は 3DCG ソフトが私自身(3DCG 初心者の個人)にとっても身近なものであるのかどうかを知りたくて研究を行った。

研究の結果、知りえた情報は以下の通りである。Blender は私とその研究を始めた当時はまだ 3DCG に挑戦を試みるのが難しい環境であった。その原因としては 3DCG ソフト自体に使用者の専門知識を要求する度合いが大きいといった技術的問題があったからであり、またそのソフトを囲む環境もわずかしかできていなかったからである。しかし、今日までの短い期間に Blender を囲む外部環境(参考書・サンプルなど)は以前よりはるかに向上し、またホットキーの表示などソフト自身の内部環境の変化もあって、Blender は 3DCG 初心者の個人また私自身にとっても一段と身近なものとなってきているのである。

引用箇所一覧

- (1) 関口和一 『情報探索術』, 日経文庫, 2006, p. 14
- (2) 関口和一 『情報探索術』, 日経文庫, 2006, p. 15
- (3) 関口和一 『情報探索術』, 日経文庫, 2006, p. 15
- (4) 山崎聡 『はじめての Blender』, 工学社, 2008, p. 22
- (5) 山崎聡 『はじめての Blender』, 工学社, 2008, p. 22
- (6) 山崎聡 『はじめての Blender』, 工学社, 2008, p. 22
- (7) 株式会社インセプト 「IT 用語辞典 e-Words」

<<http://e-words.jp/w/E383A2E383BCE382B7E383A7E383B3E382ADE383A3E38397E33838E383A3.html>>(最終アクセス 2008 年 12 月 15 日)

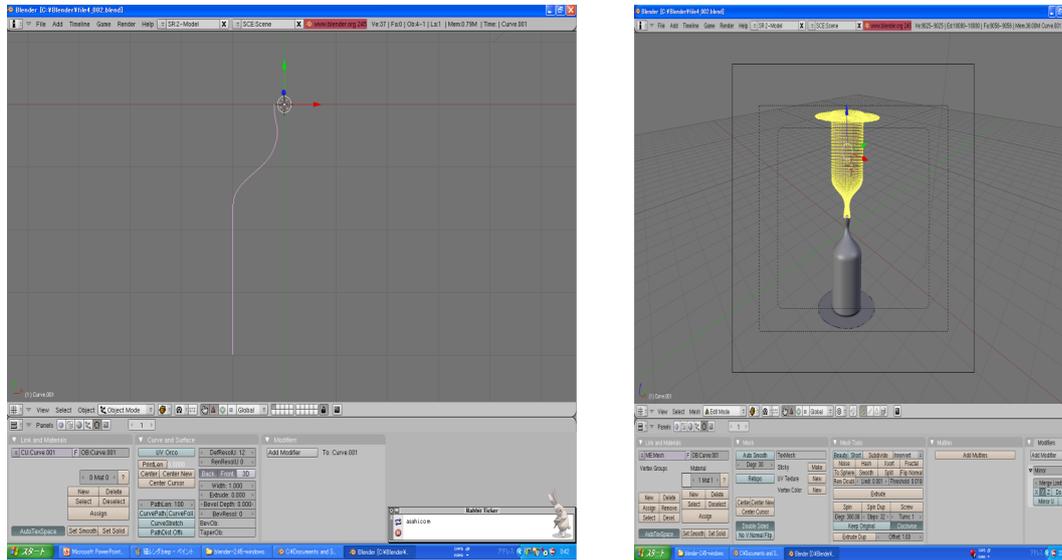
参考文献・サイト一覧

- 関口和一 『情報探索術』, 日経文庫, 2006
 山崎聡 『はじめての Blender』, 工学社, 2008
 田崎進一 『BLENDER Beginner's Bible』, 毎日コミュニケーションズ, 2006
 Tony Mullen 『Introducing Character Animation with Blender』, Wiley Publishing, 2007

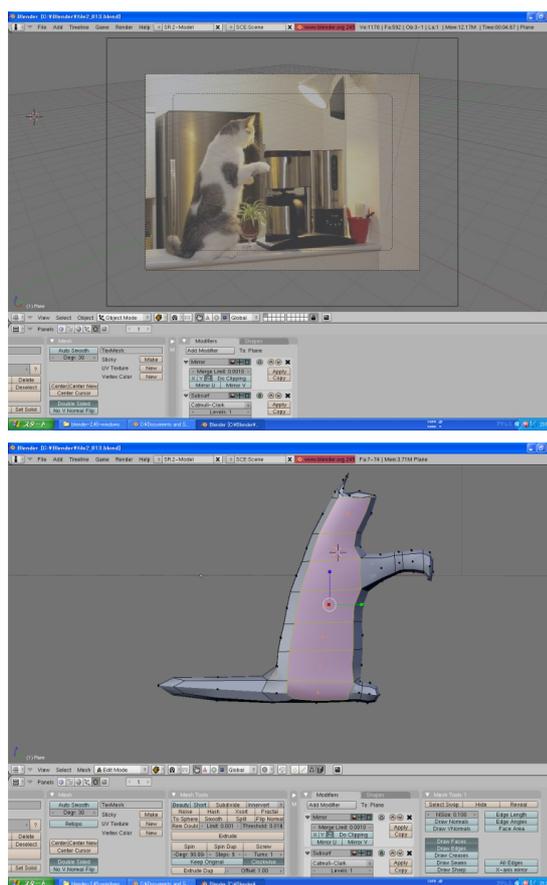
株式会社インセプト「IT用語辞典 e-Words」 < <http://e-words.jp> >
 yamyam「blender.jp」 < <http://blender.jp/> >
 Blender Foundation「Blender」 < <http://www.blender.org/> >
 bmania「Bmania HP」 < <http://bmania.michikusa.jp/> >
 APP Helmond「Mosaizer」 < <http://www.mosaizer.com/index.htm> >

