

特別研究報告書

クライアント・サーバモデルにおけるユーザグループ形成支援ソフトウェア

Support software for creating user groups using a client server model

指導教員

野中弘二

報告者

学籍番号:1055098

氏名:松本 憲治

平成 15 年 1 月 27 日

高知工科大学 電子・光システム工学コース

クライアント・サーバモデルにおけるユーザグループ形成支援ソフトウェア

Support software for creating user groups using a client server model

目次

第1章 背景

1-1 マルチメディアの現状と将来	4
1-1-1 動画・音声の配信	
1-1-2 ネットワークゲーム	
1-1-3 インターネット電話	
1-2 ブロードバンド通信	8
1-3 ネットワーク利用の現状と研究すべき課題	9

第2章 研究の手法

2-1 研究の意義	11
2-2 ソフトウェアの機能イメージ	11
2-3 本研究の検討内容	12
2-4 ソフトウェア開発の段階的目標	12

第3章 ソフトウェアの設計

3-1 プログラム開発におけるコンセプト	14
3-2 動作確認版プログラム	15
3-2-1 動作確認版プログラムの特徴・仕様	
3-2-2 プログラムの機能・役割	
3-2-3 動作確認版プログラムの動作チェック	
3-3 動作確認版プログラムの改良	17
3-3-1 グループ形成支援ソフトウェアの特徴	
3-3-2 グループ形成支援ソフトウェアの仕様	
3-3-3 プログラムの各部名称とその機能・役割	

4 処理の流れとソースコード

4-1 プログラム製作環境	24
4-2 グループ形成支援ソフトウェアの処理の流れ	24
4-3 グループ形成支援ソフトウェアのソースコード	28
4-3-1 グループ形成サーバ側プログラムのソースコード	
4-3-2 クライアント側プログラムのソースコード	
4-4 考察	50
4-4-1 ソフトウェアの特化すべき機能	
4-4-2 他のソフトウェアとの比較	
4-4-3 ソフトウェアの改善すべき点	

5 まとめ

5-1 本ソフトウェアの作成、検証から得られ知見	53
5-2 発展性・応用の可能性	53
5-3 ユーザがネットワークを利用したソフトウェアに求めるもの	54
感想・謝辞	55

第1章 背景

1-1. マルチメディアの現状と将来

2002年現在、ネットワークへの高速なアクセス手法として ADSL や、一部ではさらに高速なアクセス手段である光ファイバネットワークが急速に普及している。それらの普及に伴うように、高速、大容量通信を活かしたネットワークコンテンツが出現している。現在サービスが拡大しつつあるネットワークコンテンツとしては、動画や音楽の配信、オンラインゲームや IP 電話などがある。本章では、これらのネットワーク上で展開するコンテンツの現状について説明し、近い将来の発展形態の予測を行い、研究すべき課題を述べる。

1-1-1. 動画・音声の配信

昨今、インターネットのホームページ上で、ニュース映像やショートドラマ、アーティストのライブ映像の中継、映画、アニメ、そしてインターネットラジオなどの動画・音声の配信をするサイトが着実に増加している。

これらのコンテンツの供給量が増加した理由は、ブロードバンド通信が普及したということも然る事ながら、ソフトウェアによる動画・音声圧縮の技術の発達という部分、さらにプロセッサの高速化などによるコンピュータの性能向上という部分がより大きな要因になっていると考えられる。例えば、ソフトウェアの技術要素としては、動画圧縮であれば MPEG-4 (Motion Picture Experts Group-4)、音声圧縮なら MP-3 (MPEG-1 Audio Layer3) 技術が標準的手段としてあり、そして、ハードウェアでは、3GHz を超えるプロセッサも登場している。

近い将来、超高速ネットワークの普及や、ハードウェア、ソフトウェアの更なる発展によって、(著作権の問題や法規制などクリアしなくてはならない問題は多々存在するが)、放送局、映画会社などが蓄積している豊富な映像作品が瞬時に見られるようになったり、ネット配信を前提とした映画などでもできるだろうという見通しがなされている。さらには、ネットワークを利用して映像を見ることができ、常設カメラで、介護や留守宅の安全監視などを行えるカメラを製品化している企業もある。そして、現在は企業を中心とした動画・音声の配信元が主であるが、近い将来には、個人が映像や音声を配信したり、ネットワーク上で映像や音声の素材を持ち寄り、映像の編集を行え、今までは映像や音声などの大容量データを個々人が記憶媒体に保存し、その記憶媒体を直接渡すことで作業していたものが、ネットワーク上でほとんどリアルタイムに行えるようになると思われる。

1-1-2. ネットワークゲーム

ネットワークゲームは、近年、アメリカや韓国で楽しまれており大変盛んである。その多くがパソコン用ゲームで、国産のタイトルと比べて海外製のタイトルが多く、例えば「ウルティマ オンライン」や「ディアブロ II」、「リネージュ」などがある。日本では、2002年5月、家庭用ゲーム機「プレイステーション 2」で「ファイナルファンタジー XI」が発売された。発売された当初には、ゲーム購入者からのサーバへのアクセスが殺到して、サーバがダウンするなどの重大なトラブルが発生し、ネットワーク

ゲームのさまざまな問題点を浮き彫りにした。各方面からは失敗との声や、技術的に半成熟であったなど、さまざまな賛否両論の声が挙がった。この「ファイナルファンタジーXI」は、MMO (Massive Multiplayer Online)タイプのロールプレイングゲームである。ユーザのアクセス形態としては図 1-1 に示すようなネットワーク構成であるが、図の分析してみると、ボーダールータに入った情報がディストリビューションで振り分けられる、ディストリビューションスイッチ以降は系統的な処理を行うスター型のネットワーク構成をしているが、認証サーバにアクセスが殺到した場合、サーバが容易にダウンする可能性が高いと考えられる。

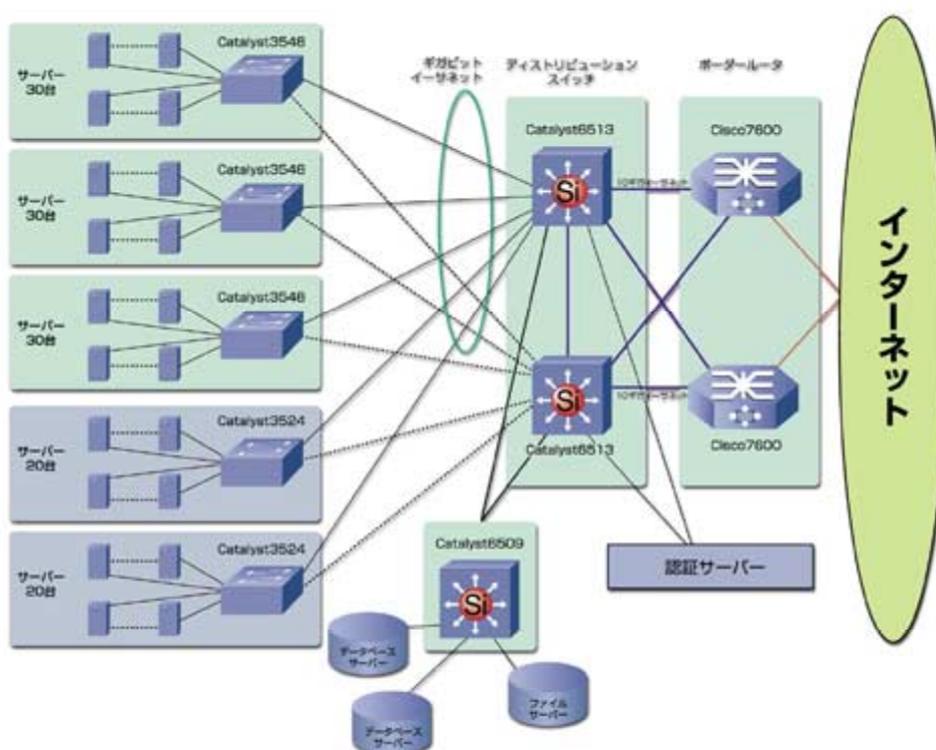


図 1-1 ファイナルファンタジーXIのネットワーク構成

<http://www.cisco.com/japanese/システムズより引用>

実際に、どんな障害が発生したかを調べた結果、「入会及び購入処理遅延に関する障害」と「サーバ群の基幹となる接続機器の障害」の2つが主因であるとの分析がなされていた。この2つの主因から推測するに、原因の前者は認証サーバ、後者はディストリビューションスイッチの処理負荷に問題があったと考えられる。

そこでネットワークを利用したゲームタイプのアプリケーションの接続形態を分類して、その利点、課題を整理してみた。

ネットワークゲームの接続タイプ

接続タイプ別に分類すると以下のような2種類のタイプに分類できる。

- MMO (Massive Multiplayer Online)タイプ

大勢のプレイヤーで1つのサーバを共有するタイプのネットワークゲームで、数千人規模で同時にプレイすることが可能である。ゲームメーカーが保有するゲームサーバへの接続はインターネットを使用することで行う。

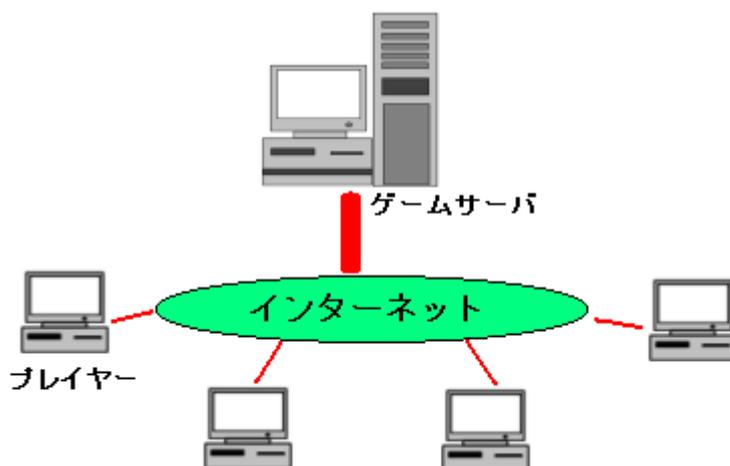


図 1-2 MMO タイプのネットワーク構成

表 1-1 MMO タイプの特徴

ネットワークの構成	インターネットを通じてサーバに接続するタイプ
利用人数	数千人規模（サーバのスペックに依存）
導入費用	サーバの導入に多大な費用がかかる
システムに障害が発生した場合	ネットワークの構成上、サーバに障害が発生した場合プレイヤーが全く利用できなくなる可能性がある。

・ピアツーピアタイプ

プレイヤーの誰かが、ゲームのホストとしての起動を行い、そのホストに対して各プレイヤーが接続するタイプのゲームで、同時にプレイできる人数(ホストへの接続台数)は 2～32 台程度が一般的である。

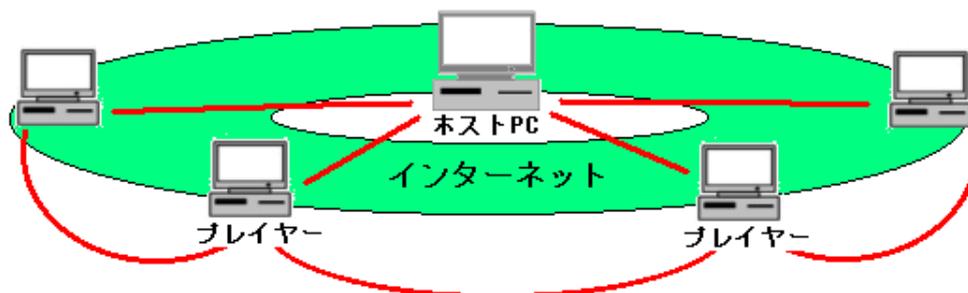


図 1-3 ピアツーピアタイプのネットワーク構成

表 1-2 ピアツーピアタイプの特徴

ネットワークの構成	ホストとして起動したプレイヤーに、他のプレイヤーが接続
利用人数	2～32 人程度
導入費用	ほとんど費用はかからない。
システムに障害が発生した場合	トラブルの発生したホストに接続していたプレイヤーのみ利用ができなくなる。

ネットワークゲームのこれから

ネットワークゲームは現在盛んになりつつあるコンテンツではあるが、解決しなければならない問題点は多数ある。例えば、上記したようなネットワークの構成に関する問題、そして、サーバなどの導入にかかる莫大な保守・運用費用の問題などである。一方で、ネットワークゲームで成功を修めている企業があるのも事実であり、ネットワークゲームの直面する課題を一つ一つクリアしていくことによって、市場は順次拡大していくと考えている。

1-1-3. インターネット電話

BB-Phone などで最近話題になっているキャリアサービス形態は、加入者同士の通信料金は無料、そして、加入電話ならどこへ掛けても 7.5 円である。これらに代表されるインターネット電話サービスの技術自体は古くからあり、1974 年にインターネットの前身である ARPANET ですでに実験は行われていた。インターネット電話の仕組みは、まず音声信号をデジタル信号に置き換え分割、IP ヘッダを付加したパケットによって音声の伝送を行うのだが、それらのパケットを伝送する伝送路が電話網でなく、インターネット網で伝送されるという仕組みになっている。電話網ではなく、インターネット網を利用するので、各回の料金にアクセス回線のみのコストと固定のネット加入料ですみ、通話料金が安くなるのである。料金が安くなるだけでなく、音声データを IP ヘッダを付加したパケットに変換することによって、トラフィック内で音声とデータを同時に扱うことができるようになる。しかしながら、インターネット電話には欠点があり、インターネットなどの IP 網では QoS (Quality of Service) が保証されていないので、トラフィックが増えると、音声パケットの到達遅延が大きくなったり、パケットが消失する頻度が増し、音質が悪化する可能性がある。しかし、これらの欠点もサーバの処理能力向上やネットワークの転送速度向上によってパケット転送遅延が低下してゆけば改善していくと思われる。

