

インターネット上でカードゲームを行うための汎用プロトコル群の開発

小 出 雅 人[†] 程 京 徳[†]

われわれはインターネット上でカードゲームを行うための汎用プロトコル群を開発した。このプロトコル群はカードゲームをサーバ・クライアントモデルで実現するための、カードの表現、プレイヤー間の通信、ルールの記述の方法などを規定したものである。またこの汎用プロトコル群に基づき、インターネット上で行うカードゲームプログラム開発のためのツールキットを設計、実装した。このツールキットを用いることにより、インターネット上で行うさまざまなカードゲームのプログラムを簡単に開発することができる。本論文ではこの汎用プロトコル群およびツールキットについて述べる。またツールキットを用いたプログラミングの例として、開発したカードゲームプログラムから一つを紹介する。

General-purpose Protocols for Playing Card Games on the Internet

MASATO KOIDE and JINGDE CHENG

We propose some general-purpose protocols which are necessary for playing card games on the Internet. The protocols govern the expressions of cards, the interaction of players, servers, and other components within a card game playing on the Internet. Based on the protocols, we have designed and implemented a toolkit for programming online card games. Using the toolkit, a card game programmer can easily implement various online card game programs. We show some examples of programming online card games using our toolkit.

1. はじめに

近年、i-mode のアプリケーションやネットワークゲームなど、インターネットを利用して行うゲーム¹⁾の利用者が増しており、それに伴い各種ゲームサーバの需要が高まってきている。また地理的、時間的制限のないインターネットの特徴を利用し、さまざまなゲームの大会がインターネット上で行われており、そのためのプロトコルもそれぞれのゲームについて開発されている²⁾。

複数のゲームに対応できるシステムやプロトコルを開発することで、ゲームプログラム開発の負担を軽減し、またプログラムの再利用を促進することが期待できる。例えば M.Buro らが設計・開発したゲームサーバ³⁾は、盤を用い、かつ盤面の初期状態が対称形であるさまざまな二人用ゲームを扱うもので、このようなゲームのためのサーバやクライアントソフトウェアを簡単に作成することができる。このようなシステムには、多種類のゲームに対応できるという機能に加え、利用者にとって見やすく扱いやすい GUI (Graphical User Interface) や、インターネット上でゲームの公

式大会を行う場合を想定したセキュリティ機能が求められる。

そこでわれわれは、多人数ゲームの代表ともいえるカードゲーム (ポーカー、ブリッジなどのトランプゲーム) について、ゲームのルールの記述やプレイヤー同士の通信を行うための汎用プロトコル群を開発した。また汎用プロトコル群に基づき、インターネット上で行うカードゲームプログラム開発のためのツールキットを設計、実装した。本論文ではこの汎用プロトコル群およびツールキットについて述べる。なお本論文内では「プロトコル」という言葉を英語的な意味で用いている。つまり、通信のみならず、あらゆる規約を意味するものであり、かつコンピュータ間の利用に限定されるものではない。

2. カードゲームに関する定義

2.1 カードゲームの定義

インターネット上で行うことを前提として、本論文ではカードゲームを以下のように定義する。

- カード (日本語でいうトランプ) を使用するゲームである。
- ゲームには 2 人以上のプレイヤーが存在するものとする。1 人用の遊びはインターネット上で行う必要がないため考慮しない。

[†] 埼玉大学大学院理工学研究科
Department of Information & Computer Sciences
Saitama University

