碁盤システムの実装（第１報）

菅 原 英 一・増 永 良 文*

Implementation of GOBAN System (1st Report)

Eiichi SUGAWARA and Yoshifumi MASUNAGA*

(1994年８月22日受理)

１．はじめに

筆者らは，これまで囲碁の対局をモデル化するに当たって，OOA（Object-Oriented Analysis）の手法を適用することを試み，各オブジェクト間のメッセージパッシングについて考察した[1]。このモデルでは，プレイヤは双方共に人間プレイヤであり，システムは単に囲碁の対局環境を提供するという意味で，これを『碁盤システム』と呼んだが，今回はこの『碁盤システム』をコンピュータ上に実装することを試みた。

２．オブジェクト指向囲碁対局モデル

２．１　モデルの改良

囲碁の対局をモデル化するに際しては，初めはできるだけ実世界（実際の対局）を忠実に反映するよう心がけたが，このままでは冗長な部分が多くコンピュータ上への実装を考えた場合，効率の良いプログラムは期待できない。そこで，当初のモデルから冗長な部分を省くなどの改良を加えた図１に示すような新しいモデルを作成し，これに基づいてシステムの設計・実装を試みた。

図１において，『碁盤システム』とは点線で囲まれた部分であるが，当初のモデルの改良点は以下の通りである。なお，【　】内はオブジェクトを表す。

(1) 知識は【プレイヤ】が属性として持つことにし，【知識】をモデルから削除する。

(2) プレイヤの着手は【碁盤】がその着点の色を変えることで実現できるので，【黒石】，【白石】をモデルから削除する。

(3) 実世界を忠実に反映するために【立会人】を設けたが，【ルール】に【立会人】の役割を兼務させて新たに【ルールマネジャ】を設けることにし，【立会人】をモデルから削除する。

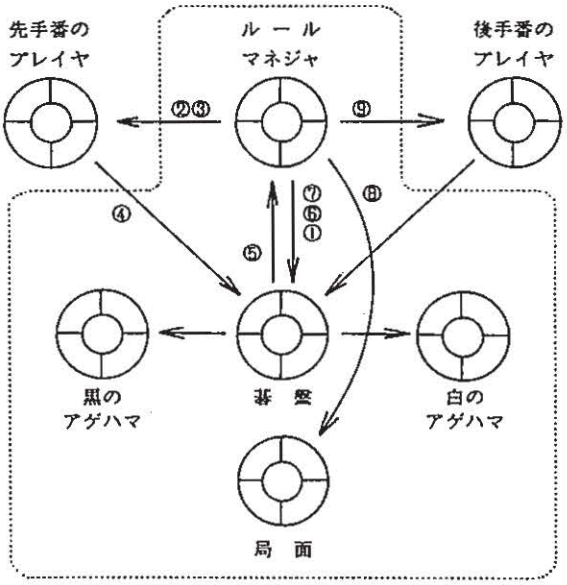

図１　オブジェクト指向囲碁対局モデル

(4) プログラミング上，【プレイヤ】が相手方の石を盤上から取り上げて味方のアゲハマに加えることは冗長な作業であるから，アゲハマを伴う処理は【ルールマネジャ】の指示によって【碁盤】が行うことにする。

２．２　多重レイヤーモデル

改良を加えたモデルについて，これをOOAの記法による多重レイヤーモデル[2]で表すと図２のようになる。

図中で，二重の長方形で囲まれたシンボルが「クラス&オブジェクト」を表し，半円形のシンボル及び三角形のシンボルがそれぞれ「汎化—特化構造」と「全体—部分構造」を示している。すなわち，【プレ

*図書館情報大学

イヤ】の特化として【黒番のプレイヤ】及び【白番のプレイヤ】が存在するということであり，【アゲハマ】の場合も同様である。また，打ち上げの単位となる【連】は【点】の集合から成る複合オブジェクトであり，同様に死活の単位となる【群】は【連】の集合から成る複合オブジェクトであることを示している。そして，【碁盤】は現時点の局面情報を提供する立場上，【群】や【連】から成る盤上の局面情報と共に双方の【アゲハマ】をも，その部分として加えることになる。

なお，「全体—部分構造」及び「インスタンス結合」における値域は以下の条件による。

(1) 【連】を構成する【点】の最大値：100
(2) 【群】を構成する【連】の最大値：50
(3) 現局面を構成する【群】の最大値：30
(4) 現局面を構成する【連】の最大値：100
(5) 双方の【アゲハマ】の最大値：300
(6) 一局の手数の最大値：400