1-2. ブロードバンド通信

ブロードバンド通信で主に使われている技術は、ADSL (Asymmetric Digital Subscriber Line)で、これらの回線の最高速度は、1.5Mb/s、8Mb/s、そして新たに 12Mb/s の回線速度のサービスが開始されようとしている。ADSL の仕組みは、音声通話用の帯域(300Hz～3.4kHz)より高い周波数帯域(4kHz 以上)を利用して広帯域デジタルデータ通信を行う技術である。ADSL で標準化されている帯域は、1.1MHz までの帯域を使用する G992.1 勧告の G.dmt 方式(フルスペック)と、その半分の 552KHz までの帯域を使う G.992.2 勧告の G.lite 方式(ハーフレート)の 2 種類がある。さらに、G.dmt 方式、G.lite 方式の規格それぞれに「AnnexA」北米仕様、「AnnexB」欧州仕様、「AnnexC」日本仕様の 3 つの地域標準が存在する。主に、日本で使用されている規格は、ITU-T G992.2 AnnexC と、ITU-T G992.1 AnnexC で、前者は最高 1.5Mb/s、後者は最高 8Mb/s の通信速度を有している。上記の規格で定められたように、ADSL は高周波数帯域を使用するため、音声通話用に使用される低周波数領域と ADSL が使用する広帯域デジタルデータ通信領域が使用する周波数領域との干渉が発生する。これにはスプリッタという音声通信用のアナログ信号と ADSL の高周波数の広帯域デジタルデータ信号を分離する装置によって干渉の問題を解決している。しかし、万能の高速通信と思われる ADSL にも欠点がある。一つ目の欠点として、メタルケーブルを使用しているため、利用者と電話局との距離が遠くなるほど周波数帯域が狭くなり、実際に伝送可能な通信速度が低下すること、二つ目として、敷設ケーブル束内の ISDN 線と ADSL 線で周波数が大きく干渉し合うということである。一つ目の欠点は避けがたいものであるが、二つ目の欠点に関しては、日本仕様の「AnnexC」によって ISDN 線に強い信号が載っているときは、ADSL 側の帯域を狭めて通信速度を落とし、ISDN 線の信号の影響が少ないときには広帯域で通信をするという仕組みを取ることによって課題を克服している。

現在は、ADSL によるブロードバンド通信が盛んであるが、その他には CATV の同軸ケーブルを利用したモデムアクセスや、光ファイバを用いた 100Mb/s の超高速光通信などのサービスが開始されている。後者の光ファイバを用いたものは、サービスエリアがまだ限られており、2002 年 9 月末現在 10 万回線程の加入者しかおらず、普及率はまだまだ低い。

表 1-3 ADSL 規格 (2001 年 DOS/V magazine より抜粋)

ITU-T	方式	帯域(Hz)	仕様
G992.1	G.dmt (フルスペック)	1.1MHz	AnnexA (北米仕様)
			AnnexB (欧州仕様)
			AnnexC (日本仕様)
G992.2	G.lite (ハーフレート)	552kHz	AnnexA (北米仕様)
			AnnexB (欧州仕様)
			AnnexC (日本仕様)

1-3. ネットワーク利用の現状と研究すべき課題

現在、各家庭にブロードバンド(高速大容量)通信が急速に普及しつつある。総務省によると、2002年8月末現在の加入者数は、約580万人で、昨年秋からの加入者の増加が急速で、今後1年以内に加入者が1000万人を超えるとの予測データも発表されている。これら利用者増加の背景には、折からのパソコンなどに代表される情報機器の家庭への普及ともあいまって、通信関連企業の競争激化による利用料金低下や、速度の向上や信頼性、利用できるコンテンツが豊富になるなどさまざまな要因が含まれていると考えられる。ブロードバンド(高速大容量)通信網の発達によって、高速大容量を活かしたサービスやコンテンツも徐々に充実してはいるが、現状は本格的サービス提供開始へのテストや利用者へ高速大容量を活かしたコンテンツやサービスを認知、浸透させるための足場固めのニュアンスのものが多い状態である。現時点で頭角を現しつつあるネットワークコンテンツとして、動画や音楽、ネットワークゲームなどのエンターテインメントや娯楽の要素を含むものが多い。通信のサービスに関連するものとしてはインターネット電話などを挙げられる。このように、ネットワークのアクセス環境を取り巻く市場は急成長しつつあるが、それに伴うサービス、アプリケーションを企画、開発する人材の供給不足の問題も発生している。なかでも、企業内の通信網を管理、補修を行う技術者や、新しいコンテンツの企画、製作など専門分野の人材不足が深刻になりつつあり、市場の発展を妨げかねないという懸念の声も出ている。

結局、6、7年前までは、一般的にコンピュータの要素といえばハードウェアとソフトウェアであった。今まで述べてきたように、近年になって、ハードウェアとソフトウェアというコンピュータの要素にネットワークという要素が結びつくようになり、現在では、ハードウェア、ソフトウェア、ネットワークが複雑に絡み合っただけでなく、現在のコンピュータネットワークの社会を構築している。そして、ネットワークやネットワークを取り巻いているインターネットのコンテンツは、休むことなく日々進化を続けているのである。これからの時代、ネットワークを利用したコミュニケーションは、今のような一方的なものではなく、リアルタイム性と双方向性を持つものとなり、それによりネットワーク上にて共同で作業を行うといったような、ネットワーク協調型のコンテンツが登場すると考えている。ネットワーク協調型のコンテンツは、例えば、企業間の製品開発、医療や介護の分野、趣味・娯楽のエンターテインメントの分野など、さまざまな場面で活躍するようになると思われる。そして、ネットワーク上にて共同で作業を行うことによって、優れた人材をさまざまな場面で共有して使うことができるようになり、前記したような人材不足を解消する技術となりえると考えられる。ここにネットワークを介して地域的に離れた人々がさまざまな共同作業を行えるアプリケーションソフトウェアの開発を研究すべき意義があると考えられる。

このようにネットワークを利用したソフトウェアは、使い方次第で無限の可能性が秘めていると思える。そこで、ネットワークを利用したソフトウェアの開発するための基礎知識を習得するとともに、その知識をもとにネットワーク上で自由にグループを形成してコミュニケーションすることを支援するソフトウェアを開発することを本研究の目的とする。

この手法の有効性を確認するためには、3つのステップが必要となる。

1) ネットワークを媒体として、リアルタイムで情報を共有し、共同で作業を行うことを支援するプラ

ットフォームの構築。

2) ネットワークの複装などさまざまな課題を意識させずに作業可能なネットワーク情報交換手法、配置。

3) 協調作業が有効なさまざまな具体的作業ソフトウェア機能の実装。

本論分では、このうち1)、2)の課題について検証するため、最も基本的なテキスト情報を動的に変動するグループ内で共有する「グループ形成支援プログラム」を作成し、動作検証、ユーザの使用感を調査した結果を報告する。次節では、支援プログラムの構造と機能を説明する。

～参考資料～

[1]「高知新聞」2002年11月8日夕刊

[2]「DOS/V magazine」2001年1月15日

[3]「DOS/V Power Report」2002年4月

[4]「DOS/V Power Report」2002年12月

[5]「電子情報通信学会誌」Vol.82 No.5 pp.523-525

[6]「情報処理学会研究報告」Vol.2002 No.95 2002-DSM-27 pp.53-58 平原貴行 山之上卓 他

[7]「情報処理学会研究報告」Vol.2002 No.32 2002-DPS-107 pp.127-132 本多芳三 太田裕之

[8]「情報処理学会研究報告」Vol.2002 No.118 2002-DPM-28 pp.67-72 堀内俊之 長谷川直樹
他

[9]CNET Japan Tech News <http://japan.cnet.com/News/>

[10]アスキーデジタル用語辞典 <http://yougo.ascii24.com/>

[11]シスコシステムズ <http://www.cisco.com/japanese/>

第2章 研究の手法

2-1. 研究の意義

ブロードバンド通信が普及するに従い、ネットワークは双方向性とリアルタイム性を持つようになり、そのような通信環境が整えば、ネットワークを介して共同で作業が行われるようなアプリケーションが必要になると考えられる。そこで、本研究はそれらのアプリケーションを開発するための基礎部分であるネットワークの構成、作業グループの形成のためのプラットフォーム作りを行い、次節以降に説明するようなグループ形成支援ソフトウェアを作成する。

2-2. ソフトウェアの構成イメージ

作製する「ユーザグループ形成支援ソフトウェア」の機能イメージは、以下のような構成のものとなる。

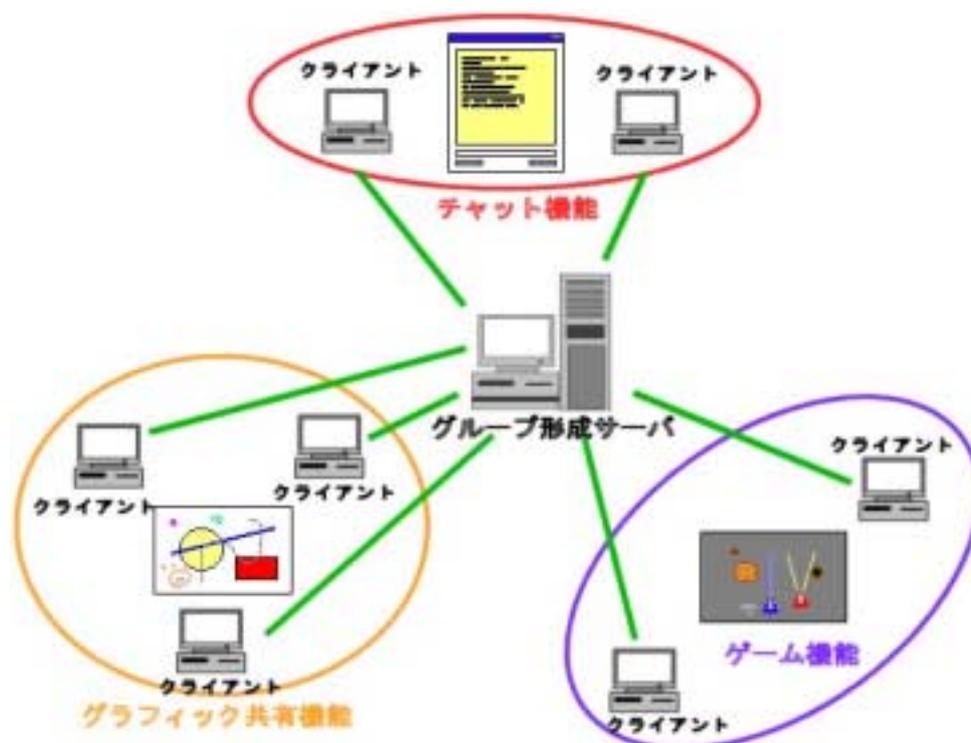


図 2-1 ユーザグループ形成支援ソフトウェアによるユーザグループ形成支援イメージ

この構成図では、グループ形成サーバがすべてのクライアントの情報を把握しており、その管理のもとにユーザ情報を共有したクライアントが自由にグループを形成する。ゲームやグラフィックボードを共有して図形をデザインしたり、ファイルを共有、相互修正したり、会話(チャット)などを行えるものをイメージしている。

そして、グループを形成する際、グループを形成したいメンバーを文字による情報の羅列の中から選択するのではなく、直感的にわかりやすいグラフィック(クライアントをキャラクター化)に

置き換え、そのグラフィックに対してボタンをクリックするなどのアクションを行い、メンバーを集めグループを形成するという方式を考えた。

次に技術的な要素だが、グループ形成サーバはクライアントの管理とグループ構成の管理だけを行い、クライアントの個々のプログラム内に作業用のサーバの機能を持たせることによって、クライアント同士が自由にグループ形成を行い、グループ形成サーバの管理を介すことなくさまざまな作業を行うことが可能になる。サーバを介すことなく個々の作業を行うことが可能になることにより、

- ・ サーバに対する負荷が軽減される。
- ・ サーバの処理待ちによる遅延が解消される。
- ・ データ流入の集中によるサーバのトラブルが減る。
- ・ クライアントが行うデータ送信の自由度が増す。

以上のような利点が生じる。

2-3. 本研究の検討内容

最終的には前記に示すようなソフトウェアの開発を目指す。本論分では、研究期間内にグループ化支援という基本機能の確認をするという目標に焦点を絞った。この基本動作の確認プログラムの作製を行った結果を報告する。本ソフトウェアでは、前節で示すような付加機能(ゲーム・絵を共同作製で描けるなど…)のユーザアプリケーションの高度なものの搭載は割愛し、本研究の基礎となるネットワーク部分のプログラム作成、クライアントの状態や情報の管理、およびクライアント同士のグループ形成部分のプログラムの作成と、その動作検証に注力する。アプリケーションにデータ送信の共有による共同作業の動作確認には、その機能としてテキストデータの表示する会話(チャット)機能のみにとどめる。機能分散処理タイプのユーザグループ形成支援ソフトウェアを作成することを本研究の到達目標にした。

本研究で作製するプログラムで確認する機能

1. クライアント・サーバーモデルによるネットワーク接続動作
2. クライアントの状態や情報の管理する部分
3. クライアントのグループ形成・移動・解消の支援
4. 会話(チャット)機能の実装

2-4. ソフトウェア開発の段階的目標

本研究におけるソフトウェアの作成にあたり以下の2段階で機能を確認する。

1. 本ソフトウェアのプログラムの骨子となる、全処理をサーバが行うタイプのグループ形成支援型のソフトウェアを製作し、その動作チェックを行う。
2. 1.で得られた結果をもとに、2-2 節のソフトウェアの機能イメージに示すような構成を実現

できる機能分散処理タイプのユーザグループ形成支援ソフトウェアの作製する。

1.に関してはプログラミングのスキルアップも含め、サーバ、クライアントタイプのプログラミング方法を身に付け、完成したソフトウェアの動作チェックを通しての改良点、問題点などを把握し、次のソフトウェア開発につなげるためのものである。

2.については、1.の部分で判明した改善すべき点、問題点を踏まえ、ソフトウェアの概要に示すような機能を実現できるソフトウェアの開発を行うための中間目標である。

以上のような目標を念頭におき、ソフトウェアを作成した。

なお、検討の結果、サーバへの通信・処理の負荷が集中することによるパフォーマンス低下が無視できないことが実験的に明らかになり、それを回避するための手法を取り入れたプログラムも検討した。

