# Un modèle à objets pour décrire les groupes du jeu de Go

Bruno Bouzy

LAFORIA-IBP
Université Paris VI - Tour 46 2ème étage
4, place Jussieu 75252 PARIS
Tél.: 44 27 70 10 Fax: 44 27 70 00
e-mail : bouzy@laforia.ibp.fr

Résumé:

Nous présentons dans cet article un modèle à objets qui décrit les groupes au jeu de Go. La programmation du jeu de Go résiste encore à l'informatisation. La notion d'objet facilite la modélisation du jeu de Go. Le concept central est celui de "groupe". Un groupe possède des propriétés intrinsèques et des propriétés interactives. Ces propriétés sont synthétisées pour définir la santé d'un groupe. La santé des groupes permet d'interpréter une position. Le modèle est implémenté dans le programme Mood Indigo. Notre méthode expérimentale itérative a permis d'identifier et de placer correctement les concepts les uns par rapport aux autres. Une description des classes du modèle objet est présentée en annexe.

Abstract:

This paper shows an object oriented description of clusters in the game of Go. Computers Go programs are still very weak. Objects keep the modelisation process easier. The main class is the "cluster" class. A cluster gets his owns properties and interactives properties. Those properties allow to define a synthetic propertie : the health of a cluster which is a good way to understand a Go position. The software Mood Indigo is an implementation of this model. We followed an iterative experimental method to identify the objects and find a good placement between them. A class description of this model stands at the end of the paper.

## INTRODUCTION

Le jeu de Go est un jeu très ancien dont l'origine remonterait à plus de 4000 ans et se situerait en Chine. En Europe, il reste encore mal connu du grand public. Le diagramme 1 ci-dessous montre une position au jeu de Go:

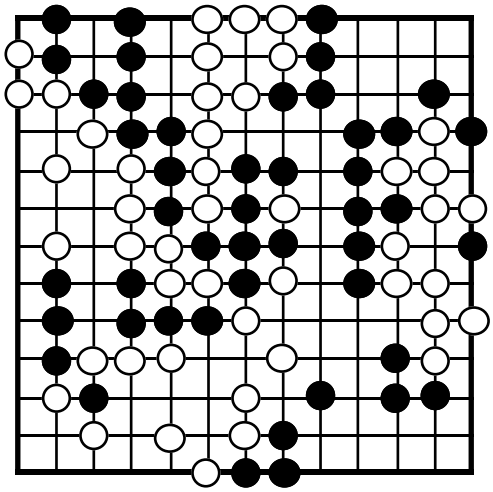


*diagramme 1*

La position paraît chaotique mais un bon joueur de Go est capable de l'interpréter, c'est-à-dire savoir quelles intersections sont contrôlées par Blanc et Noir et par conséquent savoir qui gagne.

Le but de cet article est double. Premièrement, nous montrons pourquoi la notion d'objet facilite la modélisation du jeu de Go. Deuxièmement, nous présentons un modèle à objets du jeu de Go qui permet de montrer l'ordre caché de la position du diagramme 1.

Notre modèle est implémenté dans notre système appelé Mood Indigo (My Object Oriented Design Is Now Designed In Good Objects).

### Plan de l'article:

Après avoir donné un bref aperçu de l'état de l'art de la programmation du Go, nous montrons pourquoi le jeu de Go est un domaine adapté à la modélisation orientée objet. Nous définissons des notations visuelles et ludiques. Nous présentons ensuite les concepts fondamentaux qui doivent constituer à notre avis le coeur de tout système jouant au Go. Nous discutons en conclusion de la méthode qui nous a permis de construire le modèle. En annexe, se trouve la description des classes d'objets du modèle.

### Remarque:

Le modèle présenté fait partie d'un travail plus vaste sur la modélisation cognitive du comportement du joueur de Go humain. Cette modélisation est trop complexe pour être présentée dans cet article à un non joueur de Go. C'est pourquoi nous ne présentons qu'une partie de cette modélisation. Il n'est pas nécessaire de connaître le jeu de Go pour comprendre cet article.

## ÉTAT DE L'ART DE LA PROGRAMMATION DU JEU DE GO

### Le jeu de Go résiste à l'informatisation

Les meilleurs programmes de Go, Goliath [Boon 91] , Go Intellect [Chen 89] et Handtalk, ne sont que 12ème kyu[2], c'est-à-dire du niveau d'un joueur moyen. Ceci est dû principalement au fait que le nombre de coups possibles sur une position moyenne est grand [Hamann 85] (de 50 à 300 environ) et donc qu'une force brute du type de celle employée par Deep Thought aux Echecs [Hsu 90] ne s'applique pas au Go.

Une alternative est de s'inspirer de la façon de jouer des joueurs humains [Bouzy Victorri 92]. D'un coup d'oeil, un joueur perçoit rapidement des concepts nombreux et hétérogènes. Il suffit pour s'en convaincre d'observer des parties rapides. Mais il est difficile d'expliciter ces concepts car ils sont majoritairement subconscients. D'autre part, les quelques concepts explicités actuellement sont des concepts morphologiques dont la reconnaissance et l'utilisation sont coûteuses pour les machines actuelles. Les machines restent très inférieures à l'homme pour ces deux raisons.