- 反射神経を必要とするゲーム（セブンブリッジ、スピードなど）はここでは対象外とする。通信速度による不公平が生じるためである。
- ゲームには、各プレイヤーに手札を配るなどの、ゲームの進行係の役割を持つ「ディーラー（親）」が存在する。一般にディーラーはプレイヤーの1人が兼ねることが多いが、麻雀などの「親」とは違い、カードゲームのディーラーにはゲーム上での特権はないため、ここではディーラーとプレイヤーを区別する。また基本的にディーラーの仕事はサーバが担当する。
- ディーラーは各プレイヤーについて、手札の内容を含むすべての情報を知っており、プレイヤーの行動がルールに則っているか、不正はないかを監視するものとする。また、ディーラーの行うゲーム進行が公平なものであることを、ディーラーがプレイヤーに証明できなくてはならない。
- ゲーム進行の形式は大きく分けて3つある。各プレイヤーに1人ずつ順番に手番が回ってくる形式を順番ゲーム、プレイヤーが決められた枚数のカードを出し合い、一巡したところで最も強いカードを出したものに勝ち点を与える形式をトリックゲーム、より強いカードを出し合っていく、他のプレイヤーがカードを出せなくなるまでカードを出し続けたものに勝ち点を与える形式を変則トリックゲームとそれぞれ名づける。
- 大会中およびゲーム中を除いて、プレイヤーの交替があり得る。例えばあるゲームではプレイヤーでなかった人が順番待ちをしていて、次のゲームにプレイヤーとして参加することがある。またプレイヤーだった人が次のゲームには参加せず、見学にまわることがある。このため、実際にゲームに参加している以外の人を「待機者」と呼び、プレイヤー、待機者、ディーラーを合わせてゲームの「参加者」と呼ぶことにする。参加者は全員名前を持ち、ゲームの進行状況は参加者全員に知らされるものとする。またゲーム進行時の混乱を防ぐため、同じ名前の参加者は存在しないものとする。

図1に、参加者の状態の遷移を示す。

2.2 カードゲームの用語

本論文で用いるカードゲーム用語について説明する。これらはすべてカードゲームの標準的なルールに基づいている¹⁾⁵⁾。

スーツ (suit) スペード、ハートなどの、カードのマーク。

ランク (rank) A, 2, Kなどの、カードの数字。

手札 (hand) プレイヤーが所有するカード。この呼び方は、カードが表を向いているか裏を向いて

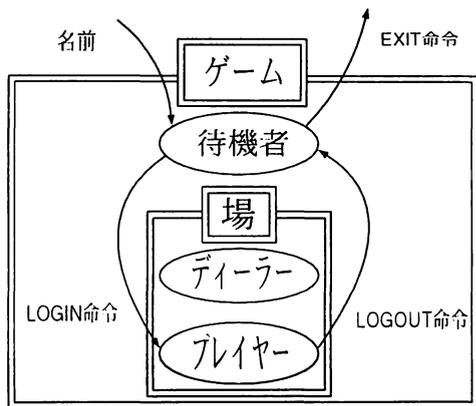


図1 参加者の状態の遷移

表1 汎用プロトコル群に含まれるプロトコル

プロトコルの内容
カードの表現方法
参加者の行動の表現方法
ゲーム進行の形式
ゲームの進行状況の通知
参加者の間の通信
セキュリティ（認証、情報の機密保持）
ルールの記述方法

いるか、また手に持っているかないかということとは無関係である。

場 (table) ゲームが行われる場所のこと。

場札 場に出されたカードのこと。

山札 (talon) 手札、場札以外のカードを、山のように入れて場に置いたもの。

トリック (trick) プレイヤーが1人1枚ずつカードを出し、カードの優劣を競うゲームの形式。または出されたカード、カード群。

手番 (turn) あるプレイヤーがプレイする番がまわってきたとき、そのプレイヤーの手番である、のように言い表す。

3. 汎用プロトコル群の設計

本章では汎用プロトコル群の概要および機能について述べる。このプロトコル群には表1に示すように、インターネット上でカードゲームを行うのに必要な、通信、ルール記述、カードの表現などのプロトコルが含まれている。

3.1 要求される機能

● カードの表現の規定

本プロトコル群では、カードの表現方法として、カードゲームの中でも世界大会などが盛んに行われているコントラクトブリッジにおける表記法を

表 2 ゲーム中のプレイヤーの行動（カードを扱うもの）

行動の内容	命令語	追加情報
手札を場に出す	DISCARD	場に出すカード
カードを山札から引く	DRAW	引くカードの枚数
手札を他のプレイヤーに渡す	PUT	渡す相手の名前、渡すカード
カードを他のプレイヤーからもらう	GET	相手の名前、もらうカードの表現 または位置
役札として手札の一部を公開する	MELD	公開するカード