第3章 ソフトウェアの設計概要

3-1. プログラムの開発におけるコンセプト

本研究において、作成するプログラムのコンセプトは以下のようなものになる。

(1) ネットワークのプロトコルはTCP/IPを利用する。

TCP/IPプロトコルの逐次確認を利用することで安定したデータ送信の環境を提供することが可能となる。

(2) クライアント・サーバモデル形式の構成にする。

すべてのユーザ(クライアント)の情報管理を統括するためにクライアント・サーバモデルの方式を採用する。

(3) ユーザ同士が自由で簡単にグループを形成できるものにする。

ユーザがグループを形成するのに、文字の羅列から選択するのではなく、グラフィックを利用して感覚的にグループ形成ができるよう以下に示すような方式を盛り込む。

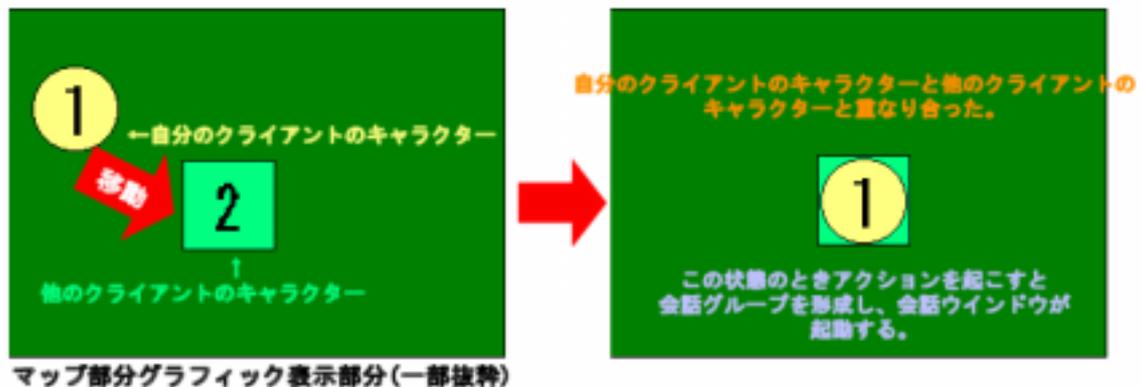


図3-1 グループ形成の方法

多数のクライアントがソフトウェアのマップ上に各自のキャラクターアイコンを持ち、そのキャラクターアイコンを操作し、他のクライアントに対してボタンをクリックするアクションを起こしたときに会話のグループを形成し、会話(チャット)が発生するというものである。

(4) ユーザが直感的に利用しやすいGUI(グラフィカル・ユーザ・インターフェース)にする。

ユーザがあまりネットワークという概念を意識することなく、ネットワークの設定は最小限にとどめ、プログラムによって自動的にネットワークの切断・接続の処理を行えるようにする。

3-2. 動作確認版プログラム

3-2-1. 動作確認版プログラムの特徴・仕様

本研究において前段階として動作確認のために製作したクライアントからの要求をすべてサーバが処理するタイプのプログラムを動作確認版プログラムと定義する。

動作確認版プログラムは、クライアント・サーバの構成によるプログラムで、クライアントすべての情報をサーバが管理・統括し、クライアントからの要求の処理をサーバがすべて行うタイプのプログラムである。それぞれの会話グループは、マップ部分を均等に4つの部分に分け、その分けた一つ一つの部分に各々ポート番号を割り当てることでグループ分けを行った。

このプログラムにおいては、動作確認ができればよいので機能を絞り、**最大クライアント接続台数は10台、最大会話グループ数は4つ**と設定して作製した。

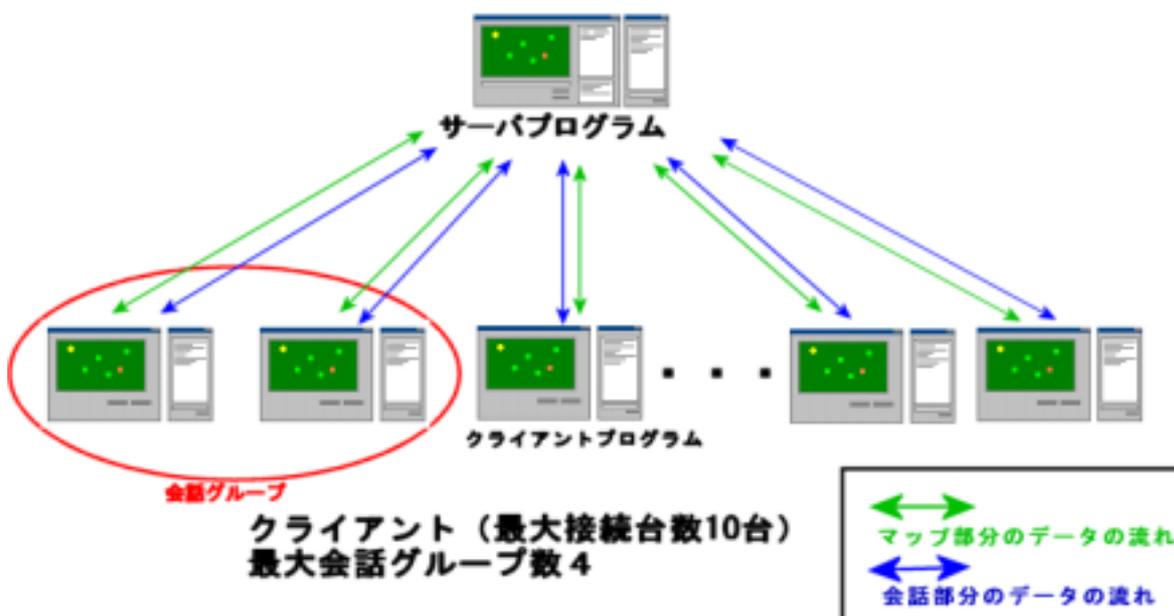


図3-2 動作確認版プログラムの特徴・仕様

3-2-2. プログラムの機能・役割

このプログラムは、クライアント・サーバの2種類のサブプログラムで構成されており、クライアント、サーバそれぞれのプログラムのサブプログラムを機能モジュール単位で分割すると、マップ部分と会話部分のプログラムに分かれる。

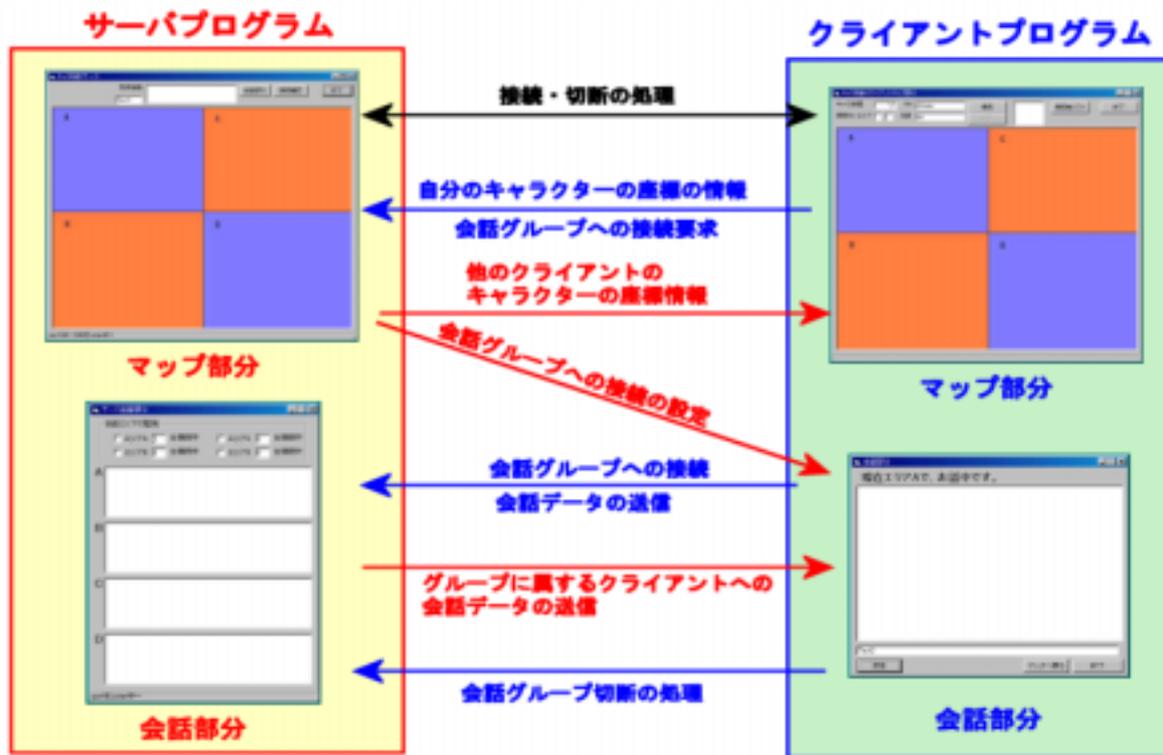


図3-3 動作確認版プログラムの機能・役割

クライアントのマップ部分における役割は、会話(チャット)をするためのメンバーを探しグループを形成する役割である。

クライアントの会話部分の役割はマップ部分でグループを形成したメンバーで、会話部分にて会話(チャット)を行う。

サーバのマップ部分の役割はクライアントの位置を逐一表示し、クライアント要求(接続・切断・会話要求など)をチェックしている。

サーバの会話部分の役割は会話グループを形成したクライアントの全会話データの内容を受信し、それぞれのグループに接続しているクライアントに対して会話のデータの送信を行っている。

3-2-3. 動作確認版プログラムの動作チェック

製作した動作確認版プログラムを使って、設定最大ユーザ数の10人が同時に使用する状態で動作テストを行い、プログラムの動作不良のチェック、このソフトウェアに対するアンケートを行った。

その中で以下のような、不具合、プログラムを改善すべき点が発見された。

問題点・原因

- 1.サーバの一つのポートに対するクライアントからのデータの流量が大きい場合、サーバの処理が追いつかず、クライアントからサーバへの要求が通らないことがあった。
- 2.すべての処理をサーバに任せているため、サーバに負担がかかりすぎ間欠的にエラーが発生した。

解決方法

- 1.については、サーバ、クライアントともにポートを追加して、クライアントからの要求を処理するポート、クライアントの情報(座標情報)を処理するポートを分離させることで問題の解決が可能である。
- 2.については、ソースコードを見直し、データを格納する変数の効率化、ソースコードの最適化を行うことで解決可能である。

アンケートより寄せられた意見

- 1.GUI(グラフィカル・ユーザ・インターフェース)がユーザに対して不親切な面があり、操作に戸惑った。
- 2.グラフィックやソフトウェアの実用面で不備(グラフィックが貧弱)などの指摘。

具体的な意見

- ・誰とチャットしているのか表示してほしい。
 - ・キャラクターが単調でわかりにくい。(キャラクターの上にハンドルネームなどを表示できるとよい。)
 - ・会話部分で誰と話をしているのかリアルタイムで表示してほしい。
 - ・会話グループにどれだけ人がいるのかわからない。(会話グループの表示。)
- などの意見があった。

3-3. 動作確認版プログラムの改良

動作確認版プログラムの動作チェック時に発生した問題点の解決手法、アンケートからの意見反映を目的に、ネットワークによる機能分担処理を用いるグループ形成支援ソフトウェアの設計を行った。

3-3-1. グループ形成支援ソフトウェアの特徴

動作確認版プログラムは、クライアントから送信されたデータ、クライアントからの要求など、すべての処理をサーバが行うことで動作していた。したがって、マップ部分の処理機能、会話部分の処理機能を1台のサーバが行っている。このため、サーバに処理が集中する。サーバマシンの処理能力が低かった場合、サーバプログラムがストップする可能性がある。そこで、サーバマシンに集中する負担を軽減するという意味で、今まではサーバで行っていた会話部分の処理を、会話サーバという概念を導入し、会話グループのクライアント間で実現する改良を行った。このソフトウェアが、グループ形成支援ソフトウェアである。このソフトウェアでは、サーバはグループの形成のための処理(マップ部分のクライアントのキャラクターの座標情報の管理、処理。クライアントからの接続要求の処理。)のみに集中するため、サーバの名称をグループ形成サーバとする。ちなみに、動作確認版プログラムではサーバと呼ぶ。

サーバで行っていた会話部分の処理をクライアント間で実現するとなると、会話グループを形成したクライアント間で会話部分の処理を行う会話用のサーバを設けなければならない。クライアント内に会話用のサーバを設けるということはネットワーク上の双方向の通信処理の手順が複雑になるが、サーバの処理負担を軽減することができる。具体的に会話部分の処理を会話グループのクライアント間で実現する処理は以下のようなものを考えた。

- (1) 会話グループへの接続要求の処理、会話グループの設定は、グループ形成サーバプログラムで行う。

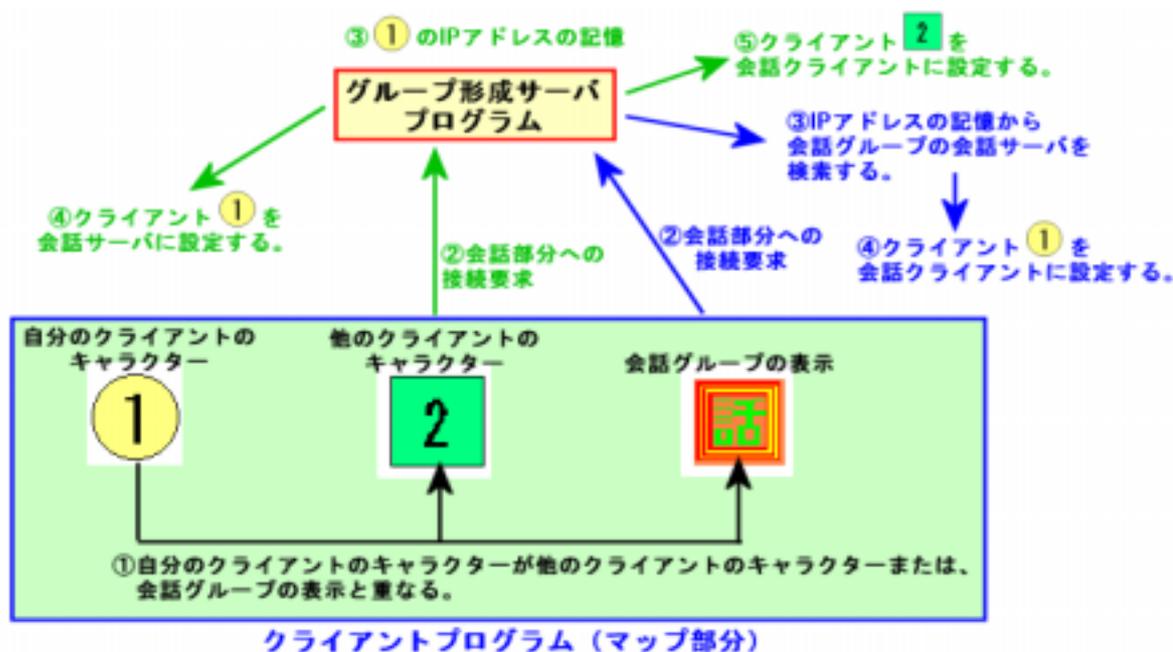


図3-4 グループ形成支援ソフトウェアの会話部分への接続要求の処理の概念

上図に示す処理は、グループ形成サーバへ、クライアントから会話要求を送信したときの処理

である。クライアントのマップ部分は、自分のクライアントのキャラクターアイコンが、他のクライアントのキャラクターアイコン、または会話グループの表示と重なるかを判断し、グループ形成サーバへの接続要求を行っている。クライアントからの接続要求を受信した、グループ形成サーバは、クライアントが、どのキャラクターと重なったかを判断し、会話部分への接続の設定をする。