### Mais de francs succès ont été obtenus sur des sous-problèmes du jeu de Go

Problèmes de vie et mort

Sur des problèmes de base de vie de groupes, Risiko résoud des problèmes du niveau des meilleurs joueurs amateurs [Wolf 92]. Ces problèmes supposent que le groupe étudié n'a pas d'interaction avec son voisinage. Notre système résoud aussi des problèmes de vie et de mort des groupes mais en tenant compte des interactions avec le voisinage d'un groupe.

Problèmes de toute fin de partie

Sur des fins de parties où le jeu de Go est décomposable en sous-jeux indépendants, une machine qui utilise une adaptation de la théorie des jeux de Conway [Conway 82] peut jouer mieux que les meilleurs joueurs de Go professionnels [Berlekamp 90]. Malheureusement, il est impossible d'appliquer cette théorie aux autres phases du jeu.

[2]Le niveau des joueurs de Go est évalué avec un système de "kyu" et de "dan". Les débutants complets sonts 30ème kyu. On peut devenir rapidement 20ème kyu en quelques parties. Les bons joueurs sont 1er kyu puis 1er dan. Les meilleurs joueurs du monde sont 10ème dan.

## LA NOTION D'OBJET EST ADAPTÉE AU JEU DE GO

Dans ce paragraphe nous montrons l'interêt d'un modèle à objets pour modéliser le jeu de Go.

### Différents niveaux d'abstraction

Au niveau bas, le joueur de Go humain voit des pierres blanches et noires disposées sur un goban. Il n'utilise pas cette description mais des descriptions abstraites du goban. Les descriptions du goban à différents niveaux d'abstraction sont facilitées par une représentation orientée objet. Pour fixer les idées du lecteur, nous listons ci-dessous les trois niveaux d'abstraction présents dans Mood Indigo.

Le niveau chaîne

C'est le niveau correspondant à celui de la règle du jeu : les objets de ce niveau sont les chaînes de pierres définies par la règle du jeu.

Le niveau groupe

C'est le niveau correspondant à celui du joueur : les objets de ce niveau sont les "groupes" de pierres qu'un joueur de Go reconnaît. Ce niveau dépend largement de la force du joueur. C'est le niveau le plus important car il permet de regrouper les intersections en "groupes" pour diminuer la complexité.

Le niveau goban

C'est le niveau stratégique qui décide de l'objectif le plus important à atteindre pour gagner la partie.

### Différents concepts

En feuilletant des livres de Go ou en écoutant des commentaires de parties de Go, nous nous sommes aperçus que le joueur de Go manipulait des concepts variés : des yeux, des connexions et des séparations, des groupes, des territoires et des frontières, des encerclements, des combats, des questions et des réponses, l'initiative, l'attaque et la défense, l'investissement, la force et la faiblesse, la rentabilité.

Comment identifier et agencer ces concepts ?

Si l'identification de ces concepts résulte de méthodes de verbalisation issues de la modélisation cognitive [Ericsson] [Vermersch91], une méthode orientée objet s'impose pour concevoir, construire et valider le modèle décrivant le comportement du joueur de Go humain.

Les concepts sont hétérogènes

Dans une position de milieu de partie, il y a des chaînes, des "ponts" ou "barrières", des "yeux", des "groupes", des "territoires", des "combats", des "encerclements". Une modélisation par objets qui propose une hiérarchie de classes facilite la classification de ces concepts. La notion d'héritage permet de regrouper les concepts en fonction de leurs points communs.

Chaque concept est instancié plusieurs fois sur chaque position

Dans une position de milieu de partie, en simplifiant énormément, il y a 361 intersections, 50 à 100 chaînes, 100 à 200 "ponts" ou "barrières", 36 "yeux" en genèse, 12 "groupes", autant de "territoires". Le jeu de Go peut donc être vu sous forme d'un multi-jeu. La modélisation objet permet d'organiser la gestion des instances au niveau de chaque classe.

Le joueur peut agir sur chaque objet qu'il reconnaît

En posant une pierre sur le goban, le joueur effectue des actions sur des objets reconnus. Il peut simplement contrôler une intersection mais aussi capturer ou sauver une chaîne, faire ou détruire un oeil, connecter ou séparer un groupe, agrandir ou réduire un territoire.

Un des problèmes que le modélisateur de Go rencontre rapidement est que le fait même d'agir sur un objet abstrait modifie l'objet abstrait lui-même (et souvent le fait disparaître, l'exemple trivial est la capture d'une chaîne) au point de remettre en cause l'idée d'effectuer cette action. La réification de ces actions clarifie la conception du modèle.

Un couple d'actions opposées sur un objet forme un jeu.

