

9. 小規模 Web サービスの RAD による評価

9.1. 序言

インターネットインフラの整備に伴い、ネットワークアミューズメント市場は拡大の基調を示している。特に DSL に代表される常時接続環境の充実、さらには光ケーブルを用いた高速かつ安定な常時接続が極めて安価に提供されつつあり、これがネットワーク上でのデータ授受を極めて容易なものとしている。これに伴い、今まで不可能であった大容量データ送受信を伴うネットワークアミューズメントの市場が生成されつつある。この市場規模は全ゲーム市場の数十パーセントになるといわれ、今後無視できない規模になりつつある。しかし、一方でサーバへの投資など、規模の増大に比例する投資は増加の一步をたどっている。数十万ユーザを同時に接続するためには、サーバを分散構築することがコスト削減の最も大きな部分となると思われる。

ここで OpenSOAP はオープンソースの環境であるために、小規模のパイロットケースとしてネットワークシステムを構築するにはリスクの少ないシステムと考えられる。SOAP ではサーバ間の連結が疎結合であるため、多数のユーザの接続に対して将来的にサーバ間分散が可能である。従ってこの仕組みを用いることで、初期導入費用の削減と市場が拡大した際の対応の両方が可能であり、トータルコストの削減に極めて有意義であると考えられる。

OpenSOAP がこのようなアミューズメント市場に対する可能性を示すことはこのシステムの普及において大きな意味を持つと思われる。

このような状況を鑑みて今回は OpenSOAP を用いたネットワークゲームの実装を試みた。SOAP を用いる際に大きな問題となるのは実行スピードであるが、今回は転送メッセージと思考ルーチンを一括転送することによってストレスのないシステムを構築した。

9.2. プロジェクト内での位置付け

このテーマにおいては、OpenSOAP サーバと OpenSOAP API を用いることでシステムを短期間で開発できることを実証することを目的としている。このため RAD プログラミングの手法を用いて実装されている。RAD とは Rapid Application Development で、一般的にはグラフィカルなユーザ・インタフェースの作成やデータベースとのインタフェース作成など、ある程度決まりきったプログラミングをやってくれるツールのことである。今回は OpenSOAP API を利用することによ

り、SOAP 関連の通信環境を簡単に実装できることを目的とする。

またもうひとつの目的として小規模システムにおいてコストをかけずにデータ授受が可能であることを実証する。特に今回はアミューズメントシステムという妥協が許されない中で OpenSOAP のフレームワークがどの程度有用であるかを検証する。

9.2.1. RAD 検証内容

RAD としての効果は、API 策定から短期間にシステムが構築されたことで実証される。今回のシステムでは、API の実装完了からシステム通信環境完成までの工数をもとに検証内容とする。

9.2.2. 環境構築コスト

サーバに関してのコストは、今回は実証システムということで、既存のサーバを流用した。ユーザ最大数などの実証はなされておらず、実用規模でのトラフィック検証などは今後の課題として残される。

一般的にサーバアプリケーションにおいては、以下の費用が見積もられる。

- A) ハードウェア費用
- B) サーバ構築費用
- C) DB
- D) サーバアプリケーション作成費用

ここで、D) のサーバアプリケーション作成費用は、今回は Web サービスの作成費用と基本システムであり、基本システムとしては OpenSOAP フレームワークを用いるので、単にサービス実装の費用が見積もられる。従って今回は D) の部分に着目してコストの論議をすることとする。

9.3. システム概要

本システムではカードゲームを対象にアミューズメントシステムを構築する。今回のカードゲームはマジック・ザ・ギャザリングを代表とする対戦型ゲームである。マジック・ザ・ギャザリングとは 1993 年に、アメリカの大学教授リチャード・ガーフィールド氏によって制作された「ファンタジー・トレーディング・カード・ゲーム」である。「マジック」は架空世界を舞台にして、2 人の「魔法使い」（プレイヤー）が様々な魔法（カード）を駆使して闘うカードゲームで、魔法にはクリーチャーを召喚したり、相手に直接火の玉をぶつかけたり、魔法の宝物を生み出すものなど様々な種類がある。カードにはその魔法を表す美しいイラストが描かれており、これを目当てにカードを買うコレクターもいる。