他のクライアントのキャラクターアイコンと重なった場合、接続要求を行ったクライアントに対しては、会話サーバとなるための設定を行う。具体的には、接続要求を行ったクライアントのIPアドレスを送信してもらい、そのIPアドレスをグループ形成サーバ内のメモリに記憶する。そして、そのメモリ内に記憶されたIPアドレスを会話要求を行ったクライアントと重なり合っていた他のクライアントのキャラクターを持つクライアントに送信することにより、会話部分への接続が完了する。

会話グループの表示と重なった場合、グループ形成サーバは、受信された情報をもとに会話要求を行ったクライアントがどの会話グループと重なったのかを判断する。その会話グループの会話サーバとなっているクライアントのIPアドレスを、IPアドレスを記憶しているメモリ内から検索し、会話要求を行ったクライアントに対して、そのグループの会話サーバとなっているクライアントのIPアドレスを送信する。IPアドレス情報を受信した会話要求を出したクライアントは、その受信した情報をもとに会話サーバへの接続を行う。

(2) クライアント内に発生した会話用のサーバは、会話用のサーバであっても自由に会話グループから抜け出すことができる。

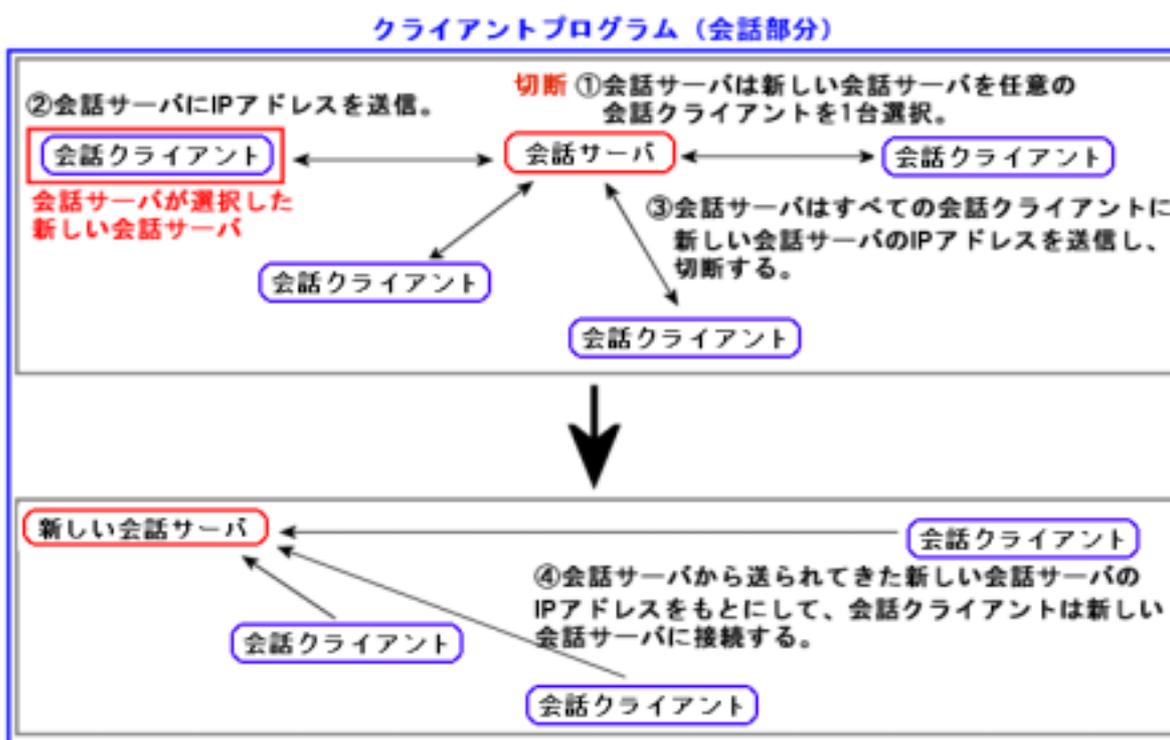


図3-5 会話サーバの切断時の切り替え処理

本研究のグループ形成支援ソフトウェアにおいては、会話サーバは、サーバでありながら自由に会話部分から退出可能であることが大きな特徴である。これは、会話サーバを切り替える機能が会話サーバのプログラムに搭載されているからである。この会話サーバの切り替え機能により、会話サーバ、会話クライアントともに、よりフレキシブルな会話グループの形成が行える。会話サーバ切り替え機能の処理は図3-5に示すようなものである。

会話サーバが会話部分を退出する操作を行うと、会話サーバは、まず上図①に示されたように、現在会話サーバに接続されている会話クライアントの中から、任意の会話クライアントを1台選択し、そのクライアントに対して、新しい会話サーバになるよう命令を送信する。会話サーバから命令を受けた会話クライアントは、自分のクライアントのIPアドレスを会話サーバに送信する。このとき、同時にグループ形成サーバにも新しい会話サーバの更新情報を送信する。会話サーバは、新しく会話サーバとなる会話クライアントから受信したIPアドレスを現在接続している全ての会話クライアントに対して新しく会話サーバとなる会話クライアントから受信したIPアドレスの情報を送信した後、自らを切断する。

会話サーバから受信した新しく会話サーバとなる会話クライアントのIPアドレスに対して会話クライアントは接続を行う。これら一連の作業を行うことにより、会話サーバはサーバでありながら自由に会話部分から退出することが可能になるのである。

3-3-2. グループ形成支援ソフトウェアの仕様

このソフトウェアが実現するネットワーク構成は以下のような構成である。

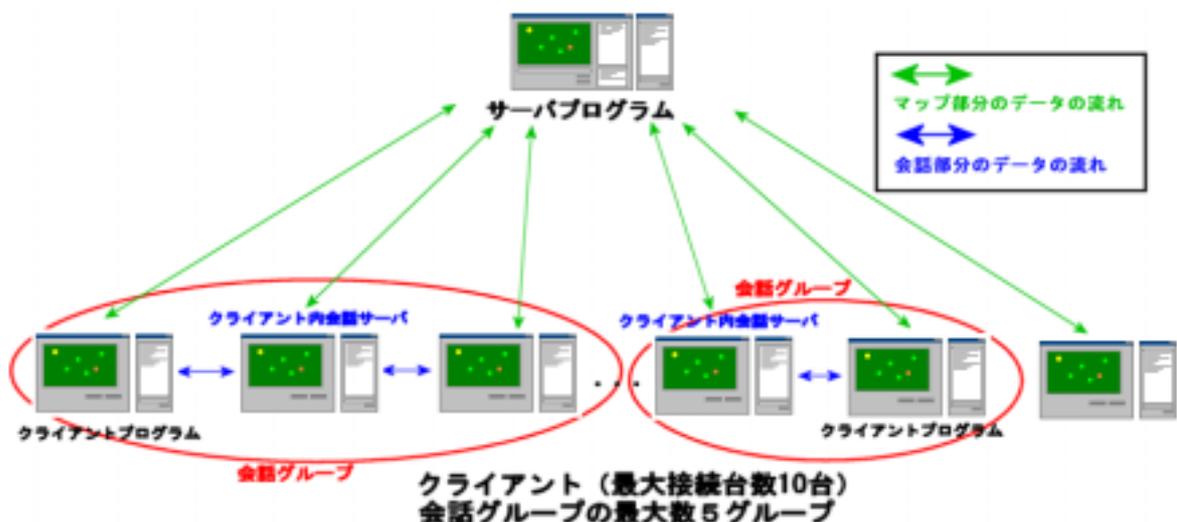


図3-6 グループ形成支援ソフトウェアの特徴・仕様

グループ形成サーバはクライアントの情報管理のみに集中させることにより負荷を軽減する。会話部分の処理についてはクライアントホスト上でプログラムの上に会話用のサーバを設定する。

会話用のサーバになったクライアントでも、自由に会話グループを抜け出すことが可能である。

実用を考えると、膨大なユーザ数の拡張が必要であるが、クライアントの**最大接続台数は当面10台、最大会話グループ形成数は5つ**に設定する。この接続台数はメモリ領域拡張により大きく拡充は可能である。

3-3-3. プログラムの各部名称とその機能・役割

グループ形成支援ソフトウェアもプログラムの構成は、動作確認版プログラムと同様、クライアント、サーバの2種のプログラムで構成されている。

グループ形成サーバのプログラムは、マップ部分のみで各クライアントの位置情報の処理や会話部分への接続の処理を行っている。クライアントプログラムを機能単位のプログラムで分割すると、マップ部分と会話サーバの部分、会話クライアントに分けることができる。

その構成は以下のようなものとなっている。

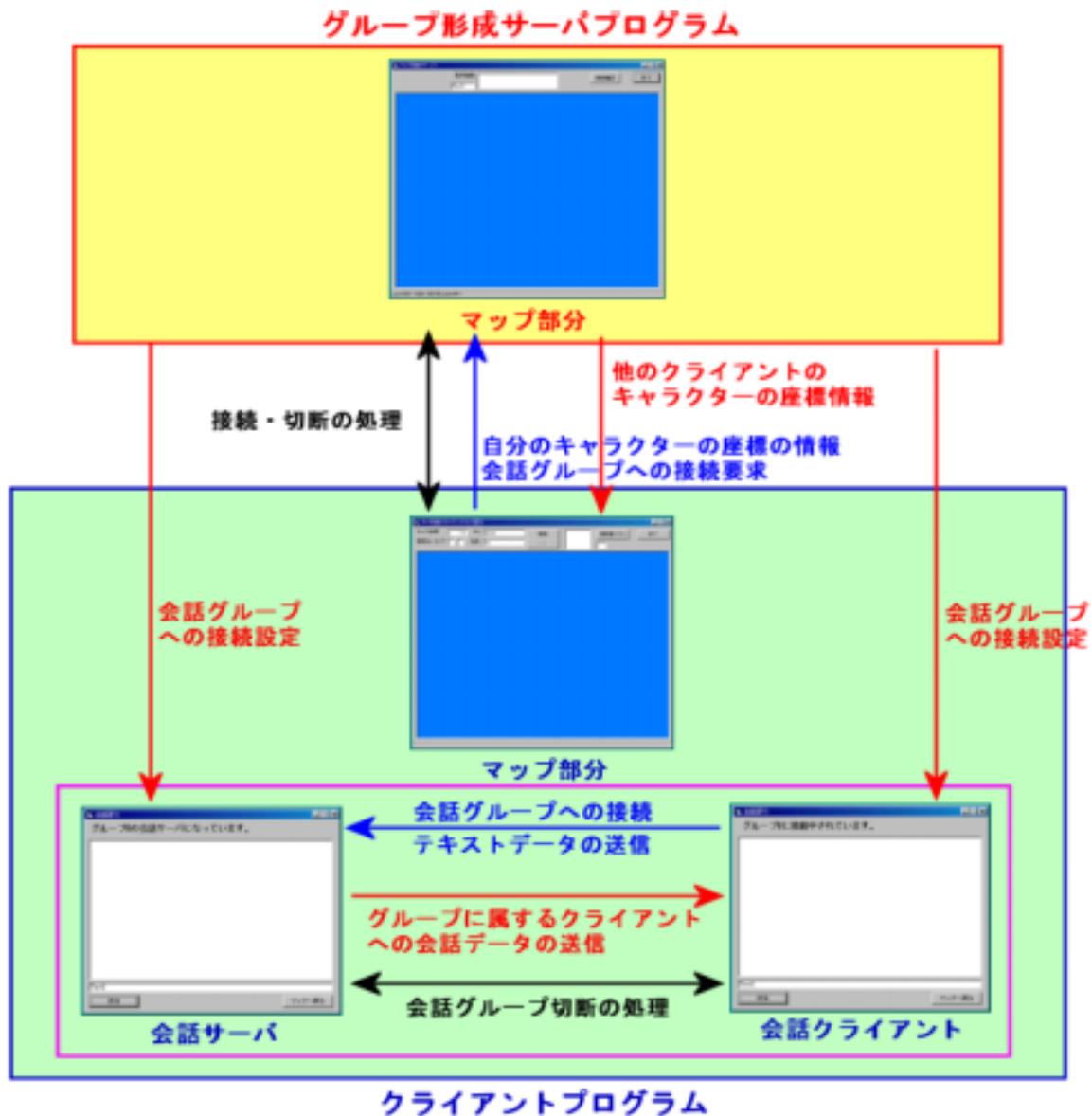


図3-7 改良版プログラムの各部名称とその機能・役割

グループ形成サーバのマップ部分では、各クライアントからの接続切断の処理、各クライアントから送られてくる座標の情報、会話部分への接続要求などの処理を行っている。

クライアントのマップ部分では、会話(チャット)をするためのメンバーを探しグループを形成する役割をしている。自分のキャラクターの位置情報を送ったり、他のクライアントの位置情報を受信しマップ部分に表示したり、会話の要求をサーバに送ったりなどの処理を行っている。

クライアントの会話サーバの部分では、クライアントのマップ部分で形成された会話グループの会話データをその会話グループに接続している全会話クライアントに対して送信をするという処理を行っている。

クライアントの会話クライアントの部分は、会話サーバに会話データを送信したり、他のクライアントから送られてきた会話データを受信したりする処理を行っている。

以上のような機能を持つグループ形成支援ソフトウェアを作成した。次章では、グループ形成支援ソフトウェアの全体の処理の流れについてソースコードを交え、プログラム部分についての説明を行う。

第4章 処理の流れとソースコード

本章では、ソフトウェアの処理の流れとその処理の流れに対応したソースコードの一部を示す。

4-1. プログラム製作環境

本研究では、ソフトウェアの作成には Microsoft Visual Basic 6.0 (サービスパック 5) を使用している。

本ソフトウェア作成するのに用いたコンピュータは Pentium III 850MHz、メモリ 512MB、オペレーティングシステムは Windows 98 である。本ソフトウェアの動作確認には、Pentium III 500MHz、メモリ 128MB、オペレーティングシステムは Windows 98 のコンピュータを 3 台用いた。