採用している⁶⁾。

● **参加者の行動の規定**

ゲームに参加するにあたって参加者が取ることのできる行動すべてについて、その内容および表現を規定する。

● **ゲームの進行状況の通知**

インターネットを介してゲームを行うため、あるプレイヤーの手番が終わり、他のプレイヤーの手番になった、ということ、プレイヤーは直接見て知ることができない。ゲームによっては、プレイヤーは現在のプレイヤーの手番かを知ることによって戦略を立てるものがある。このため、ゲームの進行状況を逐一各プレイヤーに伝える機能が必要である。同時に、プレイヤーとしてゲームに参加するために順番待ちをしている待機者がいることを考慮し、進行状況の通知は参加者全員に行われなければならない。

● **プレイヤーの行動の通知**

インターネットを介してゲームを行うため、各プレイヤーは他のプレイヤーの行動を直接見ることができない。このため、あるプレイヤーがとった行動を、参加者全員に通知する機能が必要である。また、ディーラーが各プレイヤーに対して行う行動（手札を配るなど）についても、必要なすべての情報を参加者全員に通知しなければならない。

● **情報の要求**

プレイヤーは、公開されている情報や自分に関する情報をディーラーに要求することができる。例えば、自分の手札にどのカードがあるか忘れてしまったときのために、手札情報の確認を行える機能が必要である。（クライアントソフトウェア側で記録しておくことも可能だが、断線などに備えるためにこの機能は必要である）。要求に対して、ディーラーは必要な情報を通知するという、要求に対する応答の機能も必要である。

以下は、プレイヤーおよび第三者の不正を防ぐための、セキュリティに関する機能である。

● **情報の保護**

例えば、プレイヤー自身しか見ることのできないはずの手札が他のプレイヤーに見えてしまうと、

ゲームの公正さが失われてしまう。これを防ぐため、秘密にすべき情報を保護する機能が必要である。

● **参加者の認証**

第三者のゲームへの不正な介入を防ぐため、参加者の認証を行う。

● **ゲームの公正さの証明**

例えば、ディーラーが特定のプレイヤーに対し、有利な手札を配布といった不正が考えられる。このため、ゲームが公正に行われているかどうかを、プレイヤーに証明する機能が必要である。これは、公式大会での不正を防ぐという理由に加えて、プレイヤーが他のプレイヤーやディーラーを信頼し、安心してゲームを楽しむようにするためにも必要である。

3.2 ルールの記述方法

カードゲームのルールを構成する要素を書き並べることで、さまざまなゲームのルールを統一的に記述することができる。使用するカードの枚数および種類、ゲームに参加可能な人数、ゲームの形式（2.1 節参照）、ゲームの終了条件、勝利条件、といった要素のほか、プレイヤーの行動について設定する必要がある。

例えばプレイヤーの行動のうち、カードを扱うものは5つである。これらをプレイヤーの「命令」として扱う。命令は各々1語の命令語で表される。詳細を表2に示す。

あるゲームについてルールを記述するには、これらの命令がそのゲームで使用可能かどうか、使用可能であれば個々の命令についての制限情報（カードを出すのであれば、一度に何枚出せるか、等）を定める必要がある。

このような、ゲームを構成するのに必要な設定を、本プロトコル群では条件設定と呼ぶ。

3.3 行動内容の通知手順

プレイヤーは行動を起こす際、命令語を使って行動の内容を表す。命令に追加情報を加える場合は、対象となるプレイヤー名、扱うカード、枚数などのその他の情報、の順に列記する。例えばあるプレイヤーがプレイヤーAに、手札からダイヤの3とスペードの4を渡したいときは次のように表現し、これをディーラー