※図一覧

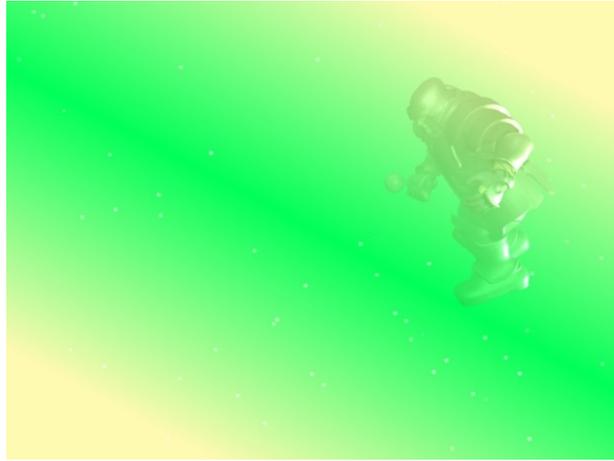
※1. 「Curve」オブジェクトを用いたモデリング作業の工程



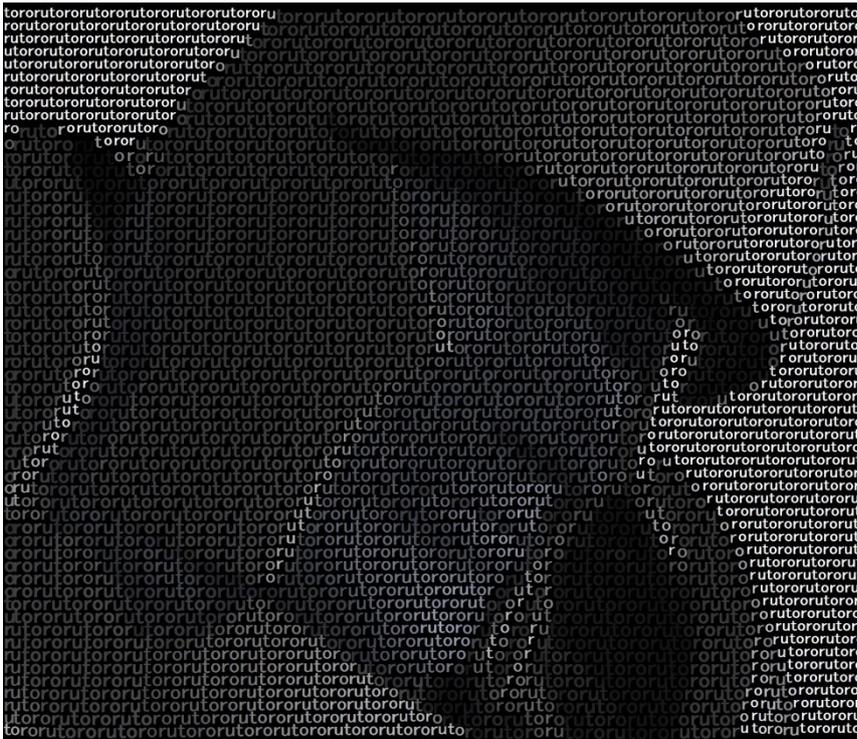
※2. 下絵からアウトラインを作成する作業工程



※3. 疑似モーション・キャプチャーによる卓球選手の動作を一部抜粋



※4. Textaizer を使用し、文字で静止画を描いたサンプル



田村 俊一

経営学部

無料ブログサービスを利用した SEO 対策の有効性の実証

はじめに

近年、インターネットの利用は社会に広く浸透している。平成 17 年度末時点で過去一年以内にインターネットを利用したことがある人は推計 8,529 万人、実に 66.8%の人が利用した試算である。企業の 85.6%が自社サイトを開設しており、2,000 人以上の企業に限れば開設率は 99.1%とほぼ 100%に近い数字である。個人のサイト開設も進んでおり、なかでも無料で開設ができるブログサービスの利用者は平成 18 年時点で 868 万人に達している。全世界では一日当たり 12 万の新しいブログが作成されている。そして世界で最もブログを書いているのは日本人であり、日本語で書かれたブログは世界中のブログの 37%を占める。

インターネットの普及は企業に、そして個人にサイト開設を通じたより積極的な情報発信を促した。ネット上に存在する情報量は年々増加している。そんな無数に散らばる情報から目当ての情報を探し出すには、検索エンジンの利用が効率的である。インターネット利用で検索エンジンを利用したことのない人はほとんどいないはずである。インターネットにおいて「検索」は欠かすことのできない地位を築いていると言える。その「検索」という行動に目を付けたのが「SEO 対策」というサービスである。SEO 対策とは一口に言えば検索結果での上位獲得を狙う技術である。検索エンジンは今や自社サイトを持つ企業にとって貴重な顧客窓口となっている。企業は一人でも多くのお客を自社サイトに呼び込みたいと考えている。そこで検索エンジンからのアクセスを増やすために重要となるのが検索順位である。そして、その検索順位に影響を及ぼすことのできる技術が SEO 対策なのである。

SEO 対策には様々な手法の対策が存在しており、文献も数多く存在する。しかし効果的な対策について論じる文献は多い反面、実際にそれを実証して見せる研究は少ない。そこでこの論文では近年利用の拡大が目覚ましい無料ブログサービスを利用して、どの様な SEO 対策が効果的であるのかを実験の実験から明らかにし、有効な SEO 戦略を模索する。

まず始めに SEO 対策、検索エンジンの基本的概念について触れ、以上をふまえた上で検索順位決定の評価要素と要素毎の SEO 対策の類型について述べる。続いて具体的な実験内容について論じ、実験結果と考察を述べ、まとめとする。

SEO 対策について

ここでは SEO 対策の基本的概念とその成り立ち、メリット・デメリットについて説明する。

SEO 対策とは

SEO 対策 (Search Engine Optimization) とは、検索エンジンの検索結果ページにおける表示順位 (検索順位) 上位に自サイトが表示されるようにサイトに対策を施すこと。あるいは、その対策そのものを指す。

消費者行動の変化と SEO 対策の誕生

SEO 対策の誕生には消費者行動の変化が大きく関係している。インターネット登場以前の社会では、消費者の購買プロセスはサミュエル・ローランド・ホールが提唱した AIDMA の法則に従っていたと言われる。AIDMA の法則では消費者が財・サービスの購入に至るまでの段階を下図左の様に分析している。しかしこの AIDMA の消費プロセスはインターネットの普及、検索エンジンの台頭によって AISAS の消費プロセスへと変化したとされる。AISAS の法則では消費者が財・サービスの購入に至るまでの段階を下図右の様に分析する。



消費者の購入に至るまでに“検索”という段階が加わっていることが分かる。AISAS の法則では消費者は興味を持った商品に対して一旦検索を通じてアプローチするのである。消費者を取りこぼすことなくサイトに誘導し購入につなげるためには、検索エンジンの存在は無視できない。ここで、検索エンジンを活用して集客を行う検索エンジンマーケティング手法の一つとして SEO 対策は誕生したのである。

SEO 対策のメリット・デメリット

SEO 対策のメリット

SEO 対策の目的は検索結果での上位獲得にある。上位獲得サイトは以下のメリットを得る。

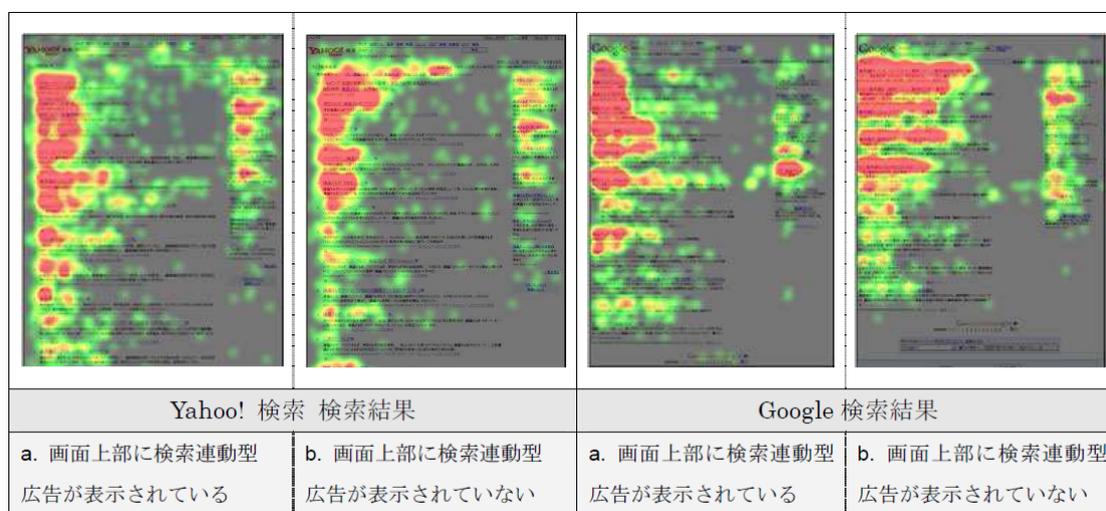
高い集客力

1 つめのメリットは高い集客力である。検索という行動が習慣化した現在、検索エンジンは非常に多くのユーザを抱えている。2007 年 5 月時点で一ヶ月あたりの検索エンジン利用者数は Google が 2,382 万人、Yahoo! が月間 4,044 万人である。しかし、検索エンジンに登

録された全てのサイトがこの膨大な検索エンジン利用者のアクセスを獲得できるわけではない。検索エンジンからのアクセスを獲得できるのはごく一握りの検索上位サイトのみである。検索上位サイトが検索エンジンからのアクセス獲得に有利であることの論拠として、以下のような実験がある。

18歳以上の検索エンジン利用者500人を対象として行われたオプトとクロス・マーケティングの検索エンジン利用状況実態調査では、検索結果閲覧ページ数は平均で3.6ページという結果が出ている。Googleのデフォルト設定では検索結果ページに検索にヒットしたページが10件表示される。この実験から分かることはユーザが検索結果で実際に閲覧するのはわずか上位36サイトに過ぎないということである。

加えてアイレップ SEM 総合研究所と JMI が Google と Yahoo! の検索結果画面での視点データを収集したアイ・トラッキング調査（下図）によれば、検索上位サイトほどユーザに注視される傾向があるという。画面の下の方に表示される低順位サイトよりも、上の方に表示される高順位サイトの方がユーザに注目されやすいということである。



※検索結果画面は、2006年9月29日～10月2日時点の画面となり、現状の画面とは異なっております

赤外線を対象者の目の網膜に照射し、その反射を利用して画面上の視線の動きを測定している。

(※図： Google、Yahoo!検索結果画面での視点データを収集したアイ・トラッキング調査、アイレップ SEM 総合研究所と JMI)

インターネットコムとインフォプラントが行った検索エンジン利用者向け調査でも、検索結果の上位に表示されることでサイトのクリック率が高まると答えたユーザは87%を占めた。クリック率が高まる理由としては、一番目に付く、上から順番に見ていくから、上位にある方が重要性が高い気がする、といった意見があがっている。また8割のユーザは検索結果の中から訪問するサイトを選ぶ際に、サイトの順位を「かなり重視する」「まあまあ重視する」と答えている。