なお、本ソフトウェアは、Windows 98、Windows Me、Windows XP にて何ら問題なく動作することを確認済みである。

4-2. グループ形成支援ソフトウェアの処理の流れ

グループ形成支援ソフトウェアの処理の流れは以下のとおりである。まずはグループ形成サーバプログラム、クライアントプログラムのマップ部分の処理から会話グループが成立し、会話ウィンドウが開くまでの処理を示す。

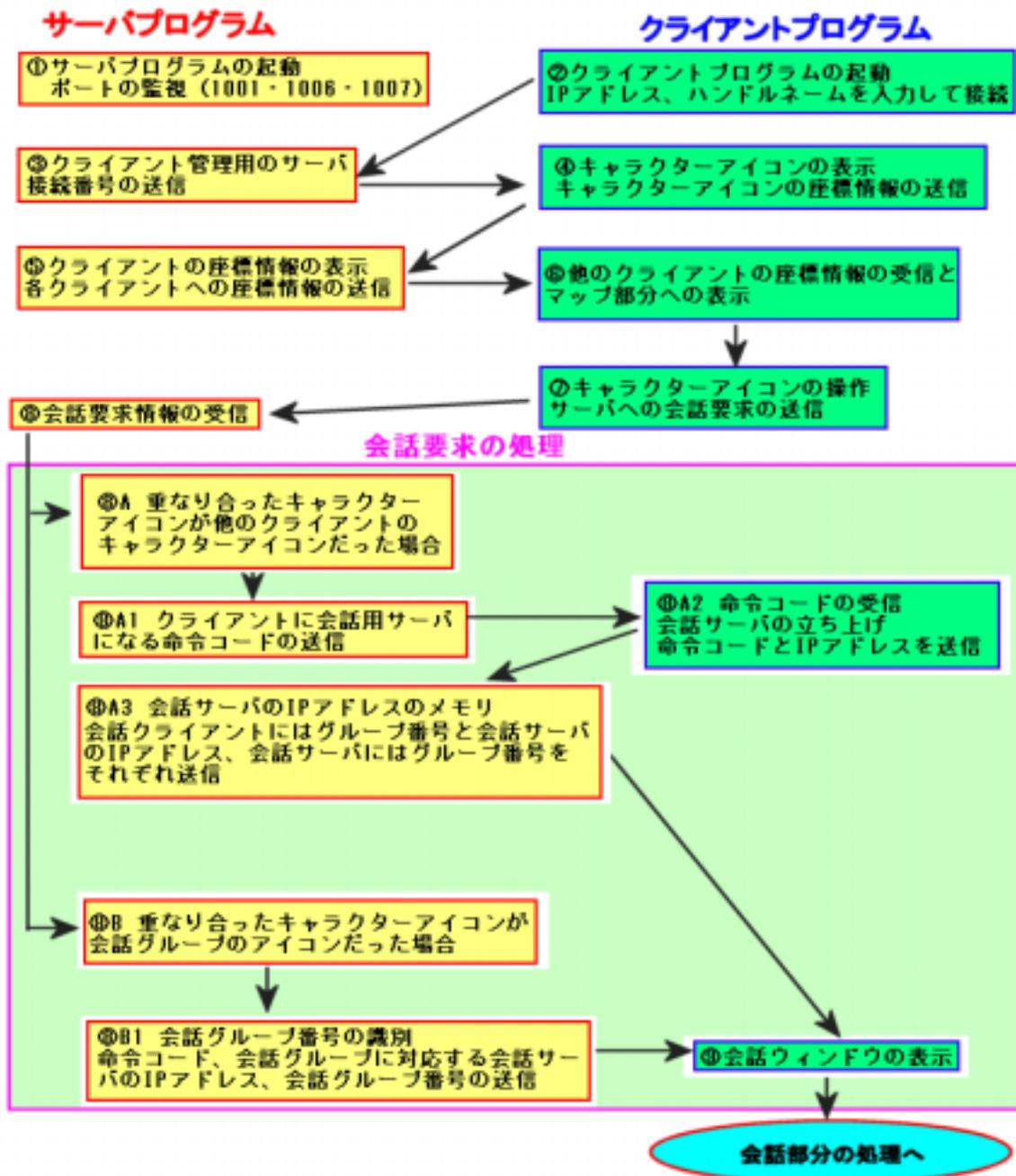


図 4-1 プログラムの起動から会話部分までの処理

次に、クライアントプログラムにて会話ウィンドウが開いた際の会話部分の処理の流れを示す。

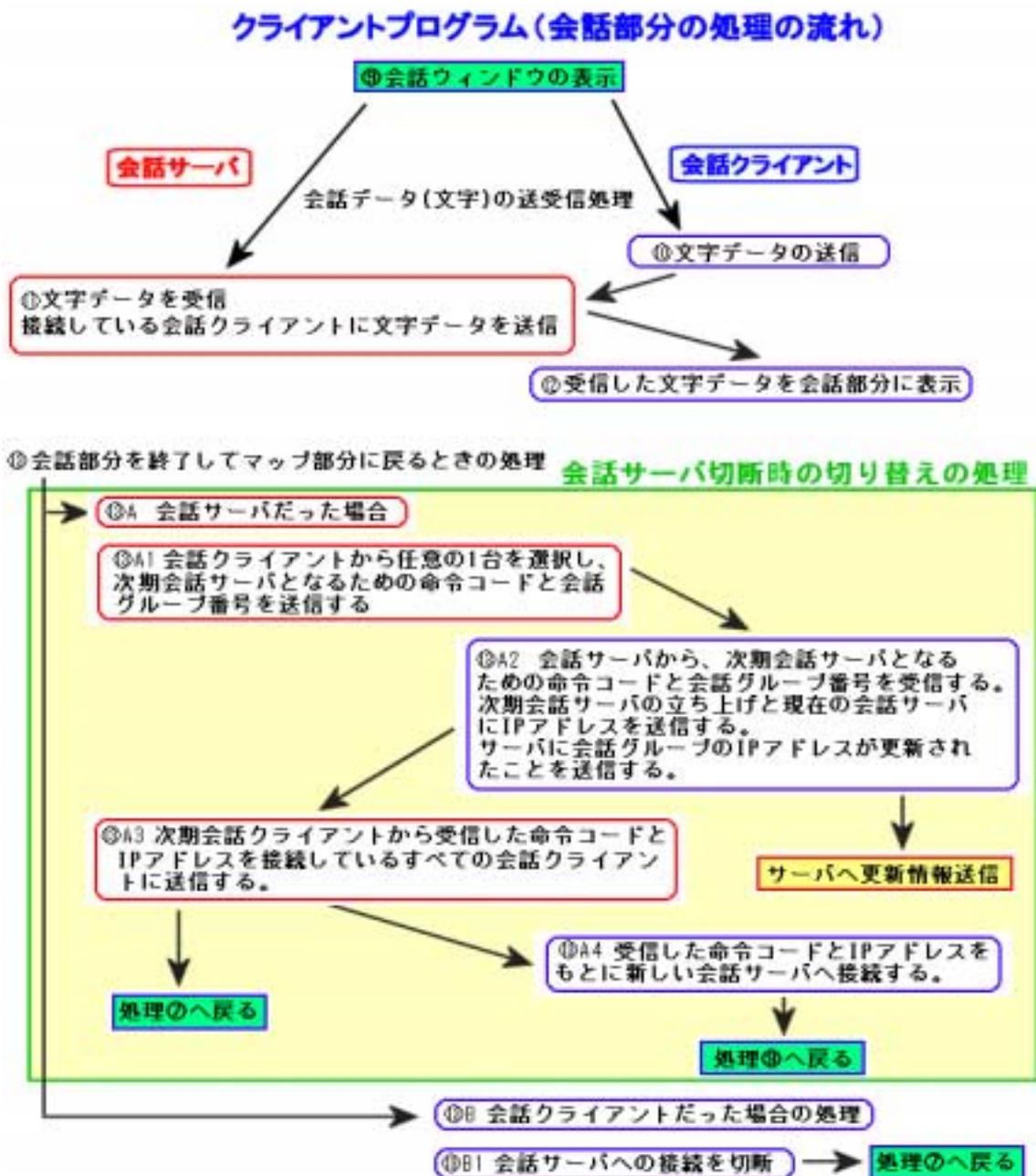


図 4-2 会話部分・会話ウィンドウの表示から会話部分の切断までの処理

接続からマップ部分の操作、表示などの処理

- ① グループ形成サーバプログラムが起動する。グループ形成サーバプログラムが起動するとポート 1001、1006、1007 の各ポートの監視を開始する。
- ② クライアントプログラムの起動。必要事項(サーバの IP またはコンピュータ名、ハンドルネーム)を入力した後、サーバに接続。
- ③ グループ形成サーバはクライアントからの要求を受け付け、クライアントの管理を行うため

の接続番号をクライアントに送信する。

- ④ クライアントは受け取った接続番号をもとにして、自分のキャラクターを表示。その位置座標をグループ形成サーバにリアルタイムで表示するため1秒間隔で送信する。
- ⑤ グループ形成サーバは受け取った位置座標データをグループ形成サーバのマップ部分に表示するとともに、受け取ったデータを現在接続しているすべてのクライアントに対して送信をする。
- ⑥ クライアントはグループ形成サーバから受け取った他のクライアントの位置座標データを処理しクライアントのマップ部分に表示する。

会話要求の送信から会話ウィンドウが立ち上がるまでの処理

- ⑦ クライアントは自分のキャラクターを操作し、他のクライアントと重なり合ったとき、会話の接続要求をクリックすると、クライアントから会話要求のデータがクライアントのポート 1006 より送信される。
- ⑧ グループ形成サーバは要求情報を処理し、要求に応じた命令コードをクライアントに対して送信する。
- ⑧A 重なり合ったグラフィックが他のクライアントのキャラクターだった場合。
- ⑧A1 グループ形成サーバは、接続要求をしたクライアントに対して会話用サーバになる命令コードを送信する。
- ⑧A2 クライアントは、受信した命令コードをもとにして、会話サーバの立ち上げを行うと同時にグループ形成サーバに対して自分の IP アドレスを送信する。
- ⑧A3 グループ形成サーバはクライアントから受信した IP を変数に記憶し、会話の相手方となる会話クライアントには命令コードと会話グループの番号と会話サーバの IP アドレスを送信し、会話サーバとなるクライアントに対しては、命令コードと会話グループの番号を送信する。
- ⑧B 重なり合ったグラフィックが会話グループの表示キャラクターだった場合。
- ⑧B1 グループ形成サーバは受信した要求情報を処理し、会話グループの番号を判断し、グループ形成サーバ内の変数に記録されている会話サーバの IP を要求があったクライアントに、命令コード、会話グループの番号、会話サーバの IP を送信する。
- ⑨ クライアントに会話ウィンドウが表示される。

会話ウィンドウが起動した後の会話部分の処理

この部分は複数話者によるグループ化が形成された状態から、会話クライアントの加入、退出などの処理、会話サーバの引継ぎなどを行う大変重要な部分である。

- ⑩ 会話クライアントより、会話(チャット)の文字データが送信される。
- ⑪ 会話サーバは、受信した文字データを処理し、会話ウィンドウ内に表示し、接続している全会話クライアントに対して、受信した文字データを送信する。
- ⑫ 会話クライアントは受信した文字データを会話ウィンドウ内に表示する。
- ⑬ 会話部分を終了してマップ部分に戻るときの処理。

- ⑬A 会話サーバだった場合の処理
 - ⑬A1 会話サーバは現在、会話サーバに接続されている会話クライアントの任意の1台を選択し、命令コードと会話グループ番号を送信。
 - ⑬A2 会話サーバから、命令コードと会話グループ番号を受信したクライアントは、命令コードと自分の IP アドレスを会話サーバに対して送信し、会話サーバの立ち上げを行う。この時、同時にグループ形成サーバに対して、会話グループの会話サーバの IP が更新されたという命令コードを送信する。
 - ⑬A3 命令コードを受信した会話サーバは、接続している全会話クライアントに対して命令コードと、新しい会話サーバの IP アドレスを送信して、マップ部分へと戻る(⑦の処理へ)。
 - ⑬A4 会話サーバから受信した、命令コードと新しい会話サーバの IP アドレスをもとにして新しい会話サーバへと接続を行う。(⑨の処理へ)
- ⑬B 会話クライアントだった場合の処理
 - ⑬B1 会話サーバへの接続を切断し、クライアントのマップ部分へと戻る。(⑦の処理へ)

4-3. グループ形成支援ソフトウェアのソースコード

4-3-1. グループ形成サーバ側プログラムのソースコード

グループ形成サーバ側のプログラムは2つの GUI、そして、2つのグループのプログラムモジュールで形成されている。



図 4-3 グループ化サーバ側プログラムの構成部品

処理の流れとソースコードの説明

ここでは、処理の流れで示した部分と対応したソースコードを抜粋して、そのソースコードの役割を説明する。

(i) Form1 モジュールのソースコード(抜粋)

(i-a) プログラムを起動時の処理

ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した①の部分に対応する。

Form1 をロードした時点でサーバプログラムは 1001、1006、1007 のポートを監視してクライアントプログラムからの接続要求を待機する。

```
Private Sub Form_Load()  
Winsock1.LocalPort = 1001  
Winsock2.LocalPort = 1006  
Winsock3.LocalPort = 1007  
StatusBar1.SimpleText = "port1001・1006・1007 を Listen 中…"  
Winsock1.Listen  
Winsock2.Listen  
Winsock3.Listen  
Load Form3  
  
End Sub
```

(i -b)クライアントの接続要求時の処理

クライアント接続時に行われる処理である。ここでは、クライアントからの接続の要求を受け付け、サーバへの接続番号をクライアントに対して送信する。ここで行われる処理はユーザグループ形成支援ソフトウェアの処理の流れで前記した ②の部分に対応する。

ポート 1001 への接続処理

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)  
For STct = WinST.LBound To WinST.UBound  
If WinST(STct).State = sckClosed Then  
WinST(STct).Accept requestID  
Text1.Text = STct  
WinST(STct).SendData "#" & STct  
StatusBar1.SimpleText = STct & "に接続がありました。"  
Exit Sub  
End If  
Next
```

(i -c)クライアントから受信した位置情報をマップ部分にキャラクターとして表示する処理

ここではクライアントから受信したキャラクターの座標情報の処理を行っている。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した ③の前半の部分に対応する。

```
Private Sub WinST_DataArrival(Index As Integer, ByVal bytesTotal As Long)  
'受信データ用の変数を定義します
```