表 3 カードの可視性

状態	状態名
すべての人に見える	CARD_VISIBLE_ALL
手札として持っている プレイヤーのみ見える	CARD_VISIBLE_PLAYER
誰も見られない	CARD_VISIBLE_NOBODY

に通知する。

PUT playerA D3 S4

ディーラーはこの行動について、ゲームのルールに基づいてエラーチェックを行った後、全参加者に対し誰がどのような行動を起こしたかを通信により伝える。ただしこのとき、行動の追加情報のうち知らせるべきでない内容は伏せて通知しなければならない。例えば、プレイヤー A がプレイヤー B に手札から 2 枚を渡した場合、どの手札を渡したかは A、B 以外のプレイヤーに見えないようにすることが多い。このときディーラーは A、B 以外のプレイヤーには「A が B に 2 枚のカードを渡した」としか通知しない。この通知は具体的には、ACTION 命令語を用いて次のように表現される。

ACTION playerA PUT playerB 2

なお、どの情報を誰に公開するかについてはルールの記述時に設定する。

3.4 手札の可視性

カードゲームの 1 人遊びでは、カードの状態は表を向いているか裏を向いているかの 2 つで十分⁷⁾であるが、複数人で行うゲームではこれだけでは不十分である。

手札のなかには、プレイヤーの手札でありながら、そのプレイヤー自身にもそれがどのカードか分からない状態のものがあり得る。カードがどのプレイヤーに所有されているか、または場札、山札に含まれているのか、といった状態に関係なく、そのカードの内容をどのプレイヤーが見る（知る）ことができるのかを、カードの可視性と定義する。3 種類ある可視性を表 3 に示す。

例えば、山札として場に積んであるカードや、プレイヤーの手札でありながらそのプレイヤー自身にも見ることのできないカードは、CARD_VISIBLE_NOBODY という状態で表される。

3.5 ゲームの進行

本プロトコル群では、以下のような手順でゲームを進行するものとする。

- (1) ゲームのプレイヤーを募る。

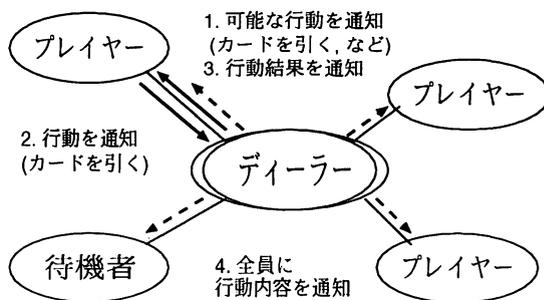


図 2 ディーラー中心のゲーム進行

- (2) プレイヤーが集まったら、ディーラーはゲーム開始を宣言し、ゲームをするための準備をする（手札を配る、等）。
- (3) ディーラーは手番であるプレイヤーに行うべき行動の種類を伝え、選択させる（例えば、カードを出す、カードを引く、等）。
- (1) プレイヤーはディーラーの指示を受けて自分の行動を決定し、その内容をディーラーに伝える（例えば、ハートの 3 を出す）。
- (5) ディーラーは自分が管理している情報と、プレイヤーから伝えられた行動内容とを比較し、行動内容に誤りがないかチェックする（例えば、プレイヤーはハートの 3 を持っているか、等）。誤りがあればプレイヤーに警告を発し、再びプレイヤーに行動をうながすため (3) に戻る。誤りがなければ行動の処理を行い、手番を次のプレイヤーに進める。
- (6) (3)~(5) の手順を繰り返し、ゲームの終了条件を満たしたところで、ディーラーはゲーム終了を宣言する。
- (7) ディーラーは必要であればスコアの集計などを行い、それを各参加者に通知する。
- (8) (1) からの手順を繰り返す。