検索上位サイトに対してユーザは好印象を抱き、なおかつ検索結果上位に表示させることでサイトのクリック率も高まる。SEO 対策により検索上位にサイトを表示させることができれば、多くの検索エンジンユーザのアクセスを獲得できる。つまりは高い集客力というメリット得ることができるのである。

需要者への効率的な集客

2つめのメリットは需要者に効率的な集客が行える点である。検索キーワードという形で表現されるユーザの情報を探したいという意図の中には、製品・サービスに対する直接的な需要が含まれている。SEO 対策はその検索意図にズバリターゲティングできる点においてテレビCMやバナー広告といった他の宣伝手法に比べて優れているのである。

例えばテレビCMの場合を想定する。バラエティ番組を見ているときに携帯電話や保険のCMが流れたとしてもバラエティ番組と携帯電話、バラエティ番組と保険には直接的な深い関連性はない。そのため携帯電話の購入を考えているユーザや保険の加入を考えているユーザ、すなわち需要者に集客を行えているのかは怪しいところである。またバナー広告にも同様のことが言える。スポーツニュースを読んでいるところに携帯電話や保険のバナー広告が表示されても、その効果の程は怪しいだろう。

一方、SEO対策では携帯電話についてのサイトであれば「携帯電話」や「ケータイ」と検索された時、保険についてのサイトであれば「保険」や「自動車保険」と検索された時に自サイトが上位表示されるように対策を施す。前者の例では「携帯電話」や「ケータイ」の検索キーワードで検索をする人は携帯電話を買いたいから情報を探している人が大半だろう。つまりSEO対策により検索上位に自サイトを表示させれば、それだけ商品の直接的な需要者からのアクセスを得やすくなるのである。SEO対策はユーザの検索意図を汲み取り、需要者に対して効率的な集客を行えるというメリットを持っているのである。

高い費用対効果

3つめのメリットは費用対効果の高さである。バナー広告やテキスト広告等の広告枠は、表示1回あたり〇円、月額固定で〇円、あるいは1クリックあたり〇円という形式が一般的である。そして契約が満了すれば広告は表示されなくなる。ここでは無論、広告が表示されている限定された期間にどれだけ多くのユーザの目に触れ、どれだけユーザがサイトにアクセスし、どれだけユーザが顧客に転換されるかが勝負となる。

一方でSEO対策は表示単位や月単位で掲載費用が掛かる広告とは全くの異質である。どれだけ表示されようがどれだけクリックされようが、それに対するコストは1円も発生しない。SEO対策業者に発注すればそれなりのコストは掛かるが、SEO対策は広告ではないので契約満了後すぐに表示されなくなるというものでもない。それはSEO対策が単なる広告ではなく、検索にヒットしやすくなるようサイトの基盤構造の最適化であることに起因する。そのため

効果が一定期間に限定されるものではなく、費用対効果が高いというメリットを持つのである。

ブランディング効果

4つめのメリットは検索上位獲得によるブランディング効果である。ブランディング効果とは自社ブランドを高める効果のことで、SEO対策によって得られる中長期的な効果の一つである。自サイトが検索上位に表示されていれば検索結果に露出する機会は自然と増える。結果、自サイトは継続的にターゲットユーザの目に触れ続ける。そうして得た認知度は自サイトの信頼性を高め、ネット上でのブランド力の強化につながる。

インターネットコムとインフォプラントが行った検索エンジン利用者向け調査によると、検索結果で上位に表示されるサイトに対する印象は「良い」と「どちらかといえば良い」という好意的な意見が合計で全体の約8割占めている。SEO対策により検索上位を獲得すればユーザからは好印象を受け、ひいては自サイトのブランディング効果というメリットが得られる。

SEO対策のデメリット

SEO対策のデメリットは効果が出るまでの時間の長さである。バナー広告やテキスト広告は購入し掲載が開始されれば、直ちに自サイトへアクセスを誘導することができる。しかしSEO対策の場合はそうはいかない。サイトに施した対策の効果が検索エンジンに反映されるまでには時差が存在する。そのためSEO対策を行っても直ちにアクセスを誘導するというのは困難である。バナー広告やテキスト広告と同じようにクリック単価や獲得単価等を指標に費用対効果を考えるのであれば、半年から一年単位で戦略を立てる必要があるだろう。

検索エンジンについて

ここではSEO対策の対象となる検索エンジンについて説明する。

検索エンジン市場とアルゴリズム

日本国内検索エンジンシェア

下図は日本の各検索サイトやポータルサイトでどの検索エンジンの技術が利用されているかを示した検索エンジン相関図である。



(図：検索エンジン相関図 2008年12月版，アイレップSEM総合研究所調査)

この図から分かりますとおり、日本の検索エンジン市場はGoogle、Yahoo!がほとんどのサイトへ検索エンジンの提供を行っている。そこにMSNが若干加わるので実際はGoogle、Yahoo!、MSNの三大検索エンジンと言えるだろう。この三大検索エンジンで日本国内検索エンジン市場のシェア96.1%を占めている。

検索エンジンアルゴリズム

ここでは三大検索エンジンの中でも2つだけで日本国内シェアの93.6%を占めるGoogleとYahoo!に注目し、検索結果を左右するアルゴリズムという機械的処理に言及する。両者は各々のアルゴリズムについて以下のように説明している。

Googleのアルゴリズム

Googleの検索結果の順位は、エンドユーザの役に立つ正確な検索結果を提供できるように、さまざまな要素に基づいて決定されます。

(Google ウェブマスター向けヘルプセンターより)

Yahoo!JAPANのアルゴリズム

検索結果に表示される順は、ウェブページのテキスト、タイトル、説明文、関連リンク、ページの特徴などを分析に基づき、検索キーワードに対する関連性が高い順に表示しています。この順序はYahoo!検索が利用している検索エンジン用ロボット独自のデータやアルゴリズムに基づくもので、詳細は非公開とさせていただきます。

(Yahoo!JAPAN 登録サイト検索ヘルプより)

以上のように両者とも全体的な意向を示すのみでアルゴリズムの詳細は明らかにしていない。アルゴリズムは検索品質との関係が深いので、競合他社による模倣や悪用を避けるためと思われる。

両者に共通して言えるのは、どちらもロボット型の検索エンジンであるという点である。ロボット型検索エンジンとは名前の通りクローラ等と呼ばれる検索ロボットが自動的に Web ページを見つけてデータセンターに登録し、アルゴリズムに基づいた順位付けを行う検索エンジンである。対して人の手によって登録が行われ、カテゴリ毎に細かく分類されている検索エンジンはディレクトリ型検索エンジンと呼ばれる。

Yahoo! は 2005 年までディレクトリ型検索エンジンを優先していた。そうした経緯のためか、ロボット型検索エンジン移行後も Yahoo! 公式ディレクトリサイトである Yahoo! ディレクトリへの登録の有無が検索順位の決定に少なからず影響しているという意見がある。ディレクトリへの登録は人の手による審査を経る。そのため検索順位に人為的な要素が介在してしまうのを避けられない。一方の Google では、公式ディレクトリサイトである Google ディレクトリが存在するものの、Yahoo! と比較するとディレクトリとの関係性は希薄である。

この論文では検索エンジンのアルゴリズムに対する効果的な SEO 対策を模索することを研究テーマとしている。順位決定から不確定な人為的な要素をなるべく排除することのできる Google の方が、対策の効果測定の正確性から研究対象に適していると判断した。以上の理由から、以降 Google に焦点を絞って論じていく。

Google について

Google とは

先に述べたように Google はロボット型検索エンジンである。Google はサイト登録を Googlebot というクローラ (スパイダー) によって行う。Googlebot という名前はクローラのユーザエージェントに由来する。Google では Googlebot による自動インデックスが終了した後、アルゴリズムに基づいた検索順位が算定される。

ロボット型検索エンジンの仕組み

ロボット型検索エンジンが Web ページに登録してから検索結果を表示するまでには「クローラ」「インデックス」「クエリプロセス」という 3 つの段階が存在する。

第一段階「クローラ」

検索エンジンは Web ページの存在を見つけるためにクローラというロボットプログラムをネット上に巡回させる。クローラはページからページへとリンクを辿って移動し、新しく作成されたページを発見する。

第二段階「インデックス」

クローラが発見した新しいページの情報を検索エンジンのデータベースに格納する（また、その格納したデータ自体もインデックスと呼ぶ）。

第三段階「クエリプロセス」

検索キーワード（クエリ）と関係のあるページをインデックスした情報の中から呼出し、アルゴリズムに従って検索キーワードと関連性の高い順に表示させる動作が発生する。



(※図：SEO 対策がわかる， 株式会社アイレップ SEM 総合研究所)