```

Dim Strdata As String
Dim STRcut() As String '切断したデータ
Dim GCtoData As String 'クライアントからの要求状態を抜き取ったデータ。
Dim GCtoYoKyu As String 'クライアントからの要求コマンド。
Dim CutCont As Integer

```

'受信データの取りこみマップ座標

```

WinST(Index).GetData Strdata
GCtoData = Right(Strdata, Len(Strdata) - 1)
STRcut = Split(GCtoData, ":")
StoCSM GCtoData
For CutCont = 0 To 2
    Select Case CutCont
        Case 0
            GwSTno = CInt(STRcut(CutCont))
        Case 1
            SPX = CInt(STRcut(CutCont))
        Case 2
            SPY = CInt(STRcut(CutCont))
    End Select
Next

```

'受信データの表示

```

List1.AddItem SPX & "," & SPY & ":" & GwSTno
List1.ListIndex = (List1.ListCount - 1)
Image1(GwSTno).Left = (SPX - 1) * 32
Image1(GwSTno).Top = (SPY - 1) * 32
If Image1(GwSTno).Visible = False Then
    Image1(GwSTno).Visible = True
End If
End Sub

```

(i -d)クライアントからの要求に対する処理

この部分ではクライアントからの会話要求を受けての接続の処理やクライアント内の会話サーバからの会話グループのグラフィック表示の処理やグループ管理を行っている大変重要な部分である。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した A1、 A3、 B1、 A2の部分に対応する。

```

Private Sub WinYo_DataArrival(Index As Integer, ByVal bytesTotal As Long)
'受信データ処理用の変数
Dim StrDataY As String
Dim STRcutY() As String '切断したデータ
Dim GCStyY As String 'クライアントからの要求状態コマンド
Dim GCtoYoKyuY As String 'クライアントからの要求コマンド。
Dim CutContY As Integer
Dim GCYInt As Integer 'Image3 のグラフィックを削除、データ(数字)保存変数
'会話部分への処理の変数
Static KSIPMem(0 To 4) As String '会話用サーバの IP アドレス
'データの取り込み会話部分
Dim CliantNo As Integer '会話要求を送信したクライアントの wST 番号
Dim ATARISCSn As Integer '会話したい相手先の wST 番号
Dim Handol As String '会話要求を送信したクライアントのハンドルネーム(クライアント名)
Dim GArea As String '会話グループの指定
Dim IPMem As String '会話要求の会話サーバ IP の一時保管
Dim Box As String '送信データの一時格納
Dim IPMCon As Integer 'IPMem の空き検索番号
Dim IPNo As Integer '会話サーバの IP メモリー番号
Dim GN As Integer '会話グループ番号

```

```

WinYo(Index).GetData StrDataY
GCStyY = Left(StrDataY, 1)
GCtoYoKyuY = Right(StrDataY, Len(StrDataY) - 1)
Select Case GCStyY

```

クライアントから会話要求が送られてきたときの処理。受信した会話要求のデータをそれぞれの情報に分割する。

```

Case "?"
STRcutY = Split(GCtoYoKyuY, ":")
For CutContY = 0 To 3
Select Case CutContY
Case 0
CliantNo = CInt(STRcutY(CutContY))
Case 1
ATARISCSn = CInt(STRcutY(CutContY))
Case 2

```

```
GArea = STRcutY(CutContY)
```

```
Case 3
```

```
Handol = STRcutY(CutContY)
```

```
End Select
```

```
Next
```

上記で処理した情報をもとに、新しい会話グループを作製するのか、成立している会話グループへ接続しているのか判断するのが以下の処理である。判断によって、送信される命令コードが異なる。

```
If GArea = "P" Then
```

```
Box = "H" & ATARISCSn
```

```
WinYo(CliantNo).SendData Box
```

```
Else
```

```
GN = CInt(GArea)
```

```
Box = "N" & KSIPMem(GN) & ":" & GArea
```

```
WinYo(CliantNo).SendData Box
```

```
End If
```

新しい会話グループを形成する際、会話サーバとなるクライアントから送信されたクライアントの IP アドレス、接続させたいクライアントのサーバへの接続番号などを受信した場合の処理。

```
Case "O"
```

```
STRcutY = Split(GCtoYoKyuY, ":")
```

```
For CutContY = 0 To 2
```

```
Select Case CutContY
```

```
Case 0
```

```
CliantNo = CInt(STRcutY(CutContY))
```

```
Case 1
```

```
ATARISCSn = CInt(STRcutY(CutContY))
```

```
Case 2
```

```
IPMem = STRcutY(CutContY)
```

```
End Select
```

```
Next
```

```
Do While IPMCon <= 4
```

```
If KSIPMem(IPMCon) = "" Then
```

```
IPNo = IPMCon
```

```
KSIPMem(IPNo) = IPMem
```

```
Exit Do
```

```
End If
```

```

IPMCon = IPMCon + 1
Loop
WinYo(ATARISCSn).SendData "%" & IPNo & ":" & IPMem
WinYo(CliantNo).SendData "G" & IPNo

```

クライアント内に成立した会話サーバが途中で切断を行った場合の処理。切断を行った際、新しい会話サーバから送信される、会話サーバの IP アドレスメモリの更新を行う。

```

Case "U"
STRcutY = Split(GCtoYoKyuY, ":")
For CutContY = 0 To 3
  Select Case CutContY
    Case 0
      CGKSNo = CInt(STRcutY(CutContY))
    Case 1
      SKX = CInt(STRcutY(CutContY))
    Case 2
      SKY = CInt(STRcutY(CutContY))
    Case 3
      IPMem = STRcutY(CutContY)
  End Select
Next

```

```

KSIPMem(CGKSNo) = IPMem
Image2(CGKSNo).Visible = True
Image2(CGKSNo).Left = (SKX - 1) * 32
Image2(CGKSNo).Top = (SKY - 1) * 32

```

会話サーバが切断したときのグラフィックの処理

```

Case "Q"
GCYInt = CInt(GCtoYoKyuY)
SCcutM "Q" & GCYInt
Image2(GCYInt).Visible = False
Image2(GCYInt).Left = 640
Image2(GCYInt).Top = 480
KSIPMem(GCYInt) = ""

```

(ii)標準モジュールのソースコード(抜粋)

(ii -a)各クライアントへのキャラクターの座標情報の送信

ここでの処理はクライアントから送られてきたキャラクターの座標情報をサーバに接続している各クライアントに送信する処理を行っている。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した の後半部分に対応する。

```
Public Sub StoCSM(StoC As String)
Dim Ncont As Integer
With Form1.WinST
    For Ncont = .LBound To .UBound
        If .Item(Ncont).State = sckConnected Then
            .Item(Ncont).SendData "!" & StoC
            DoEvents
        End If
    Next
End With
End Sub
```

4-3-2.クライアント側プログラムのソースコード

グループ形成支援ソフトウェアのクライアント側プログラムは3つの GUI、そして、大きく分けると3つプログラムモジュールで形成されている。



図 4-4 クライアント側プログラムの構成部品

Form1 は主に各クライアントの位置情報の表示と会話部分への接続要求の処理などを行っている。

Form2は会話部分の処理を行い、会話サーバ、会話クライアントなどそれぞれの役割に分かれて会話の送信、受信などの処理をしている。

Form3 は現在接続しているクライアントのハンドルネームの表示を行っている。

(i)Form1 モジュールのソースコード(抜粋)

(i -a)サーバへの接続処理

クライアントがサーバへ接続するためのコンディションを整え接続を行う。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した の部分に対応する。

```
Private Sub Command1_Click()  
'先にオープンしている場合もあるのでクローズ  
Winsock1.Close  
Winsock3.Close  
Winsock4.Close  
'ホスト名のチェック  
If Text3.Text = "" Then  
    StatusBar1.SimpleText = "ホスト名を入力してください。"  
    Exit Sub  
End If  
'ホスト名の入力、あるいは IP アドレスの入力  
Winsock1.RemoteHost = Text2.Text  
Winsock3.RemoteHost = Text2.Text  
Winsock4.RemoteHost = Text2.Text  
'ポートの指定  
Winsock1.RemotePort = 1001  
Winsock3.RemotePort = 1006  
Winsock4.RemotePort = 1007  
'接続を開始します。  
StatusBar1.SimpleText = "接続中…"  
Winsock1.Connect  
Winsock3.Connect  
Winsock4.Connect  
LHN = Text3.Text  
Picture1.SetFocus  
End Sub
```

(i -b)サーバ接続後の受信データ処理

Winsock1 での処理は主に、サーバから受け取った接続番号をもとにしてマップ部分への他クライアントのキャラクター表示の処理を行っている。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した の前半、 の部分に対応する。

```

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim StrData As String '受信データ格納
Dim CutD As String 'サーバからの指示
'Dim DNuki As Integer '受信データ格納(数字のみ)
Dim DIti As String '受信データ格納(データ)
Dim STRcut() As String
Dim CutCont As Integer

Winsock1.GetData StrData
CutD = Left(StrData, 1)
DIti = Right(StrData, Len(StrData) - 1)
Select Case CutD
以下の処理は、サーバから送信された各クライアントのキャラクターアイコンの座標情報をマップ部分に表示するための処理である。
Case "!"
'List1.AddItem DIti
STRcut = Split(DIti, ":")
For CutCont = 0 To 2
Select Case CutCont
Case 0
SCSn = CInt(STRcut(CutCont))
Case 1
GPX = CInt(STRcut(CutCont))
Case 2
GPY = CInt(STRcut(CutCont))
End Select
Next
If Image1(SCSn).Visible = False And SCSn <> wSTno Then
Image1(SCSn).Visible = True
End If
Image1(SCSn).Left = (GPX - 1) * 32
Image1(SCSn).Top = (GPY - 1) * 32
Image1(wSTno).Left = 640
Image1(wSTno).Top = 480
ZahyoMem(SCSn) = GPX & "," & GPY
ZahyoMem(wSTno) = 20 & "," & 15
List1.AddItem GPX & "," & GPY & ":" & "," & SCSn

```

```
List1.ListIndex = (List1.ListCount - 1)
End Select
End Sub
```

(i -c) 自分のキャラクターの座標情報の送信

ある一定の時間が経過すると自動的に処理を行ってくれる Timer を使って自分のキャラクターの座標情報を送信する。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した の後半部分に対応する。

```
Private Sub Timer1_Timer()
'データの送信
ITIJOHO = "*" & wSTno & ":" & PX & ":" & PY & ":" & "E"
Winsock1.SendData ITIJOHO
JiZahy = PX & "," & PY
End Sub
```

(i -d) 自分のキャラクターの操作

自分のキャラクターは、キーボードの矢印キーを操作して移動することが可能である。ここには、それぞれのキーを押した場合の処理について書かれている。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した⑦の前半部分に対応する。

```
Private Sub Picture1_KeyDown(KeyCode As Integer, Shift As Integer)
Select Case KeyCode
Case vbKeyRight
Image2(wSTno).Left = Image2(wSTno).Left + 32
PX = PX + 1
If PX >= 20 Then
Image2(wSTno).Left = 608
PX = 20
End If
Case vbKeyLeft
Image2(wSTno).Left = Image2(wSTno).Left - 32
PX = PX - 1
If PX <= 1 Then
Image2(wSTno).Left = 0
PX = 1
End If
Case vbKeyUp
Image2(wSTno).Top = Image2(wSTno).Top - 32
```

```

    PY = PY - 1
    If PY <= 1 Then
        Image2(wSTno).Top = 0
        PY = 1
    End If

    Case vbKeyDown
        Image2(wSTno).Top = Image2(wSTno).Top + 32
        PY = PY + 1
        If PY >= 15 Then
            Image2(wSTno).Top = 448
            PY = 15
        End If
    End Select

    Text1.Text = PX & ", " & PY
End Sub

```

(i -e)キャラクター同士の重なるのチェック

Timer の処理を利用して自分のキャラクターが他のクライアントのグラフィックと重なりを逐一チェックしている。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した の中盤部分に対応する。

```

Private Sub Timer2_Timer()
    Dim Suzi As Integer
    Dim SuziG As Integer
    Dim No As Integer
    Dim MNo As Integer
    'グループのチェック
    Do While Suzi <= 9
        If JiZahy = ZahyoMem(Suzi) Then
            ATARISCSn = Suzi
            Exit Do
        End If
        Suzi = Suzi + 1
    Loop

    Do While SuziG <= 4
        If JiZahy = ZahyoKaiG(SuziG) Then
            ATARIGNo = SuziG

```

```
Exit Do
End If
SuziG = SuziG + 1
Loop
```

’当たりチェック

```
No = ATARISCSn
MNo = ATARIGNo
If JiZahy = ZahyoMem(No) Then
Command3.Enabled = True
Area = "P"
Text5.Text = Area
ElseIf JiZahy = ZahyoKaiG(MNo) Then
Command3.Enabled = True
Area = CStr(MNo)
Text5.Text = Area
Else
Command3.Enabled = False
End If
```

```
End Sub
```

(i -f) 会話要求の送信

(-e) の処理で他のクライアントと重なり合ったときに、会話部分への接続要求をクリックした時の処理である。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した の後半部分に対応する。

```
Private Sub Command3_Click()
Dim Ytemp As String
’会話部分への接続要求をサーバに出す。
Timer1.Enabled = False
MemPX = PX
MemPY = PY
Ytemp = "?" & wSTno & ":" & ATARISCSn & ":" & Area & ":" & LHN & ":" & "E"
Winsock3.SendData Ytemp
Image2(wSTno).Left = 640
Image2(wSTno).Top = 240
PX = 21
PY = 7
```

```
Timer1.Enabled = True
End Sub
```

(i -g)サーバからの会話要求に対する命令を受信したときの処理

ここでの処理は、サーバに会話要求を送信してサーバからの命令を受信したときの処理方法を記述している。この部分の処理は大変重要な役割をしている。

具体的には、会話サーバになる場合の処理、会話クライアントになるときの処理、クライアントが切断したときの他クライアントのキャラクターのグラフィック処理や会話サーバ切断時の会話グループのグラフィック処理などの処理について記述している。ここで行われる処理はグループ形成支援ソフトウェアの処理の流れで前記した A 2、 の部分に対応する。