さらに「マジック」ではトレーディングカードの要素も含んでいる。「マジック」は一般のカードゲームと違い、1つの箱のカードはバラバラになっている。また「マジック」には1000種類を超えるカードがあり、店頭で普通に買って全種類を集めるのは困難となっている。そこで仲間同士などでカードを交換して、お互いのコレクションを高めるということもできる。このようなカードゲームの国内での市場規模は2001年度で、約1千億円程度である。

しかしこのカードシステムはその市場規模にもかかわらずネットワーク化されていない。その1つの要因としては、カードの設計がリアルタイムに対戦することを前提としており、ネットワーク上で不特定多数と対戦するようにはなっていないことが上げられる。カードゲームで重要なのは、どのようなカードを集めてデッキを構成するかという場合のカードのバランスであり、これをネットワーク上で行うためにはお互いのカードの認知が一致していることが求められる。しかしリアルタイムネットワークゲームとするためには追加カードの実装が必要であり、サーバーシステムへの負担は極めて大きいものとなる。

これらを鑑みて本システムでは、カードの持つアビリティを対戦することで変更(向上)させるシステム、そしてそれらを使用するための思考ルーチンを組にしてSOAPメッセージとして送出することで、相手がリアルタイムに存在していなくても対戦が可能となるように設計した。

すなわち、カードを生物のように考えその性能を動的に変化させる。長く保有することで性能が向上し、カード種類が少なくても、対戦結果の多様性を保持できる。さらに思考ルーチンを導入し、ユーザが直接戦わないようにする。ユーザとの対戦を学習することで、カードの戦闘方式が変化する思考ルーチンはユーザのエージェントとなり送出先で代理に戦闘をする。

このシステムによりユーザは任意の時間にゲームを楽しむことができる設計となっている。

ここでSOAPメッセージはカード情報と思考ルーチンを実現するためのパラメータを保持するために利用している。従来のシステムでは固定化されがちであったネットワーク情報伝達を、SOAPメッセージを用いてパッケージ化することで、ネットワーク通信の簡便化に留まらず、将来の拡張性にも対応可能となるような設計となった。

9.4. ゲーム内容

ここでは、ゲームシステムの内容について記述する。



図 9.4.1 ゲーム画面一例

9.4.1. 概要

本システムは基本的には対戦相手を選択して対戦するゲームである。対戦相手はサーバ上にリストとして持ち、相手の ID を教えてもらい登録することで増えていく。システム構成としては、例えば、メッセージサービスと同等のものであり、対戦相手の ID に戦闘情報を送付することで、対戦が開始される。戦闘情報は SOAP メッセージでパッケージ化されサーバに保存される。対戦相手はサーバから保存されたメッセージを取得し戦闘計算し結果を返信する。この作業で1つの戦闘が終了する。ここで、メッセージは表示前にサーバに返送される。従って、厳密なリアルタイムではないが、SOAP メッセージの送受信の時間を気にせずに、相互に同じ結果を見ることができる。

9.4.2. カード

カードはモンスターカード、リソースカード、エネルギーカード、罠カード・バトルカード・スキルカードからなる。リソースカード、エネルギーカード、罠カード、スキルカード、バトルカードをライブラリカードと呼ぶ
全カードに共通のパラメータは以下の3つである。

- ・名称

ユーザーA	1:サーバーへのログイン
ユーザーA	2:戦闘情報作成、戦闘相手選択
ユーザーA	2:戦闘相手(ユーザーB)への戦闘情報の送付
ユーザーA	3:サーバーから切断
ユーザーB	1:サーバーへのログイン
ユーザーB	2:自IDへのメッセージ取得
ユーザーB	3:戦闘計算
ユーザーB	4:戦闘結果をユーザーAへ返信
ユーザーB	5:戦闘結果の表示
ユーザーA	1:サーバーへログイン
ユーザーA	2:自IDへのメッセージ取得
ユーザーA	3:戦闘結果の表示

図 9.4.2 ゲームの流れ

- ・レアリティ (希少性。カードの価値を表す)
- ・所属カードセット (処理に必要な拡張プログラムを指示する)

これに追加してライブラリカードでは次のパラメータも持つ

- ・リミット (デッキに保持できる枚数)
- ・カテゴリ (カード分類。思考ルーチン用)