このように、ゲームはすべてディーラーを中心に進められる（図 2 参照）。

3.6 プロトコル群の階層化

ゲーム中でないプレイヤー、および待機者の行動を表 4 に示す。

これらはカードゲームに限らず、インターネット上で行うゲームすべてにおいて利用可能なものである。

このように、われわれが開発した汎用プロトコル群には、カードゲームに限らず、ネットワーク上で行うゲーム全般に適用できるものがある。また見方を変えれば、ネットワーク上で行うカードゲームは、ネットゲームを特化したものと考えられる（図 3）。したがって、プロトコル群を階層的に分けることにする。

表 4 ゲーム中以外のプレイヤーの行動

行動の内容	命令語	追加情報
プレイヤーになる	LOGIN	
プレイヤーから外れる	LOGOUT	
参加者になる	ENTER	(参加者としての) 名前
参加者から外れる	EXIT	
席 (プレイの順番) を変える	SEAT	席を交換したいプレイヤー名

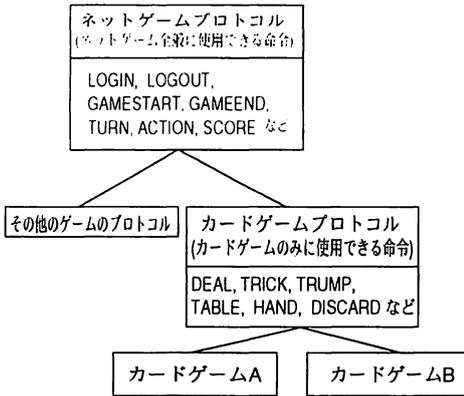


図 3 プロトコルの階層化

以下、ネットゲーム全般に適用できるプロトコルをまとめてネットゲームプロトコル、カードゲームにのみ適用できるプロトコルをカードゲームプロトコルと記述する。ネットゲームプロトコルを特化することにより、カードゲーム以外のネットゲームも同様のプロトコルで規定することが可能であると考えられる。

4. ツールキットの設計

汎用プロトコル群に基づいて設計した、インターネット上で行うカードゲームプログラム開発のためのツールキットについて述べる。このツールキットはプログラム開発のために約 60 個の条件設定およびツールをライブラリとして用意しており、これらを使うことで、サーバ (ディーラー) およびクライアント (参加者) ソフトウェアを簡単に開発することができる。

4.1 サーバソフトウェアの機能

サーバソフトウェア開発ツールは、図 3 に示したプロトコル群の階層化に基づき、カードゲームにのみ適用可能なカードゲームツールと、インターネット上で行うゲーム全般に適用可能なネットゲームツールとに分けられている。

ネットゲームツールは以下の機能を持つ。

- 参加者からの接続要求の処理
- プレイヤーの認証

- ゲーム全体の流れの制御
- 参加者からの入力 の監視
- 参加者からの入力 の処理
- 例外処理
- 特定あるいは全参加者に対する、ゲーム進行状況の通知
- データ通信時の機密保持
- 保管データの機密保持
- ゲーム結果 (集計得点、プレイの経過) の記録

このツール群では、サーバ・クライアント間の通信プロトコルとして TCP/IP 接続を用い、ソケットを用いて接続処理を行うサーバとしての機能を提供する。

参加者に対して行う通知はすべて、ネットゲームプロトコル (3.1 節参照) に基づいて行われる。ログイン、ログアウト、手番の通知、あるプレイヤーの行動を全プレイヤーに通知、などの内容を扱う。

また、各参加者ごとに通信のためのスレッドを生成し、参加者からの入力を常に監視する。これにより、参加者からの要求を逐次的に処理する。

ゲームの流れの制御とは、ゲームの状態を準備段階、ゲーム段階、結果処理の 3 つに分け、この状態遷移を制御する機能である。

例外処理とは、停電などによる通信路の断線時に行う処理である。ゲームを遊びに使う場合はさほど問題はないが、公式大会では断線時の対応はきわめて重要である。そこで、ゲームがどれほど重要であるかによって、または大会のルールによって扱いを変更できるように、断線したプレイヤーや特定の時間以上応答のないプレイヤーが出た場合は、1. ロボットプレイヤー (ゲームの結果への影響を最小限にするため、ランダムにゲームを進めるプログラムプレイヤー) に代理をさせる、2. 断線したプレイヤーの負けとする、3. 試合経過を保存しておき、次の機会に断線する前の場面からゲームを続ける、の 3 種類から選べるものとする。