```
Private Sub Winsock3_DataArrival(ByVal bytesTotal As Long)
```

```
Dim StrData3 As String '受信データ格納
```

```
Dim CutD3 As String 'サーバからの指示
```

```
Dim DNuki3 As Integer '受信データ格納(数字のみ)
```

```
Dim DIti3 As String '受信データ格納(データ)
```

```
Dim STRcut3() As String
```

```
Dim CutCont3 As Integer
```

```
Dim KSJyoHo As String
```

```
Dim IPAddKS As String '会話部分サーバの IP アドレス
```

```
Winsock3.GetData StrData3
```

```
CutD3 = Left(StrData3, 1)
```

```
DIti3 = Right(StrData3, Len(StrData3) - 1)
```

```
Select Case CutD3
```

以下の部分で会話サーバになるためのポート設定と監視を行い、サーバに対しては IP アドレスを送信し、会話サーバになるための情報を送信する。

```
Case "H"
```

```
'Timer1.Enabled = False
```

```
DNuki3 = CInt(DIti3)
```

```
Timer1.Interval = 2000
```

```
Form2.WKaiLis.Close
```

```
Form2.WKaiLis.LocalPort = 1002
```

```
Form2.WKaiLis.Listen
```

```
Form2.Show
```

```
Form2.SetFocus
```

```
KaiMOD = True
```

```

KSJyoHo = "O" & wSTno & ":" & DNuki3 & ":" & Winsock1.LocalIP & ":" & "E"
StatusBar1.SimpleText = "port1002 を Listen 中..."
Winsock3.SendData KSJyoHo

```

以下の処理はサーバから会話グループとして認証された番号を割り当てられ、その番号をもとにして会話グループの座標情報をサーバに送信する。

```

Case "G"
KGSno = CInt(DIti3)
Image3(KGSno).Left = (MemPX - 1) * 32
Image3(KGSno).Top = (MemPY - 1) * 32
Image3(KGSno).Visible = True
SKBox = "P" & KGSno & ":" & MemPX & ":" & MemPY & ":" & "E"
Winsock4.SendData SKBox
Timer1.Enabled = False
Timer4.Enabled = True
ITIJOHO = "*" & wSTno & ":" & PX & ":" & PY & ":" & "E"
Winsock1.SendData ITIJOHO
Form2.Label1.Caption = "グループ" & KGSno & "の会話サーバになっています。"

```

以下の処理は会話クライアントとなった場合の会話サーバへの接続の処理です。会話サーバへ接続するためのポートや IP アドレスの設定を行い会話サーバへと接続する。

```

Case "%"
STRcut3 = Split(DIti3, ":")
For CutCont3 = 0 To 1
    Select Case CutCont3
        Case 0
            GetArea = STRcut3(CutCont3)
        Case 1
            IPAddKS = STRcut3(CutCont3)
    End Select
Next
MemPX = PX
MemPY = PY
Form2.Winsock2.RemoteHost = IPAddKS
Form2.Winsock2.RemotePort = 1002
Form2.Winsock2.Connect
KaiMOD = False

```

```

StatusBar1.SimpleText = "会話サーバに接続中"
Image2(wSTno).Left = 640
Image2(wSTno).Top = 240
PX = 21
PY = 7
Timer1.Enabled = False
ITIJOHO = "*" & wSTno & ":" & PX & ":" & PY & ":" & "E"
Winsock1.SendData ITIJOHO
Form2.Label1.Caption = "グループ" & GetArea & "に接続されています。"

```

以下は既存の会話グループに接続する場合の処理です。サーバから送信された会話サーバの IP アドレスをもとにして接続を行う。

```

Case "N"
STRcut3 = Split(DIti3, ":")
For CutCont3 = 0 To 1
  Select Case CutCont3
    Case 0
      IPAddKS = STRcut3(CutCont3)
    Case 1
      GetArea = STRcut3(CutCont3)
  End Select
Next
Form2.Winsock2.RemoteHost = IPAddKS
Form2.Winsock2.RemotePort = 1002
Form2.Winsock2.Connect
KaiMOD = False
StatusBar1.SimpleText = "会話サーバに接続中"
Timer1.Enabled = False
ITIJOHO = "*" & wSTno & ":" & PX & ":" & PY & ":" & "E"
Winsock1.SendData ITIJOHO
Form2.Label1.Caption = "グループ" & GetArea & "に接続されています。"

```

'他クライアントが切断したときのグラフィック、座標の処理

```

Case "X"
DNuki3 = CInt(DIti3)
Image1(DNuki3).Visible = False
Image1(DNuki3).Top = 640

```

```
Image1(DNuki3).Left = 480
Form3.Text1(DNuki3) = ""
```

```
'会話グループのグラフィックの消去
Case "Q"
```

```
DNuki3 = CInt(DIti3)
Image3(DNuki3).Visible = False
Image3(DNuki3).Left = 640
Image3(DNuki3).Top = 480
End Select
End Sub
```

(ii) Form2 のソースコード(抜粋)

クライアントプログラムの Form2は処理の流れの⑨以降の処理を主に行っている。

(ii-a) 会話送信の処理

ここでは、処理の流れ⑩に対応するテキストデータを送信するための処理を行っている。

```
Private Sub Command1_Click()
Dim KAIWABox As String
```

以下の処理は会話サーバがテキストデータを送信する場合の処理である。

```
If KaiMOD = True Then
    KAIWABox = "T" & Form1.Text3.Text & ":" & Text2.Text
    Form2.List1.AddItem Form1.Text3.Text & ":" & Text2.Text
    List1.ListIndex = (List1.ListCount - 1)
    '全クライアントへデータの送信
    SKaiwaC KAIWABox
    SHBufK(KaisuK) = LHN & ":" & Text2.Text
        KaisuK = KaisuK + 1
        If KaisuK = 6 Then
            KaisuK = 0
        End If
    Text2.Text = ""
    Text2.SetFocus
```

以下の処理は会話クライアントがテキストデータを送信する処理である。

```
ElseIf KaiMOD = False Then
'Form1.Timer1.Enabled = False
```

```

KAIWABox = "T" & LHN & ":" & Text2.Text
Winsock2.SendData KAIWABox
Text2.Text = ""
Text2.SetFocus
End If
End Sub

```

(ii -b) 会話サーバのデータの受信処理

ここでは、会話データを会話サーバの会話表示部分に表示したり、会話サーバが切断する場合、現在接続している会話クライアントへ新しい会話サーバへの接続情報の申し送りをしたりしている。ここでの処理は処理の流れで示した⑪の前半部分、⑬A3の部分の処理に対応する。なおこの部分は、会話サーバ切り替え機能における重要な役割を果たしている部分である。

```

Private Sub WcK_DataArrival(Index As Integer, ByVal bytesTotal As Long)
'会話のバッファ
Dim GDataK As String 'データ格納用
Dim GDMeireiK As String 'クライアントからの要求
Dim GDataKaiwaK As String 'クライアントからの要求を除いた会話データ
Dim KClName As String '接続しているクライアント名
Dim WcKMem As Integer 'WcKの番号一時保存
Dim GCK() As String 'データ分割用
Dim GCcK As Integer 'データ分割用カウンタ
Dim W As Integer

```

```

WcK(Index).GetData GDataK

```

```

GDMeireiK = Left(GDataK, 1)

```

```

Select Case GDMeireiK

```

以下はテキストデータを受信したときの処理である。受信したテキストデータを表示するだけでなく、接続台数が3台以上になったときバッファに蓄えたテキストデータを送信するために、バッファに受信したテキストデータを蓄えている。

```

Case "T"

```

```

GDataKaiwaK = Right(GDataK, Len(GDataK) - 1)

```

```

SHBufK(KaisuK) = GDataKaiwaK

```

```

KaisuK = KaisuK + 1

```

```

If KaisuK = 6 Then

```

```

KaisuK = 0

```

```

End If

```

```

List1.AddItem GDataKaiwaK

```

```
List1.ListIndex = (List1.ListCount - 1)
```

```
'データを渡す。
```

```
SKaiwaC GDataK
```

以下は切断の処理時に新しく会話サーバになる会話クライアントから送信される接続グループの番号、新しい会話サーバの IP アドレスを現在接続している会話クライアントに送信するための処理である。

```
Case "F"
```

```
With Form2.WcK
```

```
    For W = .LBound To .UBound
```

```
        If .Item(W).State = sckConnected Then
```

```
            .Item(W).SendData GDataK
```

```
            DoEvents
```

```
        End If
```

```
    Next
```

```
KaiMOD = False
```

```
Form1.Visible = True
```

```
PX = MemPX
```

```
PY = MemPY
```

```
Form1.Image2(wSTno).Left = (PX - 1) * 32
```

```
Form1.Image2(wSTno).Top = (PY - 1) * 32
```

```
Form1.Picture1.SetFocus
```

```
Form1.Timer1.Interval = 1000
```

```
Form1.Timer1.Enabled = True
```

```
Form1.Timer4.Enabled = False
```

```
Unload Me
```

```
End With
```

```
End Select
```

```
End Sub
```

(ii -c) 会話クライアントのデータ受信処理

ここでの処理は会話クライアントがデータを受信したときの処理である。処理の流れ⑫の会話(チャット)のテキストデータを表示する処理、テキストデータのバッファを受信したときの処理、処理の流れ⑬A2 の会話サーバ切断時の新しい会話サーバに選択されたときの処理、新しい会話サーバへの接続の処理などを行っている部分であり本プログラム最重要の部分の一つである。

```
Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)
```

```
Dim StrHana As String 'データ格納用
Dim SMEirei As String 'サーバからの命令
Dim Hanasi As String '命令を抜き取ったデータ用
Dim IPAddNext As String '会話サーバが突然切断したときの次の接続 IP
```

```
Winsock2.GetData StrHana
```

```
SMeirei = Left(StrHana, 1)
```

```
Hanasi = Right(StrHana, Len(StrHana) - 1)
```

```
Select Case SMeirei
```

以下は会話のテキストデータが会話サーバから受信した場合の会話データ表示部分への処理である。

```
Case "T"
```

```
Form2.List1.AddItem Hanasi
```

```
List1.ListIndex = (List1.ListCount - 1)
```

以下はクライアントが3台以上接続した場合、会話サーバから送信されるテキストデータのバッファを受信した場合の会話データ表示部分への処理である。

```
Case "@"
```

```
Dim BfCut() As String
```

```
Dim Bfc As Integer
```

```
Do While Bfc <= 5
```

```
BfCut = Split(Hanasi, "@")
```

```
If BfCut(Bfc) = "" Then
```

```
Exit Do
```

```
Else
```

```
Form2.List1.AddItem BfCut(Bfc)
```

```
End If
```

```
Bfc = Bfc + 1
```

```
Loop
```

```
'会話サーバよりの強制切断
```

```
Case "S"
```

```
Winsock2.Close
```

```
'Form1.Timer1.Enabled = False
```

```
Form1.Visible = True
```

```
PX = MemPX
```

```
PY = MemPY
```

```
Form1.Image2(wSTno).Left = (PX - 1) * 32
```

```
Form1.Image2(wSTno).Top = (PY - 1) * 32
```

```
Form1.Picture1.SetFocus
```

```
Form1.Timer1.Interval = 1000
```

```
Form1.Timer1.Enabled = True
```

```
Form1.Timer4.Enabled = False
```

```
Unload Me
```

以下は、新しい会話サーバに選ばれた場合の処理である。この時点でこの会話クライアントは会話サーバとしてのネットワーク設定を行うとともに、サーバへ会話サーバの更新の情報送信を行う。

```
Case "R"
```

```
KGSno = CInt(Hanasi)
```

```
Winsock2.SendData "F" & Form1.Winsock1.LocalIP
```

```
Form1.Timer1.Interval = 2000
```

```
Form2.WKaiLis.Close
```

```
Form2.WKaiLis.LocalPort = 0
```

```
Form2.WKaiLis.LocalPort = 1002
```

```
Form2.WKaiLis.Listen
```

```
Form2.Label1.Caption = "グループ" & KGSno & "の会話サーバになっています。"
```

```
Form2.SetFocus
```

```
KaiMOD = True
```

```
Form1.StatusBar1.SimpleText = "port1002 を Listen 中…"
```

```
SKBox = "U" & KGSno & ":" & MemPX & ":" & MemPY & ":" & Form1.Winsock1.LocalIP & ":" & "E"
```

```
Form1.Winsock3.SendData SKBox
```

```
Form1.Timer1.Enabled = False
```

```
Form1.Timer4.Enabled = True
```

```
Case "F"
```

```
If KaiMOD = True Then
```

```
Winsock2.Close
```

```
ElseIf KaiMOD = False Then
```

```
IPAddNext = Hanasi
```

```
Winsock2.Close
```

```
Winsock2.RemoteHost = IPAddNext
```

```
Winsock2.RemotePort = 1002
```

```
Winsock2.Connect
```

```
KaiMOD = False
```

```
End If
End Select
End Sub
```

(ii-d) 会話部分を終了する場合の処理

ここでの処理は、会話サーバ、会話クライアントによって処理の方法が全く違う。処理の流れ⑬B⑬B1 に示すように、会話クライアントの場合は会話サーバとの切断をして、マップ部分にクライアントのキャラクターアイコンを表示する。会話サーバの場合は、まず会話クライアントの接続台数によって処理の方法が変化する。接続している会話クライアントが 1 台の場合は、接続しているクライアントを強制的に切断するための命令コードを送って切断する。接続している会話クライアントが 2 台以上の場合、処理の流れ⑬A1 に示すように、接続している会話クライアントの任意の 1 台を選択し、その会話クライアントを新しい会話サーバとするように命令コードを送信する処理を行う。ちなみに、この部分も本プログラム最重要の部分の一つである。

```
Private Sub Command2_Click()
```

```
Dim Co As Integer
```

```
Dim GikiS As Integer
```

```
Dim D As Integer
```

以下は会話クライアントの場合の処理である。

```
If KaiMOD = False Then
```

```
Winsock2.Close
```

```
'Form1.Timer1.Enabled = False
```

```
Form1.Visible = True
```

```
PX = MemPX
```

```
PY = MemPY
```

```
Form1.Image2(wSTno).Left = (PX - 1) * 32
```

```
Form1.Image2(wSTno).Top = (PY - 1) * 32
```

```
Form1.Picture1.SetFocus
```

```
Form1.Timer1.Interval = 1000
```

```
Form1.Timer1.Enabled = True
```

```
Form1.Timer4.Enabled = False
```

```
Unload Me
```

以下は、会話サーバであるときの切断処理で、会話クライアント接続台数 1 台の場合のときの処理である。

```
Elseif KaiMOD = True Then
```

```
If KCon = 1 Then
```

```
'サーバに会話グループのグラフィック処理をさせる命令
```

```
Form1.Winsock3.SendData "Q" & KGSno
```

'会話サーバに接続しているすべてのクライアントに切断命令

```
SKaiwaC "S"  
KCon = 0  
Form1.Visible = True  
PX = MemPX  
PY = MemPY  
Form1.Image2(wSTno).Left = (PX - 1) * 32  
Form1.Image2(wSTno).Top = (PY - 1) * 32  
Form1.Picture1.SetFocus  
Form1.Timer1.Interval = 1000  
Form1.Timer1.Enabled = True  
Form1.Timer4.Enabled = False  
For D = 0 To 5  
SHBufK(D) = ""  
Next D  
Unload Me
```

以下に示すのは会話サーバ切断時に会話クライアントが2台以上接続している場合の処理である。