クローラによるページへのアクセスはページをインデックスに追加する際の一回きりではない。一度アクセスしてインデックスを作成したページでも定期的にアクセスし直すことで最新情報への更新を行っている。

Google がデータを更新する頻度には2つのパターンがある。1つ目は月に1、2度のパターン、2つ目はほぼ毎日更新するパターンである。Google がインデックスしているページ数は80億以上にのぼる。それら全てをチェックしてインデックスを常に最新の状態に保つのは非常に困難な作業である。そこでGoogle ではインデックスしたページを「頻繁に更新の行われるページ」と「それ以外のページ」の2種類に分類している。頻繁に更新の行われるページは「頻繁にデータを更新すべき」と判断し、ほぼ毎日インデックスの更新を行う。そしてそれ以外のページについては月に1、2度インデックスの更新を行う。

インデックス情報と実際のページ情報の乖離

Google では定期的なインデックスの更新を行っているが、ページが更新されるタイミングとインデックスの更新は必ずしも一致しない。サイト更新のタイミングはページ作成者によってまちまちであるため、Google がそのタイミングを把握することは難しい。そのためページが更新されても、しばらくの間はGoogle のインデックス情報が更新されないという状態がしばしば発生する。検索にヒットしたページに検索キーワードと合致する情報が存在しないといった事態は、こうした実際のページ情報とインデックス情報の乖離が原因である。

SEO 対策の対象要素

検索エンジンはHTML を読み込むことで対象のWeb ページを評価し、検索順位を決定する。検索エンジンは人間と同じように視覚的表現を含めたかたちでWeb ページを認識することができない。そのため評価の対象はテキストコンテンツとなる。そして検索エンジンは論文の様に論理構造が整ったテキスト情報を価値が高いと判断し、高評価を与える。テキスト情報が題、部、章、節、項、小見出しと整理されており秩序立った構造を持つページの方が、ユーザにとっても検索エンジンにとっても情報の把握が容易であり、情報の価値が高いと考えるためである。そして検索エンジンのWeb ページ評価を考える上で忘れてならないのがリンクの存在である。リンクにはサイト内部に張り巡らされた内部リンクと、サイトの外部につながる外部リンクが存在する。特に外部リンクではGoogle のPageRank アルゴリズムにも通じる学術論文の引用を参考とする、Web ページのポピュラリティ(評判)を基準とした評価が強く見受けられる。このアルゴリズムには近年頻繁に更新が加えられており、検索順位の評価における重要度が増していると言えるだろう。

以上のことをふまえると、SEO対策は検索エンジンの評価対象要素である①テキスト要素、②リンク要素、③ポピュラリティ要素を対象とした三つに分類できる。①テキスト要素、②リンク要素がページの内部的要素、③ポピュラリティ要素がページの外部的要素にあたる。以下で各要素について説明する。

テキスト要素

Web サイトは通常、HTML によって記述されている。クローラがHTML を読み込む際、特に注目するのが<title>タグ、<meta>タグ、<h>タグ、<body>タグ内のテキストである。これらのテキストを対象に下される評価が検索順位決定のテキスト要素にある。これらのテキストを対象として施行されるSEO対策がテキスト要素に類する。

リンク要素

クローラはHTML を読み込むことによってページのインデックスを作成する。従ってHTML はクローラが読みやすい論理的な構造を持ち、かつ軽量であるのが望ましい。

一方、Web ページを閲覧するユーザに向けてはデザインが簡素過ぎないように配慮する必要がある。テキストコンテンツのみではなく、ページ全体のレイアウトも使用性の高いものにするのが重要である。サイトはクローラ、ユーザ両方にとって見やすく巡回のしやすい内部リンク構造を取る必要がある。このページ内構造を対象に下される評価が検索順位決定のリンク要素である。このページ内構造を対象として施行される SEO 対策がリンク要素に類する。

ポピュラリティ要素

ポピュラリティ要素とは主に外部リンクに対する価値評価である。最も重要視され、アルゴリズムの更新に晒されることが多いのがこのポピュラリティ要素である。どんなページからリンクされているか、どれだけのページからリンクされているかが自サイトのポピュラリティ要素の評価基準となる。ポピュラリティ要素は他サイトから自サイトへのリンクを基準としているので評価の客観性が高い。他サイトからのリンクはいわば自ページに対する「指示投票」である。リンク一票あたりが評価に影響する大きさは一定ではなく、リンク元サイトの評価の高さ、自サイトとリンク元サイトのコンテンツの関連性等により評価も上下する。ポピュラリティ評価を分かりやすく確認できる指標として PageRank という概念がある。

PageRank とは

Google のポピュラリティ評価の目に見える指標の一つが PageRank である。Google は PageRank について自身のページ、『Google の人気の秘密』で以下のように語っている。

PageRank™ は、Web の膨大なリンク構造を用いて、その特性を生かします。ページAからページBへのリンクをページAによるページBへの支持投票とみなし、Googleはこの投票数によりそのページの重要性を判断します。しかしGoogleは単に票数、つまりリンク数を見るだけではなく、票を投じたページについても分析します。「重要度」の高いページによって投じられた票はより高く評価されて、それを受け取ったページを「重要なもの」にしていくのです。

PageRank は 0～10 の十段階で表され、ランクが高くなるほど評価も高い。Google ツールバー等で直接確認できる指標のため外部リンクの価値判断に利用できる。

以上の様に、外部から自ページへの価値評価を対象に下される評価が検索順位決定のポピュラリティ要素である。この外部から自ページへの価値評価を対象として施行される SEO 対策がポピュラリティ要素に類する。

ブログを利用した SEO 対策の実証実験

ブログの SEO 的優位性

利用者の増加と共にブログサイトが検索上位に顔を覗かせることも多くなった。多くの専門書ではブログが SEO 対策に適していることに言及している。それはブログサイトが従来型サイトと比較して主に以下の点で優れているからである。

内部リンクが自動で構築される

ブログは新規にページを作成する際、新しいページに対して自動で内部リンクが構築される。従来型サイトでは内部リンクの構築はサイト管理者の仕事であり、ページが増加するに従ってその労力は増していく。従来型サイトがリンク切れ等の手違いが起りやすいのに対して、ブログは最初に設定さえしてしまえば全て自動で内部リンクが構築されるので、手違い無く隅々までリンクの張り巡らされた論理的な構造のサイトを構築することができる。

外部リンクが獲得しやすい

ブログにはトラックバックという相互リンク生成機能が標準で装備されており、従来型サイトと比較して容易に外部リンクを獲得できる。近年ではトラックバックに nofollow 属性が付加されているブログサービスが多く直接のポピュラリティには繋がらないが、大きな宣伝効果を有していることに変わりはない。トラックバックのリンク対象はページ毎なので、他サイトから関連性の高いユーザをアクセス誘導できる。同様にコメント機能も自ブログの知名度向上に効果を発揮する。

論理構造とデザインが分離している

ブログはテキスト等の文書の構造を表す「論理構造 (HTML)」とテンプレートといったサイトのデザインを表す「視覚構造 (CSS)」が基本的に分離している。これは W3C の提唱する Web 標準化されたサイトの構造である。ブログはサイトのデザインを CSS で管理することにより、HTML にはクローラから重要視されるテキスト情報のみを記述することができる。

ここからは近年利用の拡大が目覚ましく、SEO 対策に適したブログサイトを利用して SEO 対策の効果実証の実験を行う。そして有効な SEO 戦略を模索するものである。

ブログを利用した SEO 対策の効果実証実験

実験内容

実験用のブログサイトを作成し、各種 SEO 対策と Google 検索順位の相関関係を調査する。期間は三ヶ月（92 日）、調査対象のキーワードは三つ。毎日基本的に 23 時過ぎに各キーワードについてブログトップページの検索順位を測定し、施行した SEO 対策と共に記録を取る。検索順位は 500 位まで調査を行い、それ以下の順位は調査圏外扱いとする。検索順位の調査には特定キーワードの検索順位を 500 位まで調べることができるフリーツールを利用する。なお、ブログサイトの作成には無料のブログサービスを利用する。

サイトの開設

一口に無料ブログサービスと言っても、その数は百以上存在する。今回の実験には FC2 ブログを利用することにした。FC2 ブログは以前使用していたこともあり操作を熟知しているという理由もあるが、SEO 対策的に見ても以下に示す重要な理由が存在する。