ゲーム結果の記録とは、ゲームの結果 (集計得点やプレイの経過) をファイルやデータベースに記録する機能である。作成したプログラムを公式大会に使用する場合には必要不可欠な機能である。

カードゲームツールはこのネットゲームツールを拡張し、カードゲーム特有の機能を付加したものである。カードゲームツール固有の機能は以下のものである。

- カードゲームで使用する命令の処理
- ゲームルールの規定
- プレイヤーの行動をルールと照合し、正誤を判定
- プレイヤーの行動が反則であったときの処理
- ゲームで使用するデータの管理

1 つめの機能はカードゲームプロトコル (3.1 節参照) に基づいた機能で、実際にこのプロトコルをゲームに使用する手段を提供する。

また 3.2 節で述べたゲームのルールの記述を行うため、カードゲームツールにはさまざまな条件設定を用意している。ルールを読み込んでインタープリタのように処理する方式⁸⁾⁹⁾ も考えられるが、カードゲームの種類が多いこと、各ゲームに固有のルールが多いことから、ライブラリを使ってルールを記述する方式を採用した。

4.2 クライアントソフトウェアの機能

クライアントソフトウェア開発のためのツール群も、図 3 で示したプロトコル群の階層化に基づき、インターネット上で行うゲーム全般に適用可能なネットゲームツールと、カードゲームのみに適用可能なカードゲームツールとに分けられる。

ネットゲームツールは以下の機能を提供する。

- ゲームサーバ (ディーラープログラム) との接続
- ソケットによる通信
- 保管データの機密保持
- データ通信時の機密保持

機密保持についてはサーバと同様である。

カードゲームツールは以下の機能を提供する。

- GUI によるプレイヤーの操作の補佐
- GUI 表示部の表示方法をゲーム製作者の好みに応じて変更できる機能

GUI により、利用者はマウス操作と若干のキーボード操作 (名前の入力程度) だけでゲームを楽しむことができる。ネットゲームツールのみを使ってもゲームを行うことが可能であるが、操作が面倒なこと、字面ばかりでゲームの面白さが削られることがあり、GUI は不可欠である。

4.3 ツールキットの拡張

ツールキットにはプログラム開発を助ける約 60 個の条件設定が用意されている。これらのツールは多くのカードゲームに共通する要素から選択したものであるため、単純にこれらのツールを組み合わせるだけで完全にルールを表現できるゲームは少ない。

この問題を解決するには、新たに各ツールを追加し

たり、用意されているツールを拡張する機能があればよい。図 4 に示すように、ツールキットによるプログラム開発は積木の組み立てのようにとらえることができる。このとき、新しい設定を追加するのは新しい形のブロックを用意すること、ツールの拡張はブロックの形を変形させたり、飾りをつけたりすることと見なすことができる。

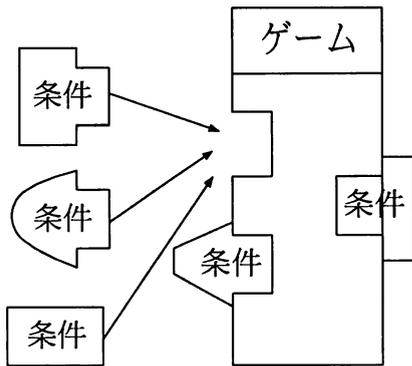


図 4 積木の組み立てに例えたプログラム作成

4.4 Java による実装

われわれは設計したツールキットを Java で実装した。これによりさまざまなプラットフォーム上で動作させることができる。またツールキットの階層的な構造やツールの拡張といった性質は、Java の持つオブジェクト指向言語の特徴を利用すると表現しやすい。