9.4.3. モンスターカード

モンスターカードは、カードの中核をなす部分で、戦闘の対象となるものである。これらについては以下のプロパティを持つ

- ・名前・種族(名) プレイヤーが決定する
- ・オーナー名 現オーナー名、最初のオーナー名の2種
- ・レベル
- ・HP
- ・攻撃力
- ・防御力
- ・素早さ
- ・精神力

- ・ レアリティ
- ・ 経験値
- ・ 必要経験値 非表示
- ・ 攻撃系統別耐性値 非表示
- ・ クリティカル確率 非表示
- ・ ステータス攻撃耐性 非表示

ステータスには以下の種類がある。

死亡、マヒ、眠り、気絶、呪い、毒、腐蝕
 錯乱、呆然、飢餓、不器用、 スキル封じ（地水火風無）、
 防御力低下、攻撃力低下、素早さ低下、精神力低下、攻撃成功率低下

9.4.4. リソースカード。

「リソース」は属性ごとのエネルギーを生み出す源であり、スキルカード等を用いるためには規定量必要となる。

リソースカードに必要なパラメータは以下の通りである。

- ・ リソースの属性 地水火風無のいずれか
- ・ リソース値 リソースから生み出されるエネルギーの
 最大量

9.4.5. エネルギーカード

エネルギーはエネルギーを発生させるためのカードであり、以下のパラメータを持つ。

- ・ 発生するエネルギーの属性 （地水火風無のいずれか）
- ・ 発生するエネルギーの量
- ・ 特殊カード属性

9.4.6. 罠カード・バトルカード・スキルカード

- ・ コスト スキル等を使用するのに必要な

	エネルギー
・ 効果	ステータス変更、パラメータ操作、属性変更、地形変更、通常、攻撃以外の基本攻撃。
・ 対象	敵全部、味方全部、敵味方全部
・ 継続時間	カードによって与えられた効果の続く時間
・ 表示エフェクト	既に組み込まれているエフェクトを名称で指定する
・ 起動条件	罨カードのみに実装。罨の起動条件
・ 素早さ	バトルカードのみ

9.4.7. 対戦計算

ユーザは対戦においては、デッキと思考ルーチンを選択し、現在登録されている対戦ユーザに送りつける。その結果以下の処理を行い対戦結果

対戦相手はデータを受信後、バトル処理を行う。

バトル処理は各カードの素早さパラメータに応じて順番に1つずつ処理される。

- 1) 攻撃対象決定 攻撃対象を決定する。攻撃にはライブラリカードが用いられるので、これを使用する場合には同時にライブラリカードを選択する。
- 2) 攻撃対象の防御 攻撃された側は持ち札によって防御を行う
- 3) 結果反映フェーズ 攻撃によって HP カウントダウンの処理がなされる。また、行動不能や特殊ダメージを被った場合はその処理がなされる。
- 4) バトル終了なら終了、さもなければ1) から繰り返す。

9.5. 構成環境

対象 OS は Windows 98/Me/XP である。サーバは特に制限はない。サーバへの接続は常時接続を想定している。ただし、一般通信回線でも 56k 以上のアクセス速度であれば、問題なく接続できる設計となっている。送受信に必要なデータ量はおよそ 2~10kb である。

SOAP サーバとのやりとりは HTTP プロトコルを用いるので、ネットワークゲームにあるようなファイアウォールの問題はない。接続もそれぞれ 2 度で終わるので、接続を保存する必要もない。