URL がサブドメイン形式である

無料ブログサービスで利用可能な URL の形式には以下の 2 種類がある。

- ・サブドメイン形式 URL : `http://ユーザ ID. XXXX. jp/`
- ・ディレクトリ形式 URL : `http://www. XXXX. jp/ユーザ ID/`

サブドメイン形式 URL とディレクトリ形式 URL の相違点は、URL に含まれる各ブログサービスユーザ ID の配置にある。サブドメイン形式 URL がスキーム名の直後にユーザ ID を配置するのに対し、ディレクトリ形式 URL では URL の最下層であるパス名の部分にユーザ ID が配置される。

Google では同一ドメインを持つサイトは二つまでしか検索結果に表示されない。つまり検索で同一ドメインとみなされるディレクトリ形式 URL を持つブログの場合、検索にヒットした同一ドメインをブログの上位 2 位以内に入っていなければ検索結果に表示されない。一方サブドメイン形式 URL を持つブログの場合は検索にヒットするもの全てが表示される。検索エンジンからのアクセス誘導をという SEO 対策の効果を考えて、サブドメイン形式の URL が有利となることは明白である。

加えて、検索結果に表示されないのでは実験の効果測定のしようがない。以上のような理由からサブドメイン形式 URL を持つブログを選択した。

サーバの安定性

ブログを選択する上で重要な要素に、サーバの安定性がある。サーバが安定的であれば閲覧者も管理者もページを素早く表示することが可能であり、閲覧・編集の煩雑さを減らせる。FC2 ブログは記事やプラグインの更新で逐一ブログ全体を再構築する必要がなく、サーバに掛かる負担が少ない。更新に割く時間も少なく済み、更新途中で通信がとぎれてしまう危険が低く安定した運用が可能である。

テンプレートの拡張性

SEO 対策を施すためには HTML の編集は必須である。しかし中にはテンプレートの編集を禁じているブログサービスもある。FC2 ブログは広告タグ 1 行を残せばテンプレートの編集は自由であり、拡張性が非常に高い部類に入る。

検索キーワードの選定

ブログを開設したところでまず始めに取り組まなければならないのが何というキーワードで検索上位を目指すかの決定、SEO 対策の対象となる「検索キーワード」の選定作業である。

今回は SEO 対策の効果を分かりやすく測定するために、SEO 対策の効果が比較的現れ易いと思われる検索ヒット数の少ないスモールキーワードを主軸に、他サイトから影響を受けやすく検索ヒット数の多いビッグキーワード、そしてその中間のヒット数のミドルキーワードの 3 つを対象キーワードに設定する。

次にキーワードの具体的な内容である。キーワードはページのテーマに合致するものでなければならない。実験の効果測定はブログのトップページを対象として行う。今回トップページに表示される記事数は 5 件に設定した。そのためトップページに表示される記事 5 件の内容全てが検索順位決定に反映されることとなる。さらにトップページの内容は記事を増やすとともに古い記事が新しい記事へと入れ替わっていく。つまり今回 SEO 対策の主軸とするスモールキーワードは特定記事ページのテーマを表す内容ではなく、ブログ全体を表すテーマと言ってよい。

そこでテーマとして選択したのが「ページランク対策」というキーワードである。もともと SEO 対策に関連する話題をまとめたブログの作成を考えていた。そのことから連想したキーワードである。

このキーワードは「ページランク」というキーワードと「対策」というキーワードの複合キーワードである（厳密に言うと Google は「ページ」と「ランク」を別々のキーワードとして認識するので、3 語からなる複合キーワードである）。「ページランク（検索ヒット数約 3,130,000 件）」と「対策（約 3,130,000 件）」という二つのビッグキーワードを組み合わせ、一つのニッチなスモールキーワードを作成したのである。

このスモールキーワードを基準として以下の三つのキーワードを選定した。

スモールキーワード：ページランク対策（検索ヒット数約 10,700 件）

ビッグキーワード：ページランク（検索ヒット数約 2,760,000 件）

ミドルキーワード：グーグルページランク（検索ヒット数約 419,000 件）

「ページランク」というワードを全てのキーワードに含めている。全てのキーワードに「ページランク」の語を含めることでブログのテーマをぶれにくくし、三つの全てに SEO 対策の効果が現れるように設定している。

SEO 対策の実行

ブログのテーマとキーワードの選定が終了したところで具体的な SEO 対策に取りかかった。ここからは既に論じた論じた SEO 対策の三類型、①テキスト要素、②リンク要素、③ポピュラリティ要素に沿って施行した SEO 対策を説明する。

テキスト要素

テキスト要素に対して実行した SEO 対策について説明する。ブログのテンプレートはゼロから作成するのではなく、フリーで配布されているものを改良して利用した。

<title>タグの最適化

<title>タグ内には

```
<title>ページランク対策.com <%sub_title></title>
```

と記述した。テキストの先頭に必ず「ページランク対策.com」というブログタイトルが表示されるように設定している。ブログタイトルの後ろに付いている<%sub_title>のタグは FC2 の独自タグで、閲覧者の行動に応じてエントリータイトル、日付、検索ワード等を表示する。例えば閲覧者が特定の記事ページを閲覧している場合は「ページランク対策.com エントリータイトル」のように表示される。<title>タグは文字通りページのタイトルを表すタグである。検索エンジンはそのページがどんな内容を扱っているのか<title>タグ内のテキストによって判断する。つまり<title>タグ内テキストでそのページについての適切な記述を行えば、検索エンジンがページを理解する上での助けとなり、高評価を獲得することができる。<title>タグ内のテキストは検索結果ページでのアンカーテキストとしても使用されるため、ユーザがページ内容を判断する基準にもなる。ユーザの興味を引くテキストを記述することでアクセスの増加にもつながるのである。

<title>タグを記述する際に注意したいことは、重要なキーワードほど<title>タグ内の前方に記述するよう意識することである。検索エンジンは<title>タグに限らずテキストの前方にあるワードほど重要であると考えられる。

<meta>タグの最適化

各<meta>タグに最適化を施した。<meta>タグはHTMLの<head>タグ内に記述され、いくつかの種類がある。ここではSEO対策と関係の深いもののみを抜き出し説明する。

<meta charset>タグの最適化

<meta charset>タグとはページの文字コードを伝えるタグである。

```
<meta http-equiv="content-type" content="text/html; charset=euc-jp" />
```

上記の様にEUC-JPの日本語文字コードを指定した。

検索エンジンが正しくページを表示できるのは、このタグから適切な文字コードを判別しているからである。このタグが正しく記述されていないと文字化け等のトラブルが発生する。

<meta description>タグの最適化

<meta description>タグは、そのページの説明文を伝えるタグである。

```
<meta name="description" content="〇〇〇〇" />
```

〇〇〇〇の部分に具体的な説明テキストを記述する。今回はここに「ページランク対策に関するレポート、その他ニュース、いろいろ。」とキーワードを盛り込んだ説明文を記述した。テキストの先頭にターゲットキーワードの「ページランク対策」を配置して検索エンジンの評価を得やすくしている。

Googleではここに記述した説明文が検索結果に反映され、ユーザの目に直接触れることになる。タグ内にキーワードを盛り込むことで検索エンジンにページのテーマを伝えることができるほか、ユーザにもページに興味を持って貰うことができる。

<meta keyword>タグの最適化

<meta keyword>タグとはページにとって重要なキーワードを記述するためタグである。

```
<meta name="keywords" content="〇〇〇〇, 〇〇〇〇, …, 〇〇〇〇" />
```

〇〇〇〇の部分にキーワードとなるテキストを記述し、キーワード一つひとつを半角「,」で区切る。今回はここに「ページランク, 対策, グーグル」とターゲットキーワードを構成するワードを重要度順に記述した。

<meta keyword>タグ内では前方には位置されたキーワードほど重要であると検索エンジンに判断され、複数のキーワードを記述することが可能である。しかしキーワード数は10個以内に収めるのが無難だろう。何十何百とキーワードを列挙するとスパム行為と認識される恐れがあるためである。

フリーテンプレートの<meta>タグは初期設定ではテンプレート作者が自由に設定している場合がほとんどである。そのためユーザ側で編集を行い、SEO 対策を行うことで他のブログとの差別化を図ることができる。