また，矢印の付いた点線はオブジェクト間のメッセージパッシングを表し，丸で囲んだ数字はその発生順序を示している。図2は交互着手による局面の進行過程の例であるが，局面は以下に述べるようなメッセージのやりとりによって進行する。

すなわち，【ルールマネジャ】が対局に先立って【碁盤】に対して初期値設定を指示し，次いで対局開始宣言と同時に【プレイヤ】に対して着手を促すことで対局が始まる。手番の【プレイヤ】が着手を決めて【碁盤】に伝えると，【碁盤】はこの着手の合法性を【ルールマネジャ】に問い，戻り値として“合法”を返されたとき，【ルールマネジャ】の指示に従って各種テーブル類を更新し，【局面】は着手を記録する。これらの作業が終わると，【ルールマネジャ】は【プレイヤ】に手番の交替を告げ，以上の繰り返しによって交互着手による局面の進行がはかられる。

3．碁盤システム

3．1　システムの機能

このシステムでは，両対局者共に人間プレイヤであり，石の死活や手入れの問題についてはプレイヤの判断に依存している。すなわち，死活の単位となる【群】を対象外としているため，図2における全体—部分構造のうち【碁盤】—【群】—【連】の部分がシステムに取り込まれていない。したがって，システムとしての機能は以下に示す通りである。

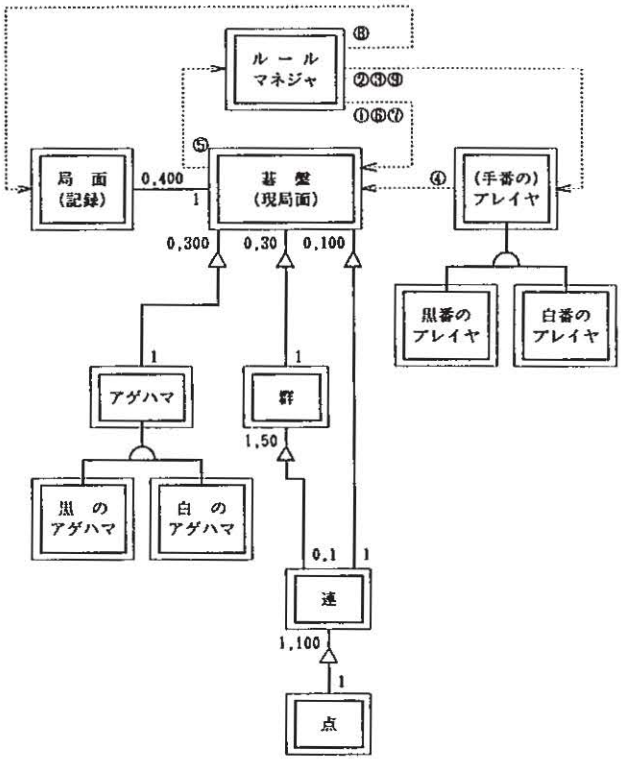

図2　多重レイヤーモデル（局面進行過程の例）

(1) 着手の合法性の判断
（日本囲碁規約との照合）
(2) 違法着手に対するメッセージの出力
(3) 着手の記録
(4) 終局後の地の計算と勝敗の決定

3．2　システムの基本構成

本システムは大きく分けて以下の3つのモジュールから構成されている。

(1) 対局モジュール
対局に直接かかわる部分で，【ルールマネジャ】が統括している。
(2) 画面表示モジュール
グラフィックデータを画面に表示する部分で，次のようなデータを表示する。
・プレイヤ情報（アゲハマの数，考慮時間など）
・対局情報（通算手数，消費時間など）
・メッセージ（例えば違法着手の場合などに出力）
・各種タイトル（入力選択用のメニュー画面など）
(3) 制御モジュール
入力装置としてのマウスの制御およびグラフィック画面のページの制御を行う。