例えばサーバソフトウェア開発のためのネットゲームツールの構成は、図 3 に示すように、最上位クラスのネットゲームツールでネットゲームプロトコルを実装し、これを継承した下位クラスのカードゲームツールでカードゲームプロトコルを実装している。実際にカードゲームプログラムを開発するには、カードゲームツールを継承したクラスを用意し、ゲームのルールに沿って条件設定を記述すればよい。こうすることで、上位クラスの持つ通信、データ管理などの機能をそのまま引き継ぐことができるため、ゲームに依存しない機能をプログラミングする労力を省くことができる。

またゲームのルールを記述する際には、ツールとして用意されている既存のメソッド (関数) をオーバーライド (上書き) することで、ツール内容を拡張、変更できるため、さまざまなゲームに対応させることが可能になる。

5. ツールキット利用の実例

われわれは汎用プロトコルおよびツールキットを評価するため、開発したツールキットを使って、ホイス

ト、ページワンという2種類のカードゲームのプログラムを開発した。ホイストは歴史が古く、欧米で人気の高いゲームである。またページワンはバリエーションを多く持つゲームであり、実現にあたっては各バリエーションに柔軟に対応する必要がある。本論文ではページワンのプログラムを紹介する。

次ページに、ツールキットを利用して作成した、ページワンのサーバプログラムのソースコードを掲載する。PageOneServer クラスがページワンサーバプログラムの本体であり、このクラスはサーバソフトウェア用のカードゲームツールである CardGameServer クラスを継承している。

ソースの大部分は条件設定に費されているが、ページワンというゲームはツールキットで用意しているツールだけではルールを記述することができない。これを解消するため、メソッド drawCheck() をオーバーライドして機能を拡張している。拡張された drawCheck() は「前の手番のプレイヤーの場札と、同じスートまたは同じランクのカードを手札として持っているとき、そのプレイヤーは山札からカードを引くことができない」というルールを表現している。

他のメソッドについての解説はここでは省略する。大まかな内容についてはソース中のコメント文を参照いただきたい。

```

1 public class PageOneServer
2     extends CardGameServer
3 {
4
5     public static void main(
6         String args[])
7     {
8         // ゲーム本体の用意と作動
9         getArgs(args);
10        PageOneServer server =
11            new PageOneServer();
12        server.gameServe();
13    }
14
15    /** カード配布に関する条件設定 */
16    protected void conditionsForDeal()
17    {
18        deal_setCardNumEach(5);
19    }
20
21    /** 山札に関する条件設定 */
22    protected void conditionsForStock()
23    {
24        stock_toTableNum(1);
25        stock_allFaceDown();
26    }
27
28    /** ゲームの終了条件に関する条件設定 */
29    protected
30        void conditionsForGameEnd()
31    {
32        gameend_noHandOne();
33    }
34
35    /** DISCARD 命令に関する条件設定 */
36    protected
37        void conditionsForDiscard()
38    {
39        discard_available();
40        discard_turnEnd();
41        discard_visible(CARD_VISIBLE_ALL);
42        discard_lastNumberOrSuitOnly();
43        discard_numberLimit(1);
44    }
45
46    /** DRAW 命令に関する条件設定 */
47    protected
48        void conditionsForDraw()
49    {
50        draw_available();
51        draw_numberLimit(1);
52        draw_tableToStock();
53    }
54
55    /** その他の条件設定 */
56    protected
57        void conditionsForOthers()
58    {
59        setName("Page 1");
60        setPlayersNumLimit(3, 7);
61        setGameFormat(ORDER_GAME);
62        setLastOneCardMessage("Page 1!");
63        setCardNum(52);
64    }
65
66
67    /** DRAW 時のルールチェック */
68    protected boolean drawCheck(
69        int num, String params)
70    {
71        // conditionsForDraw() で指定した
72        // ルールに反する場合は DRAW 不可
73        if(!super.drawCheck(num, cmd))
74            return false;
75
76        // ページワンに固有のルール:
77        // 場札と同じランクまたはスートの