<hx>タグの最適化

<hx>タグにブログ内で重要度の高いキーワードを含む部分を指定した。実際に何を指定したものを以下に示す。

<h1>タグ：ブログタイトル

<h2>タグ：ブログ説明文、トップページに表示されている記事のカテゴリ

<h3>タグ：トップページに表示されている記事のタイトル

初期段階では<h3>タグに記事カテゴリ、<h4>タグに記事タイトルを指定していた。しかし途中からこれらに指定する<hx>タグを一つずつ繰り上げている。<h4>、<h5>、<h6>タグは重要度が低く、さらに使用することでHTML 構造を崩してしまうと判断したため最終的には排除したかたちになっている。

<hx>タグはHTML テキストの論理構成を示す見出しに使用されるタグである。<hx>タグのx には1~6 の数字が入り、<h1>から順それぞれのタグが題、部、章、節、項、小見出しの役割を持つ。テキストはその役割に応じて文字大も変化して表示される。<h1>タグで指定されたテキストが最も大きく表示され、後は順に小さくなっていく。

<hx>タグの使用で重要なのは<h1>~<h6>のタグにキーワードを重要度の高い順に配置すること、HTML テキスト上で<h1>~<h6>のタグを順番通り配置することの二つである。

まず前者だが、検索エンジンは各<hx>タグで指定されたテキストをh1>h2>h3>h4>h5>h6 の順で重要度が高いと判断する。この評価は各タグの役割を考えれば当然と言える。<title>タグと同じく一ページにつき一つしか使用のできない<h1>タグにはページのテーマとなるようなテキストを指定するのが望ましい。後は<h2>から順に重要度の高いキーワードを指定していくこととなる。

後者は先に述べた検索エンジンが論文のようなHTML 構造を嗜好することと関係がある。<hx>タグがバラバラの順番で配置されている場合は、検索エンジンは題、部、章、節、項、小見出しというページの論理構造を正しく認識することができないためである。

今回利用したものもそうであったがフリーテンプレートの場合、初期設定で<hx>タグが記事の日付、コメント、トラックバックやサイドメニュー等見出しとして強調する必要性のない項目に設定されていたり、タグが順番通りに並んでいないものがほとんどだった。そのため<hx>タグへのSEO 対策により他のブログとの差別化に一定の効果が期待できる。

キーワードの強調

ブログ記事内のテキストにおいて、「ページランク」「対策」「グーグル」といったターゲットキーワードに含まれるワードをタグを用いて強調した。

テキスト内で検索エンジンにアピールしたいワードを個別に強調する際に効果を発揮するのがタグやタグである。タグで囲まれたテキストは太字、タグで囲まれたテキストは斜体でブラウザ上に表示される。検索エンジンはこれらのタグで囲まれたワードをページ管理者がテキスト内で特に強調したい部分と認識する。これらのタグは「論理的強調タグ」と呼ばれる。同様にテキストの見た目を変更するタグでタグや<i>タグがある。しかしこれらのタグはテキストの見た目を変えるのみで、検索エンジンに強調したい部分を伝える効果は持たない「視覚的強調タグ」である。

キーワード出現率の操作

ページ内に含まれる「ページランク」と「対策」という2つのキーワードの出現率を5%程度に維持することを意識してキーワードを多く含ませるようブログ記事を記述した。しかし、実際の出現率は5%よりもかなり低い水準にとどまった。原因はブログの性質にある。ブログはその性質上、記事を増やすにしたがってトップページの記事が入れ替わっていく。初期の記事は「ページランク」「対策」というキーワードに言及したものが多く、出現率も10%以上の高水準を維持していた。しかし記事が増えるにつれ特定のキーワードをのみに言及した記事ばかりを量産するのは困難となる。結果、中期以降トップページのキーワード出現率は5%未満の低水準に下落してしまった。

キーワード出現率とは管理者が検索エンジンに対して強調したいターゲットキーワードが、ページ内のテキストにどの程度含まれているかという割合である。キーワード含有率は単純に高ければよいというわけではない。キーワード含有率を100%にしたところで、検索エンジンからは恐らく確実にスパム行為とみなされてしまうだろう。一般的にはページあたり5%程度のキーワード含有率を維持するのが望ましいと言われている。

ターゲットキーワードをテキスト前方に配置する文章の組み立て

ターゲットキーワードをできる限りテキストの前方に組み込むように文章の組み立て、言い回し等を工夫した。これは<title>タグの際にも述べたが、検索エンジンがテキストの前方にあるワードほど重要であると考えためである。

HTMLの軽量化

配布されているテンプレートには不要な装飾、広告等のタグが含まれている場合がある。それら不要と思われる部分は全て削除し、HTMLの軽量化を行った。クローラの巡回効率を考えるとHTMLは少しでも軽微なものが望ましい。ちなみにGoogleは以前、100KBを超えるHTMLを持つページはインデックスされなかった。そのためSEO対策を考えるとHTMLは100KB

以下に抑えるべきである。だが実際に HTML のファイルサイズが 100KB を超えることは稀である。

リンク要素

リンク要素に対して実行した SEO 対策を説明する。

内部リンク構造の最適化

カテゴリ分けを 2 層構造にすることで記事を細かく分類し、リンク構造の整理を行った。細かくカテゴリ分けすることでよりページの内容に即したカテゴリに各ページを分類することができる。これは検索エンジンやユーザのページ内容の理解を容易にする効果がある。

記事の増加と共に自動で内部リンクを構築してくれるのはブログの長所の一つだが、記事が多数になると細かなカテゴリ分けが必要となってくる。Google は一つのページに 100 以上の無秩序なリンクが存在している場合、そのページをリンクファームというスパムページとみなす。リンクファームとみなされたサイトはページランクや検索順位の低下、検索結果からの除外といったペナルティを受ける。以下にリンクファームの簡単な定義を示す。

リンクファーム (英: Link farm) とは、ハイパーリンクによってページを相互にかつ大量にリンクしているウェブサイト群を指す。人が手で構築することもあるが、通常はプログラムによって自動的に生成される。リンクファームは検索エンジンスパムの一種である。意味のあるリンクの仕方をしているサイトは、リンクファームとは呼ばれない。

出典: フリー百科事典『ウィキペディア (Wikipedia) 』

最後の一文曰く、「意味のあるリンクの仕方をしているサイトは、リンクファームとは呼ばれない」のである。つまり様々なページを一つのカテゴリに格納するのではなく、ページ毎に内容に即したカテゴリに細かく分類すればスパムページとはみなされない。検索エンジンやユーザにページのテーマも伝わりやすくなるので一石二鳥である。

ほとんどのブログサービスは記事カテゴリの機能を備えているので、リンクの分類が非常に容易である。にもかかわらず多くの記事を抱えるブログであっても、記事のカテゴリ分けは大雑把であることが多い。他のブログよりも細かなカテゴリ分けをすることで SEO 対策的には差別化の効果がある。

ナビゲーションリンクの追加

使用性向上のためにタイトルバナー下にナビゲーションリンクを追加した。トップページから特定カテゴリを選択し、カテゴリページから特定記事を選択した場合以下のような表示が出る。

トップページ >>カテゴリ >>記事タイトル

この表示を加えることでユーザは自分の閲覧しているページがブログの内部リンク構造のどこに位置しているのか容易に把握することができる。

アンカーテキストにキーワードを使用

内部リンクのアンカーテキストには可能な限りターゲットキーワードを使用した。ブログ全体が「ページランク対策」という一つのテーマに基づいて作成されているため、アンカーテキストにも自然にキーワードを取り入れることができた。

アンカーテキストとはHTMLの中で、リンクが設定された文字列のことである。検索エンジンはリンク先ページの内容を判断する材料としてアンカーテキストを利用している。つまりアンカーテキストにターゲットキーワードを盛り込むことで、リンク先ページのテーマとしてターゲットキーワードを検索エンジンに伝えることができるのである。

ちなみに検索エンジンはアンカーテキスト前後のテキストもリンク先ページの判断材料にしている。テキスト中で何の脈絡もなく不自然なアンカーテキストを記述しても効果は期待できない。

記事内のキーワードから参照できる用語集を作成

アンカーテキストによるページのテーマ付けをより強力なものにするため、用語集のコンテンツを作成した。記事内のSEO対策用語を別記事で詳しく解説し、そのページへのリンクを張る。通常記事では用語に関する説明を省き、ブログ全体のターゲットキーワードである「ページランク対策」に専念することができるので、記事内のキーワード含有率を下げる心配もない。用語集のページでは、その用語を扱ったページからのリンクを一手に集めることができるので、その用語に関する強力なテーマ付けが可能となる。

サイトマップの登録

クローラの巡回効率を上げるためにサイトマップをGoogleウェブマスターツールに登録した。サイトマップを登録することでサイトの隅々までクローラの把握が可能となる。ページのインデックス漏れといった危険を防ぐことができる。