3．3　クラス設計の例

対局モジュール中の盤上の局面に関連するクラスについて，その主な属性とサービスを以下に示す。

◆　クラス&オブジェクト：点

〈属性〉

・点の盤上における位置
・その点の（石の）色
・その点の属する連の番号
・同一連内の点同士の連結用ポインタ

〈サービス〉

・盤上における点の位置データの取得
・点の色データの取得
・点が属する連の連番号の取得
・連結用ポインタの取得

◆　クラス&オブジェクト：連

〈属性〉

・連を構成する石の色
・隣接空点の数
・連を構成する石の数
・連を構成する点へのポインタ

〈サービス〉

・連を構成する石の色データの取得
・隣接空点数の取得
・連を構成する石数の取得
・連を構成する点へのポインタの取得

◆　クラス&オブジェクト：碁盤

〈属性〉

・着点に隣接する味方連の連番号
・着点に隣接する相手方連の連番号

〈サービス〉

・着手による点の生成
・着手による連（単連）の生成
・着点に隣接する連の識別
・連の連結
・連の隣接空点数のカウント
・盤上からの石の取り上げ
・連の消滅（削除）
・アゲハマのカウント
・着手の合法性判定の依頼
（⇨ルールマネジャ）

このシステムはBORLAND社のオブジェクト指向プログラミング言語Turbo　C++バージョン1.01（AT & T C++バージョン2.0に完全準拠）で実装されており，図3に【連】のクラス定義の一部を示す。なお，【連】を構成する【点】の集合はリスト構造を用いて実現している。

```
class Ren {
        int color;  // 連を構成する石の色
        int empnm;  // 隣接空点の数
        int tennm;  // 連を構成する石の数
        Ten *pt;    // 連を構成する点へのポインタ
public:
        Ren() {color=0;empnm=0;tennm=0;}
        void set_color(int cd){color=cd;}
        void set_empnm(int ed){empnm=ed;}
        void set_tennm(int td){tennm=td;}
        void set_tenpt(Ten *pd){Ten *pt=Ten *pd;}
        int get_color(void){return(color);}
        int get_empnm(void){return(empnm);}
        int get_tennm(void){return(tennm);}
        void get_tenpt(void){return(Ten *pt);}
};
```

図3　【連】のクラス定義

3．4　連の生成・成長・消滅

盤上への着手によって（黒または白の）点が生成されるが，この点はまた最小の連（単連と称する）でもある。そして，点の連結によって連が成長し，また隣接空点のなくなった連は盤上から取り上げられて消滅する。図4にこの流れを示す。

盤上の着点（x，y）への着手があると，この石一個から成る連を仮に生成する。次いで，盤上においてこの点に隣接する点（位置パラメータiで識別）に味方（同色）の連がある場合には連結して成長する。また，隣接する点に相手方（異色）の連がある場合には，この着手によって当該連の隣接空点数が零になれば盤上から取り上げられて消滅する。

4．碁盤システムの実装

前述のように，今回のシステムはBORLAND社のTurbo C++言語で書かれ，これを実装するハードウェアとしてはEPSON社のパーソナルコンピュータPC-286VFを用いている。なお，他にハードディスク100MBと入力装置としてのバスマウスが必要である。

本システムの画面表示の一例を図5に示す。初めにメインメニューにおいて「対局」，「置き石」，「終了」の中から，互先の対局であれば直ちに「対局」を選択することで対局が開始され，第一手目の入力待ちとなる。プレイヤ間に棋力の差がある場合には「置き石」を選択することによりサブメニューが表示され，棋力差に応じた置き石を設定できる。

対局開始後は，画面上の碁盤の対応する点にマウスカーソルを移動してクリックすることが着手ということになるが，これに先立ってサブ入力画面が表示されて，「着手」の他に「着手放棄」や「投了」も選択できる。