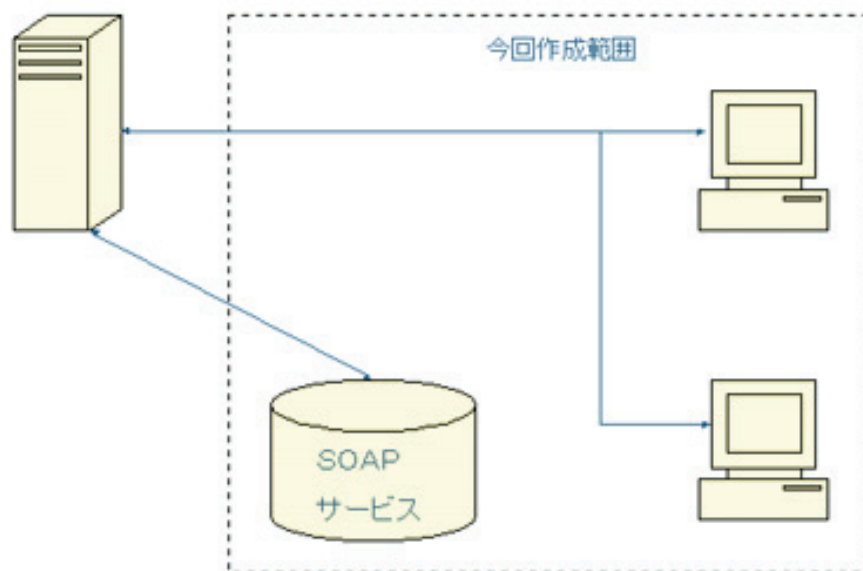


図 9.5.1 ネットワーク構成図

9.6. 接続メッセージの流れ

以上の構成で SOAP メッセージが作成されてサーバ経由で相手に渡される。このときのメッセージの流れは以下ようになる。

最初にユーザ 1 がログインを行う。これにより、ユーザの認証が行われ、ユーザ ID の有効性が確認される。そののちにユーザ 1 が対戦データを作成する。この処理ではカードの選択、対戦相手の選択などをユーザが行う。これによりカードデータと思考ルーチンを表現するためのパラメータが決定され、しかるのちにこのデータを SOAP メッセージとして変換する。SOAP メッセージには相手の ID も格納され、これを SOAP サーバ上のサービスへと転送する。サーバでは SOAP メッセージを保存する。