```

```

78 // カードを持っているか
79     if(player(num).hasThisSuit(
80         Card.getSuit(
81             cardsOnTable(1))) ||
82         player(num).hasThisNumber(
83             Card.getSuit(
84                 cardsOnTable(1))))
85     {
86 // 持っている : DRAW 不可
87         sendWarning(num, W_ILLEGAL
88             +" : You can discard!");
89         return false;
90     }
91 // 持っていない : DRAW 可
92     return true;
93 }
94 }

```

6. 考 察

ツールキットの実例として開発した 2 種類のゲームのプログラムソースはいずれも 100 行未満であった。開発をツールキットなしで行った場合、プレイヤー間の通信や情報管理、ゲーム進行の制御などの処理を書く必要があるため、ソースコードは約 1200 行になった。ツールキットを使用することで、ソースコードの分量を抑えることができたばかりでなく、通信などの機能を上位クラスで実現しているため、ゲームのルールの記述にさえ注意すれば簡単にプログラムを開発することができた。ルールの記述に際しては細かい点まで留意しなければならないが、このわずらわしさはツールキットの有無に関係なく生じるものである。

またツールキットの利用は、同様の方法で他のゲーム用のプログラムも開発できるという利点もある。ただし、いくつかの機能（ビidding、ゲーム結果の記録など）についてはまだ実現していないため、そのような機能が必要なゲームについては開発者自身が機能を実現する必要がある。

プロトコル群に関しては、このプロトコル群に従ってゲームを行うことができることが確認できた。しかしカードゲームには多彩なバリエーションがあるため、现阶段のプロトコル群ではカバーしきれないゲームもあると考えられる。

7. おわりに

われわれはネットワーク上で行うカードゲームのための汎用プロトコルの開発を行った。また、汎用プロトコルに基づき、カードゲームプログラム開発のためのツールキットを設計、Java 言語で実装した。さらにこのツールキットを用いて 2 種類のカードゲームプログラムを実際に開発する実験を行い、プロトコルの

有用性、ツールキットの有益性を確認した。

プロトコル群については、より多くのゲームに対応できるように、命令やルールの記述方法を改良する必要があると思われる。またゲームを記録するためのフォーマットの規定、停電などの突発的事態に対するより有効な対処法の検討などが課題として挙げられる。

ツールキットに関しては、多くのゲームをわかりやすく表現できるような GUI の表示方法の検討、および、ルールにバリエーションを持つゲームを開発する際、1つのプログラムだけでさまざまなバリエーションに対応できる機能を付加することが、今後の課題である。

なお本研究で実装したツールキットのプログラムは、<http://www.aise.ics.saitama-u.ac.jp/~koide/cardgame/>で公開する予定である。

謝辞 論文の執筆にご協力頂いた埼玉大学大学院理工学研究科の高橋大介助手、並びに埼玉大学工学部情報システム工学科研究室の皆様には感謝致します。

参 考 文 献

- 1) Yahoo! Inc: Yahoo! Games. <http://games.yahoo.co.jp>.
- 2) Blue Chip Bridge: Blue Chip Bridge. <http://www.bluechipbridge.co.uk/protocol.htm>.
- 3) Buro, M. and Durdanovic, I.: An Overview of NECI's Generic Game Server, *Proceedings of The 6th Computer Games Olympiad Workshop* (2001).
- 4) 木村健太郎: トランプの遊び方, 梧桐書院 (1996).
- 5) 大島史郎: トランプ遊び, 日東書院 (1985).
- 6) American Contract Bridge League: American Contract Bridge League Home Page. <http://www.acbl.org/>.
- 7) Department of Computer Science Rochester Institute of Technology: Using the Eiffel Card Game Classes. <http://www.cs.rit.edu/icss232/proj2/cardgame.html>.
- 8) Pell, B.: A Strategic Metagame Player for General Chess-like Games, *Computational Intelligence*, Vol. 12.
- 9) Pell, B.: Metagame: A New Challenge for Games and Learning, *Heuristic Programming in Artificial Intelligence 3 The Third Computer Olympiad* (1992).