FC2ブログの場合、サイトマップは自動で作成される。ユーザは作成されたサイトマップを登録するだけでよい。

各ブラウザでの表示の調整

Internet Explorer、Firefox、Safari等の異なるブラウザでブログを閲覧した際に表示に崩れがないように調整した。普段自分が利用しているブラウザで問題なく閲覧できていても、他の閲覧者が他のブラウザから閲覧したときに表示が崩れてしまったのでは意味がない。

表示が崩れるだけならまだ良いが、使用性にも影響が出る恐れがある。アクセス解析からも利用者の多かった Internet Explorer、Firefox、Safari でブログを閲覧しても問題の無いように対策を施した。

ポピュラリティ要素

ポピュラリティ要素に対して実行した SEO 対策を説明する。外部リンク獲得のため、ブログの宣伝活動を中心とした SEO 対策を行った。

多種検索エンジンへの登録

先に述べた三大検索エンジン以外の日本国内外二〇あまりの検索エンジンに登録し、閲覧者の増加、外部リンクの獲得をねらった。閲覧者が増えれば、それだけリンク等を獲得するチャンスも増す。日本の検索エンジンシェアの大半は Google、Yahoo、MSN の 3 強が占めていることは前述したとおりだが、無数に存在する他の検索エンジンも無視できる存在ではない。

新着記事の PR

ping 送信サービスを利用して FC2 以外の ping サーバ・ping サイトにも ping を打ち、ブログの宣伝を行った。ブログでは通常、新しく書いた記事は各ブログサービスの新着記事欄に記事が追加される。しかしそれだけでは新着記事はそのブログサービスの利用者に伝わるのみで、主な閲覧者はそのブログサービスの利用者のみに限られてしまうこととなる。

ブログの更新を広く伝える手段として「ping」というものがある。インターネットには様々なブログサービスの更新情報を収集し、新着記事を一覧できるようにしているサイトが存在する。そういったサイトは「ping サーバ」あるいは「ping サイト」と呼ばれる。この ping サーバに対して更新情報を送信することを ping を打つと言う。ping を打っておけば、ping サイト利用者全てに更新情報を閲覧できるため、記事を広く宣伝することができる。

トラックバックの送信

新しい記事を 1 件書く度に他のブログあるいはトラックバックの送信できるニュースサイトへのトラックバックを送信した。記事一つにつき、その記事と関連のあるキーワードで Google 検索した上位 3 つのサイトにトラックバックを送信した。

ディレクトリ登録

ディレクトリ型サイトへの登録をすることで、直接的な外部リンクの獲得を試みた。ディレクトリは登録サイトを細かくカテゴリ分けして管理しているので、登録されることで自ブログと同一のテーマを扱ったページからのリンクを獲得することができる。

ロボット型検索エンジンが主流となった現在でも、未だディレクトリが検索結果に及ぼす影響は大きい。ディレクトリに登録されるサイトはカテゴリごとに分類されている場合が多い。つまりそういったディレクトリサイトに参加するということは、自分と同様のテーマを扱うサイトからのリンク獲得と同義なのである。加えてディレクトリはサイト同士の交流の場としての機能も併せ持っている。特定テーマのサイトコミュニティに属することで自サイトの知名度向上にもつなげることが可能である。

ランキングサイト登録

SEO 対策・Web・コンピュータ関係のランキングサイトに登録した。特定テーマのランキングサイトに登録することは、ディレクトリと同様の効果が得られる。

コミュニティサイト登録

SEO 対策・Web・コンピュータ関係のコミュニティサイトに登録した。同じ話題を扱うサイト管理者内での知名度を向上させ、外部リンクの獲得を狙う。

アクセス交換サービスの利用

サービスに参加する他ブログを閲覧することで、自ブログも閲覧してもらえるアクセス交換サービスを利用して自ブログの知名度を高めた。

ニュース記事への言及

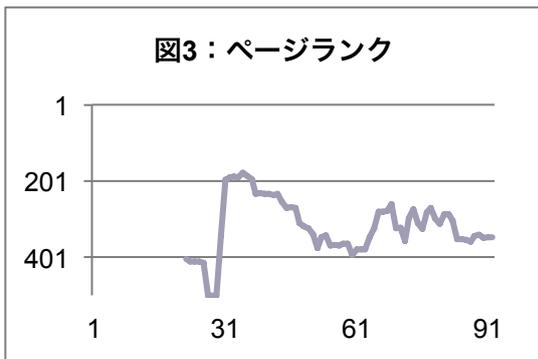
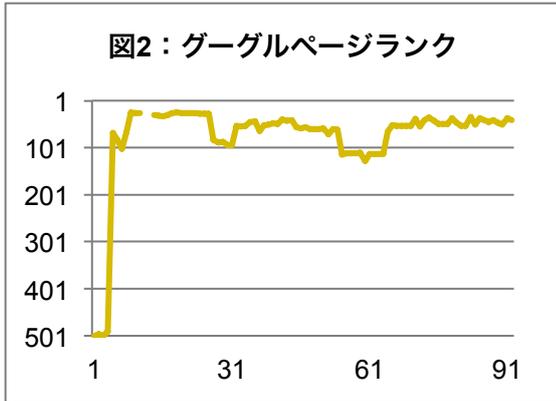
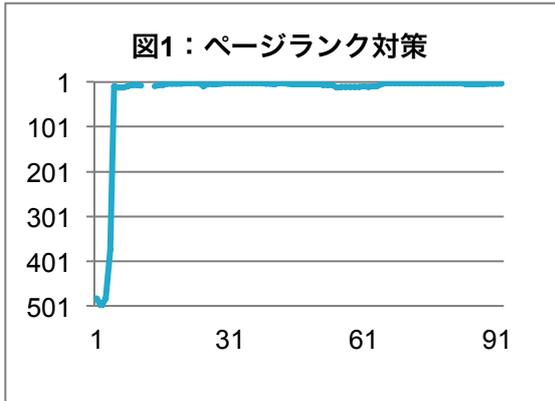
Google ニュースで「SEO」と記事検索し、関連の深いニュースを記事として取り上げた。可能な所にはトラックバックを送信し、知名度の向上を図った。

結果と考察

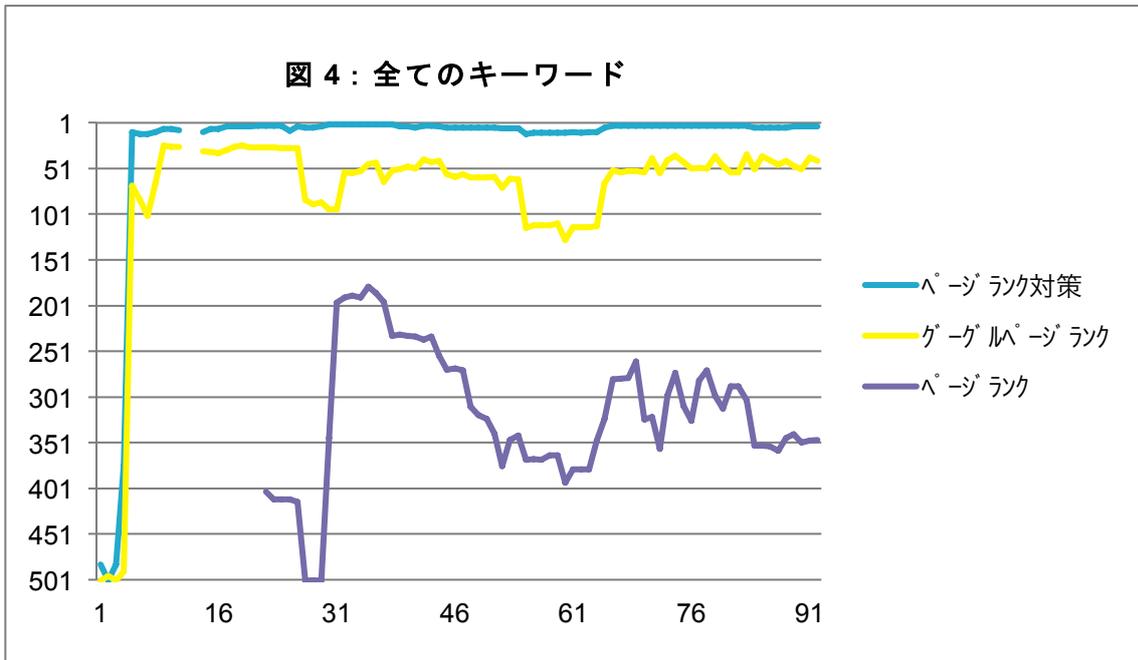
結果

各ターゲットキーワードの検索順位の推移を以下にグラフで示す。縦軸が検索順位、横軸が実験開始からの経過日数を表している。

※使用している検索順位チェックツールがメンテナンスに入ってしまったため「ページランク対策」、「グーグルページランク」の12、13日目の順位データはない



【各キーワード最高順位】
 ページランク対策 : 3 位
 グーグルページランク : 26 位
 ページランク : 180 位
 (※キーワード「ページランク」は22日目から調査を始めたため初期の順位データはない)