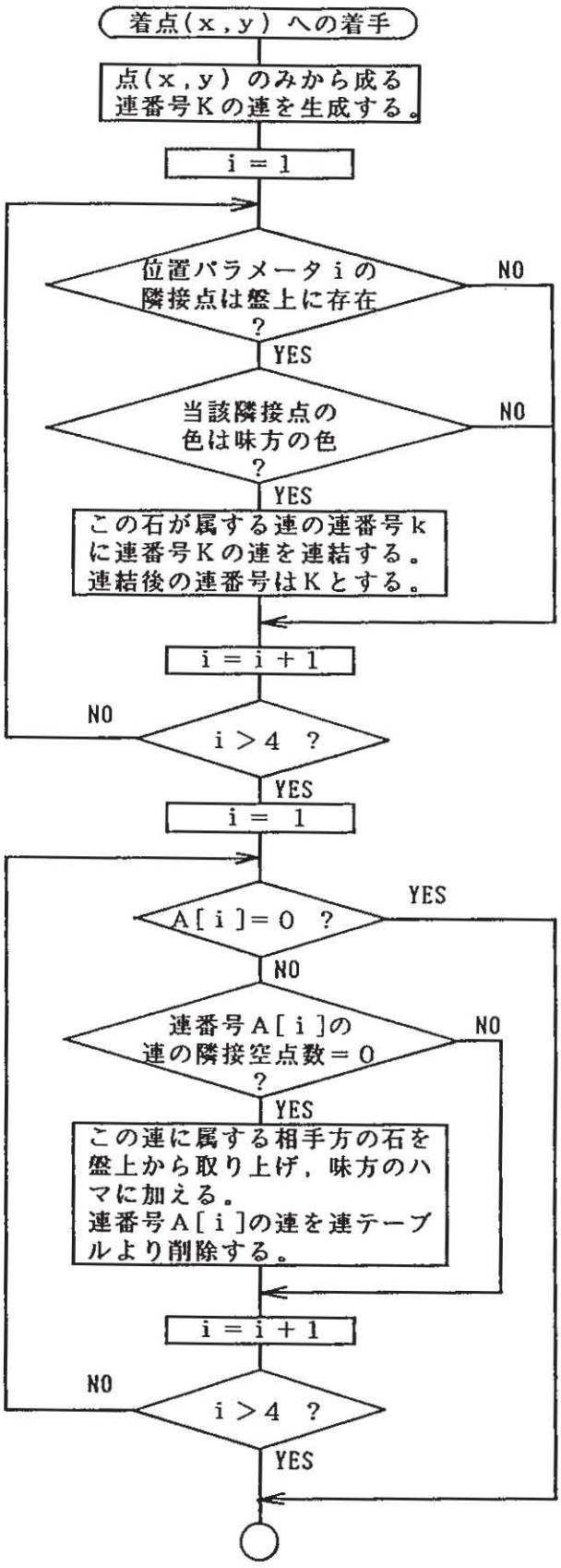

図4　連の生成・成長・消滅

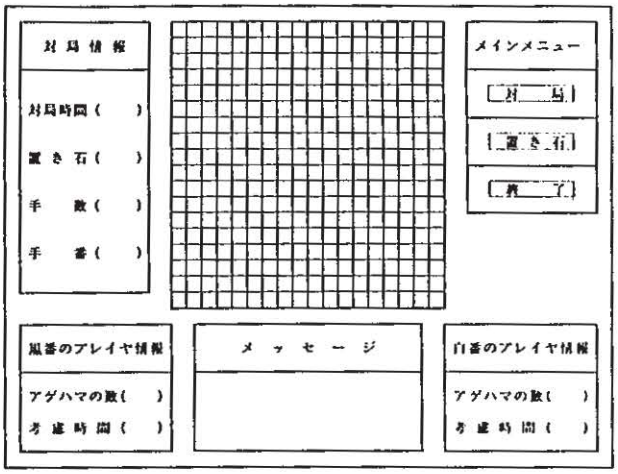

図5　画面表示の一例

対局中は画面に対局情報として「対局時間（消費時間）」・「置き石（何子局）」・「通算手数」・「現在の手番」が表示され，さらにプレイヤ情報として現時点における双方の「アゲハマの数」と「考慮時間」が表示される。また，違法な着手が入力された場合にはメッセージ画面にその理由と共にメッセージを表示し，再度の着手を促すことにしている。

対局を終了させるにはメインメニューで「終了」を選択すればよい。現在のところ，終了後の画面がそのまま初期画面になっている。

5．おわりに

今回のシステムでは石の死活などプレイヤに依存する部分もあるが，コンピュータ上での囲碁の対局を可能にした。すなわち，死活の単位となる【群】をシステムに取り込んでいないため，その判断はプレイヤに依存する形になっている。したがって，初心者同士の対局ではよく見かける光景であるが，終局時に石の死活をめぐるプレイヤ同士の見解が異なる場合でも，現在はシステムとして対処すべき方策はない。

今後は，このような場合でも中立の立場で厳正な判定が下せるよう，【群】を取り込むことによってプレイヤに依存することのない完全な形での碁盤システムの実装が次の目標である。

終わりに，プログラムの作成に当たってご尽力いただいた昨年度当研究室所属の草薙輝章，小野寺優の両君，ならびに本年度当研究室所属の佐藤敦彦君に感謝の意を表する。

参考文献

1) 菅原英一，増永良文：囲碁プログラムのオブジェクト指向モデル，秋田工業高等専門学校研究紀要，No. 29, pp. 32-39, (1993)
2) Peter Coad and Edward Yourdon: Object-Oriented Analysis (book), Prentice Hall Inc., 233p., (1991)