対戦データ送出フェーズ

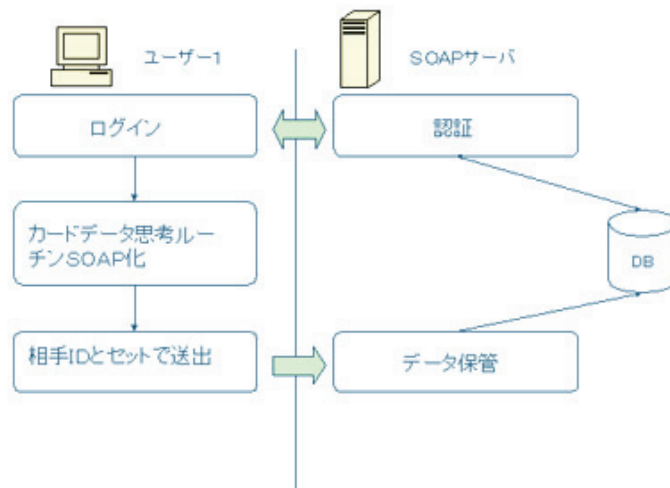


図 9.6.1 対戦データ送出フェーズ

対戦計算フェーズ

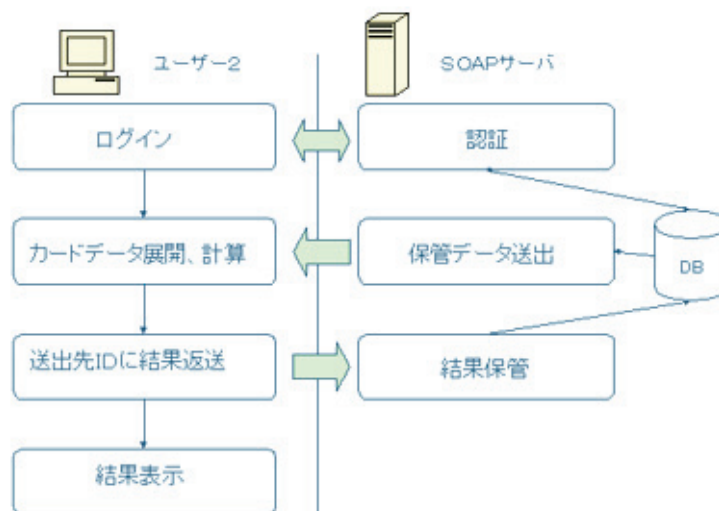


図 9.6.2 対戦計算フェーズ

送出されたデータは、戦闘相手のログインのタイミングで相手に送られる。データは、戦闘計算され、送り元の ID に対して返送される。そののちに表示が開始される。

返送された戦闘結果はユーザ 2 では連続して再生され、ユーザ 1 は再度ログイン後、結果ファイルとして受信する。そして時間差でユーザ 2 と同じ状況で再生

される。

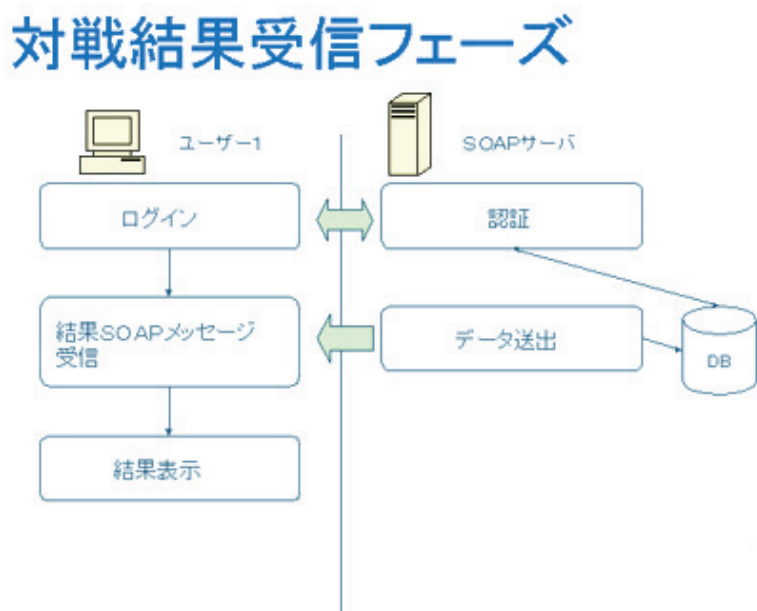


図 9.6.3 対戦結果受信フェーズ

9.7. 検証結果

以上のカードゲームについて実装を行った。結果として3ヶ月程度で通信関連の実装は終了し、短期間に接続が可能であることが検証された。また、すでにある既存のサーバにサーバアプリケーションを作成したが、実験システムではあるものの1ヶ月程度で実装が完了し、サーバーコンテンツ作成として有意義であることが検証された。

9.8. ビジネスへの取り組み

このシステムはあくまで実証用のものであるが、実際には販売等アカウントティングに結びついてはじめて意味があるものである。システム上良いとされたものも、実用上の可能性がないのであれば、単なる実験システムで終わってしまう。そこで、今回このようなシステムが実用のものであるのかという点において評価を試みた。

評価の方法としては、デモとシステムの概要についての資料を元に複数の雑誌社、インターネットプロバイダなどに評価をお願いした。数値化は困難であるの

で、回答としての例を上げることとする。対象は雑誌社3社、プロバイダー、インターネット関連等業者3社である。

- ・雑誌社

雑誌でのパブリシティ（記事としての公開）は問題ない
簡単にこのようなシステムが構築できるのであれば企画したい
現在カードゲームはホットな話題であり、実用として考えられる
どのような内容であるのかがわかりづらい
カードの配布についてどのようなポリシーがあるのか

- ・プロバイダー等

コンテンツとしては可能性がある
ブロードバンドコンテンツとして考えられるのか
月額使用料などの問題がクリアできれば企画としては面白い
実用システムとしてはどの程度の追加工数が必要なのか
実装上サーバーシステムとしての規模感はどの程度になるのか
ユーザから SOAP という仕組みは見えないのでポイントにはならない
このような回答が得られ、サーバを含めたコスト、告知の問題がクリアできれば実用となり得る感触を得た。

これを元にホームページでの告知作業を進めている。OpenSOAP としての取り組みとしては、あくまで業者向けとなるので、SOAP を用いた場合にどの程度のコスト削減となるのかということが差別化の意義となると思われる。

9.9. 結言

OpenSOAP サーバを用いることで、簡便なシステムでネットワークゲームを構築できた。また OpenSOAP が HTTP プロトコルを用いていることで、ネットワーク接続の際のファイアウォールの問題が解決し、アミューズメントシステムとして実用的であることが確認できた。

また、SOAP メッセージにデータを詰めて一括して送受信する方法で、リアルタイムに向かないとされるアミューズメント系コンテンツにおいても方法次第では実装が可能であることを示した。このため SOAP のメッセージの遅延が問題としないことを確認した。