考察

図4のグラフから分かる通り、各キーワードの検索ヒット数に応じて検索順位に明確な差が出た。「ページランク対策」、「グーグルページランク対策」の両キーワードの順位が実験開始5日目で早くも急上昇していることが分かる。「ページランク」は11位/10,700件、「グーグルページランク」は70位/419,000件とどちらもかなりの高順位を獲得している。

この順位の急上昇は、ブログに対するGoogleの認知が完全なものとなったことに起因するだろう。実験開始3日目にGoogleのクローラがブログに来訪していることから、インデックスが更新され、更新が検索順位に反映された5日目に順位が急上昇したと思われる。

続く二度目の順位上昇は9日目に観測された。「グーグルページランク」の検索順位が9日から10日目にかけて70位近く順位を上げている。分かり辛い「ページランク対策」もわずかに順位を上げている。この上昇の理由はブログのテキスト要素に対するSEO対策の成果と判断できる。〈title〉タグ、〈meta〉タグ、〈h〉タグといったサイト内部のテキスト要素に対するSEO対策を完成させた後にこの二度目の順位上昇が現れているからである。この後も小さな手直しは入れているが、ブログの原型はこの時点で既に完成している。

だがこの検索順位も27日目付近で急激に低下する。「ページランク対策」の順位が6位低下、「グーグルページランク」の順位が56位低下、「ページランク」の順位に至っては416位から調査圏外まで低下している。この順位の低下は23日目に行った強調テキストをワードごとに切り離す試みが原因だと考えられる。この試み以前まではテキスト中で“ページランク”というキーワードを一語とみなして、

〈strong〉ページランク〈/strong〉

とキーワードの強調を行っていた。しかしGoogleは“ページ”と“ランク”を別々のワードとして認識する。そのため“ページ”と“ランク”を切り離して個別に強調した方が強調する頻度が増し、キーワード出現率を増やすことができると考えた。この試みでは“ページランク”というキーワードを以下の様に強調する。

〈strong〉ページ〈/strong〉〈strong〉ランク〈/strong〉

今までに作成した全ての記事の強調形式をこの様に書き換えたところ、全くの逆効果で全てのキーワードの順位が低下してしまった。この試みが順位低下の原因である根拠として、実験27日目に強調テキストを以前と同様に戻したところ順位が回復したという事実がある。

しかし回復後も「グーグルページランク」と「ページランク」の両キーワードは60日目を谷底とした順位の低下に向かう。両者の違いは「グーグルページランク」が55日目に急激に順位を下げているのに対して、「ページランク」が徐々に順位を下げている点だ。この間実行したSEO対策を見渡す限り、特別問題になるような対策は見当たらない。「グーグルページランク」が急激に順位を下げた55日目には関連した用語集の統合を行っているが、

その後用語集に改善を加えずとも 64 日目には順位を回復していることから用語集が無関係であることが分かる。他には近い時期に、キーワードの出現率を高めることを目的として過去に作成した出現率の高い記事をトップページに持ってきている。しかしこれらトップページに持ってきた古い記事が新しく作成した記事に取って代わられる前に、順位は回復している。そのためこちらも直接的な因果関係は無いだろう。

「グーグルページランク」と「ページランク」の両キーワードは比較的検索ヒット数の多いビッグキーワードである。つまり競合他サイトからの影響という線が有力だと考えられる。

ここでポピュラリティ要素について言及する。実際のところ、ポピュラリティ要素に対する SEO 対策は手段は尽くしたものの、実験期間中に獲得できた外部リンクはわずか 18 だった。それも外部リンクを獲得したからといって速やかな検索順位の向上は観測できなかった。

SEO 対策において最も重要なのは外部リンク、すなわちポピュラリティ要素であるというのが現在の検索エンジン業界で大勢を占めている理論である。以前重要視されていたテキスト要素への SEO 対策は今日ではあまり効果が期待できないというのが通説である。しかし今回の実験を通して、私はニッチなスモールキーワードではページの内部的要因、特にテキスト要素こそが重要な役割を持っていると感じた。

<title>タグ、<meta>タグ、<hx>タグ、タグやアンカーテキストを利用して一貫した一つのキーワードをページテーマとして検索エンジンに伝えることで、強力なポピュラリティを持つサイトが少ないスモールキーワードでは充分優位に立つことができる。

外部リンクはある程度自サイトの知名度が上がらなければ獲得が困難である。高いポピュラリティをもたらすサイトは、相手も高いポピュラリティをもたらすようなサイトでなければ安易にリンクを張ろうとはしない。つまり自サイトがターゲットしているキーワードにおいて、ある程度の順位になれば外部リンクの獲得は期待できないのである。ポピュラリティ要素といった外部的要因に対しての SEO 対策ではより長期的な視野を持つことが重要である。

そこで有効な SEO 戦略を考えると、初期段階は内部的要因であるテキスト要素・リンク要素への対策でサイトのテーマ・構造を固めスモールキーワード業界で順位を高める。スモールキーワード業界である程度の順位を獲得し、知名度を高めることができれば、スモールキーワードと関連のあるビッグキーワードに対してリンク獲得に挑むという SEO 戦略をとるのが有効であると考えられる。

【参考・引用資料一覧】

- ・ SEO「検索エンジン最適化」の教科書. 吉村 正春
- ・ 図解 SEO 対策がわかる. 株式会社アイレップ SEM 総合研究所
- ・ SEO 検索エンジン最適化プロジェクト. 住 太陽
- ・ [SEO] ロボット型検索エンジン最適化. 住 太陽
- ・ Web 検索エンジン Google の謎. 水野 貴明
- ・ SEO と検索キーワード連動型広告. 高幡 泰

- アクセスアップのための [SEO] ロボット型検索エンジン最適化. 住 太陽・アングラーズネット
- 平成 17 年「通信利用動向調査」. 総務省
http://www.soumu.go.jp/s-news/2006/060519_1.html
- ブログ及び SNS の登録者数 (平成 18 年 3 月末現在). 総務省
http://www.soumu.go.jp/s-news/2006/060413_2.html
- The State of the Live Web, April 2007. Technorati
<http://www.sifry.com/alerts/archives/000493.html>
- 第 23 回インターネット利用者アンケート結果. Yahoo! JAPAN
<http://docs.yahoo.co.jp/info/research/wua/200804/page14.html>
- 2007 年 5 月度月間インターネット利用動向調査. ネットレイティングス
http://csp.netratings.co.jp/nnr/PDF/Newsrelease06252007_J.pdf
- 検索エンジン利用状況実態調査. オプト、クロス・マーケティング
http://www.opt.ne.jp/upload/000000238_0K1VJUKNJo.pdf
- Google、Yahoo! 検索結果画面での視点データを収集したアイ・トラッキング調査. アイレップ SEM 総合研究所、JMI
<http://www.irep.co.jp/press/pdf/20070611.pdf>
- 検索結果は上から順に。上位表示サイトのクリック率高まる、87%。インターネットコム、インフォプラント
<http://japan.internet.com/research/20020320/1.html>
- 検索エンジン・ポータルサイト 日本でのシェア (使用率)
<http://www.webcreate.ga-pro.com/search.html>
- Google の人気の秘密
http://www.google.co.jp/intl/ja/why_use.html
- Wikipedia
<http://ja.wikipedia.org/wiki/?keyword=V&>
- IT 用語辞典 e-Words
<http://e-words.jp/>

さくいん

お

小澤 慎規, 9

か

神谷 僚, 17

く

黒崎 将史, 31

こ

児玉 育也, 83

児玉ゼミナル, 7

さ

サーブレット, 27

坂田 匠, 41

し

Java, 10, 13, 21

た

田村 俊一, 93

て

データベース, 14

デザインパターン, 15

に

ニコニコ動画, 64

ま

松崎 有樹, 59

松原 正和, 65

ろ

ロングテール現象, 18