```
ElseIf KCon >= 2 Then  
Co = 0  
With Form2.WcK  
Do While Co <= 4  
If .Item(Co).State = sckConnected Then  
GikiS = Co  
Exit Do  
End If  
Co = Co + 1  
Loop  
End With  
WcK(GikiS).SendData "R" & KGSno  
For D = 0 To 5  
SHBufK(D) = ""  
Next D  
End If  
End If  
End Sub
```

(iii) 標準モジュールのソースコード(抜粋)

以下の処理は、会話サーバに受信されたデータを各会話クライアントに送信するための処理である。ここでの処理は処理の流れの⑬A1の後半部分に相当する。

```
Public Sub SKaiwaC(SCKaiwaD As String)
Dim Ncont As Integer
With Form2.WcK
    For Ncont = .LBound To .UBound
        If .Item(Ncont).State = sckConnected Then
            .Item(Ncont).SendData SCKaiwaD
            DoEvents
        End If
    Next
End With
```

以上で主要部分のプログラム説明を終了する。

4-4. 考察

4-4-1. ソフトウェアの特化すべき機能

本研究で開発を目指したユーザグループ形成支援ソフトウェアには自由にユーザが作業グループをできるという特徴があり、その有用性を確認するため初歩的なプログラムで原理確認をした。本研究で作製したプログラムには以下のような機能が盛り込まれている。

(1) 会話グループを形成する際、グラフィックを利用して行えるので煩わしい文字列を検索することなくゲーム感覚で会話メンバーを構成できる。

グラフィックを多用したことで誰でも直感的に利用が可能となった。動作確認版ソフトウェアの動作チェックでもすべてのユーザが操作に迷うことなく使用できていた。

(2) ネットワークを意識させることなく最低限の情報を入力するだけで、プログラムがネットワークの処理を自動的に行う。

ネットワークを利用したもの＝設定が難しい、専門の知識が必要、という考えを持つ人が多いと考え、ユーザが行うネットワークの設定は必要最小限に抑え、アプリケーションを統括するサーバ、会話サーバの切り替え機能も含めてほとんどの処理を自動的にプログラム内で行うという方式を採ったことにより、ユーザはネットワークをさほど意識することなくソフトウェアを使用することができた。

4-4-2. 他のソフトウェアとの比較

現在、盛んに学会などで取りざたされているマルチキャスト配信木構造を利用した動画配信プログラム[11]との比較を行った。マルチキャスト配信木構造を利用した動画配信プログラムの特徴は、グループ通信主体のため、特定のサーバを必要としないことである。ネットワーク資源の有効率の高いマルチキャストは有用な通信技術の一つだと考えられている。本研究で作成したユーザグループ形成支援ソフトウェアとの相違点は、ネットワーク構成部に違いがある。以下の図に、その比較を示す。

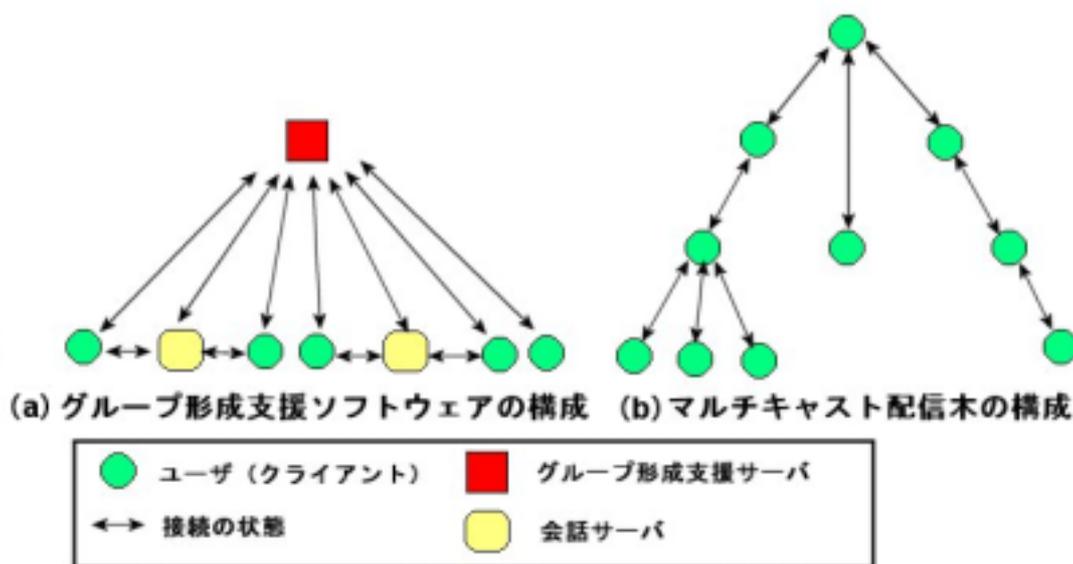


図 4-5 マルチキャスト配信木・グループ形成支援ソフトウェアの構成図

(a)のグループ形成支援ソフトウェアでは、すべてのクライアントを統括するためのグループ形成サーバとクライアントが形成したグループを統括する会話サーバが存在するが、(b)のマルチキャスト配信木の構成ではサーバは存在せず、ユーザ(クライアント)同士が階層的に接続するという形式となっている。各々の利点・課題を以下に述べる。

グループ形成支援ソフトウェアの構成の利点・課題

サーバを利用しているのでクライアントの管理が容易であるが、その反面、サーバがクライアントの接続数を制限してしまう。2階層のネットワーク構成であり、サーバが、クライアントを管理しているため、切断、接続の処理が比較的簡単である。

マルチキャストの配信木の構成の利点・課題

サーバが存在しないので、ユーザが自由にグループを形成できるが、グループ全体を把握するのが容易ではない。階層的にユーザを接続しているため、ユーザが切断したときに配信木の構成を更新しなくてはならず、処理が非常に複雑になる。

表 4-1 グループ形成支援ソフトウェアとマルチキャスト配信木構成との特徴比較

	利点	欠点
ユーザグループ形成支援ソフトウェア	クライアントの管理が容易。 接続・切断の処理が容易。	サーバによるクライアント数の制限。
マルチキャスト配信木構成	サーバが不必要。 ユーザの自由なグループ形成。	ユーザ切断時の配信木再構成の処理が複雑。

以上のような特徴により、本研究のソフトウェア構成は、ネットワーク内のクライアントを完全に把握しなければならない場合などに有効であると考えられる。

4-4-3. ソフトウェアの改善すべき点

(1) 動作確認版ソフトウェアの動作チェック時に行ったアンケートをもとに、ソフトウェアの改善を行う。

具体的には、クライアントのハンドルネームや会話グループに誰が属しているか、など各クライアントの情報の表示、マップ部分やキャラクターアイコンなどのグラフィックの改善を行う。

～参考資料～

- [1]北山洋幸 著 「はじめての Visual Basic 6.0 インターネットプログラミング」
- [2]谷尻豊寿 監修 谷尻かおり 著 「はじめての Visual Basic 6」
- [3]谷尻豊寿 監修 谷尻かおり 著 「Visual Basic 6 レベル 2 プログラミング」
- [4]笠原和弘・山本美孝 著 「はじめての Visual Basic 6.0 入門(活用編)」
- [5]玉川理英 著 「入門 Visual Basic 6.0」
- [6]松田猛・小高郁 著 「Visual Basic 6.0 300 の技 ネットワーク+データベース編」
- [7]井上俊宏 著 「Visual Basic 6.0 の応用 50 例」
- [8]北山洋幸 著 「はじめての Visual Basic 6.0 グラフィックス&ゲームプログラミング」
- [9]野田晃 著 「Visual Basic 6.0 ではじめる Windows98 プログラミング」
- [10]池田京子 増田智明 木村祐樹 プロジェクト A 著 「VisualBasic 逆引き大全 500 の意」
- [11]「情報処理学会研究報告」Vol.2002 No.108 2002-DPS-110 pp.7-12 中村嘉隆
廣森聡仁 山口弘純 安本慶一 東野輝夫
- [12]「情報処理学会研究報告」Vol.2002 No.54 2002-DPS-108 pp.55-60 森田悟史
古川雄宣 山田和宏 佐藤文明
- [13]「情報処理学会研究報告」Vol.2002 No.32 2002-DPS-107 pp.43-48 山口弘純
中村嘉隆 廣森聡仁 安本慶一 東野輝夫 谷口健一
- [14]「情報処理学会研究報告」Vol.2002 No.32 2002-DPS-107 pp.97-102 大場正弘
中沢実 服部進実

第5章 まとめ

5-1. 本ソフトウェアの作成、検証から得られた知見

本研究のユーザグループ形成支援ソフトウェアのプログラムの作成、検証により得られた知見を列挙する。

- ① 会話グループを形成する際、グラフィックを利用することにより、ユーザが直感的に利用できる環境を提供できた。
- ② ユーザが行うネットワーク設定を最小限にし、後はプログラム内部で自動的にネットワークの設定や接続を行えるようにし、ユーザにネットワークを意識させないようにすることができた。
- ③ ユーザグループ形成、解消時に(会話)サーバの切り替える機能は、よりフレキシブルなユーザ(会話)グループの形成を可能にした。

などの有効性が明らかとなった。さらに本機能を利用することによる発展性を次節で述べる。

5-2. 発展性・応用の可能性

本研究では搭載するアプリケーションとして、テキスト情報を共有するチャットのみ機能としてソフトウェアを作製したが、テキストベースの情報を送ることができたならば、コンピュータ内にあるデータならストリーム、またはテキスト形式データに変換することにより送受信、共有が可能ということになり、動画、音声、データなどあらゆるデータのグループ間の送受信、共有が可能であると思われる。

本プログラムのコードは、Visual Basic 6.0 でほとんどシリアル系列の処理で組んでおり、冗長ではあるが、改良の余地を残すため機能拡張を容易に行えるように配慮してある。

本ソフトウェアで実現した、会話用サーバを自由に切断、接続できるという会話サーバの切り替え処理の概念は、オンラインゲームにおけるネットワークの処理に利用できると考えている。

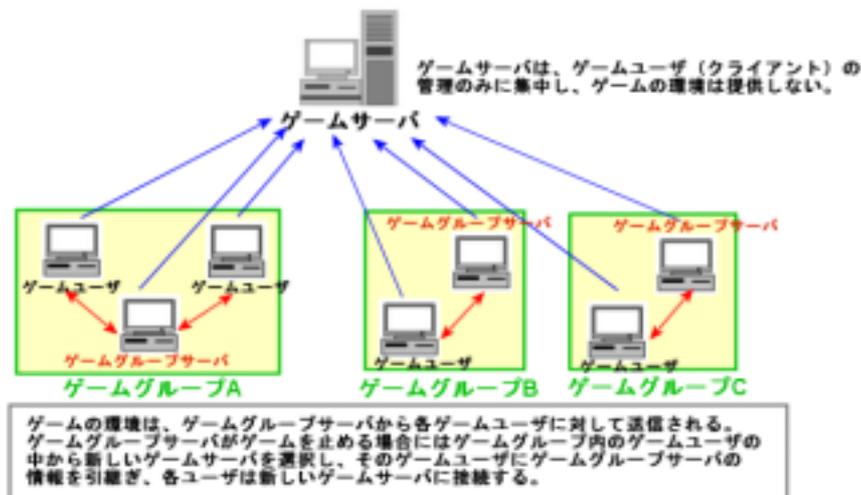


図 5-1 オンラインゲームへの応用例

他の利用法として、大量のデータ流入によるサーバに障害が発生しそうな場合に他の余力のあるサーバに処理を分担する場合や、サーバがダウンしてしまった場合の代理のサーバへのネットワークの切り替えなど、ネットワーク分散型処理の分野でも利用できると思われる。

5-3. ユーザがネットワークを利用したソフトウェアに求めるもの

これは、動作確認版ソフトウェアの動作チェック時のアンケートの傾向から推察するユーザの意識を、以下に列挙する。ネットワークを利用したソフトウェアにはユーザの立場からこのような要素が求められているようである。

1. グラフィックのデザイン性や個性、ユニークさ。
2. リアルタイム性。
3. ゲーム的な要素。
4. ネットワークを意識しなくていいもの。

などである。

本研究の結論として、ネットワークを利用したソフトウェアを作製する上で重要な部分は、ユーザにネットワークを利用しているという意識をさせないようなプラットフォームを作り、専門的な知識がない人でも簡単に使えるということと、グラフィックを利用し、誰でも簡単に扱えるような工夫をすることである。そして、普及を考えた場合、当然の条件として、機能や性能も然る事ながら、ゲーム的な遊びの要素をも取り込まなければユーザに受け入れてもらえないということである。

謝辞

最後に、熱心にご指導してくださった野中弘二先生。本当にありがとうございます。それから、いつもわがままばかりですみませんでした。研究にアドバイスをくれた浜田正典君、本当にありがとうございます。アドバイスは大変役に立ちましたし、大変勉強になりました。そして、寺田広之君、プログラムのアイデアや意見を出してくださってありがとうございました。それから、福永将君、プログラムの動作チェックの手伝いや、プログラムに対していろいろな意見をありがとうございました。本当に助かりました。

この研究で一番感謝しなければならないのは、プログラムの動作チェックに協力してくれた要約筆記サークルのみなさんです。本当にありがとうございます。要約筆記サークルのみなさんの協力のおかげで、いいプログラムを作ることができました。私自身、動作チェックをする前は、こんなプログラムを人前に出しているものか？これが研究になるのか？といつも不安に思っていました。しかし、動作チェックの時に、みなさんが楽しそうにプログラムを使っている姿を見て、内心ほっとしましたし、自分の研究に少し自信が持てるようになりました。みなさんが書いてくれたアンケートの意見は、無駄にしないようにソフトウェアの方に反映させています。

以上の方々には、大変お世話になり感謝しております。この紙面を借りてお礼を申し上げます。本当にありがとうございました。