Une chaîne d'une couleur donnée possède deux actions associées : sauver la chaîne et capturer la chaîne. Le joueur de la couleur joue des coups pour sauver la chaîne et le joueur de l'autre couleur pour la capturer. Un jeu au sens de Conway [Conway82] se crée. Le joueur de la couleur gagne s'il sauve la chaîne et perd si elle est capturée. L'utilisation par des objets de plus haut niveau du résultat de jeux sur des objets de bas niveau oblige aussi à réifier ces jeux.

## NOTATIONS

### Notation des objets reconnus sur le goban:

Nous employons une notation détaillée et une notation macroscopique pour représenter visuellement les objets reconnus sur le goban. Ces objets ont deux attributs, la couleur de l'objet, Blanc ou Noir, et l'ensemble des intersections sur lesquelles repose cet objet.
Un groupe en notation détaillée:

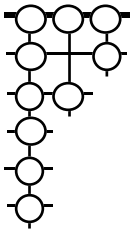


*figure 1*

Le même groupe en notation macroscopique:

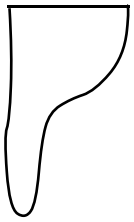


*figure 2*

*Les exemples présentés sont tous issus de la position initiale du diagramme 1.*

### Notations des propriétés ludiques des objets ou attributs d'objets

Elles sont inspirées de la théorie des jeux de Conway [Conway 82] et de son adaptation au jeu de Go [Berlekamp 94]

Définition statique d'un jeu:
Un jeu oppose deux joueurs A et B. Il possède des coups légaux pour A et B. Il peut être dans l'un des trois états statiques: "perdu", "gagné" ou "autre". Par convention, nous dirons que le jeu est "gagné" si A a gagné et "perdu" si B a gagné. Dans les autres cas, le jeu est "autre".

Etat dynamique d'un jeu:
Un jeu a un état dynamique qui peut prendre plusieurs valeurs:
"?" signifie que l'état dynamique du jeu est inconnu
">" signifie que le jeu est gagné pour A même si B commence le premier.
"<" signifie que le jeu est gagné pour B même si A commence le premier.
"*" signifie que le jeu est gagné pour celui qui commence.
"0" signifie que le jeu est perdu pour celui qui commence.

Dans la suite de l'article, pour une action X qui s'effectue sur un objet,
"X>" signifie que X sera atteinte même si l'autre joueur essaie de l'en empêcher,
"X<" signifie que X ne pourra plus être atteinte même si on essaie de le faire en jouant le premier,
"X*" signifie que X sera atteinte si on joue le premier et non atteinte si l'autre joueur joue le premier.

## DES CONCEPTS UTILES POUR COMPRENDRE UNE POSITION AU GO

Nous présentons dans ce paragraphe des concepts du niveau abstrait correspondant à celui des groupes. C'est le niveau le plus intéressant car c'est ce niveau qui permet d'interpréter une position de Go.
Nous montrons d'abord ce qu'est un "groupe" au sens de la connexion. Ensuite nous présentons les propriétés d'un groupe en deux parties : les propriétés intrinsèques et les propriétés interactives. Une propriété interactive d'un groupe synthétise à un premier niveau l'état des interactions de ce groupe avec ses voisins. Enfin, nous présentons la propriété de "santé" qui synthétise à un deuxième niveau l'état des différentes propriétés intrinsèques et interactives. Nous verrons que si l'état de la santé d'un groupe est inférieur à un seuil, le groupe est "mort" et une "catastrophe" se produit.
Chaque paragraphe présente un concept réifié dans Mood Indigo. La taxonomie et la description des classes se trouvent en annexe.

### Un concept central: le "groupe"

Le concept de "groupe" est le concept central du jeu de Go. Avec la connaissance des forces et des faiblesses des "groupes" de pierres, le joueur de Go interprète la position et décide du coup à jouer. Il est clair que nous ne pouvons décrire complètement ce qu'est un "groupe" pour un joueur humain tant cette notion est à la fois vague et complexe. Donc, dans cet article, le mot "groupe" est pris au sens du jeu de la connexion ou "pont" dont nous parlons ci-dessous.

Le pont

Un pont est une relation de connexion entre des chaînes de pierres.

exemples:

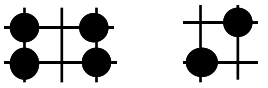
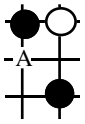
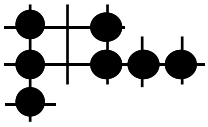
pont>:  pont*: 

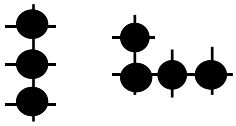

*figure 3*

Les deux ponts de gauche sont “>“ car si Blanc joue sur une des deux intersections vides, Noir se connectera en jouant sur l’autre. Le pont de droite est “*” car si Blanc joue le premier en A, Noir ne pourra plus se connecter et si Noir joue le premier, il jouera en A et sera connecté grâce au pont > du milieu de la figure.

Le modèle considère les ponts > comme des éléments qui permettent de fusionner les groupes entre eux. Par exemple, le groupe:



est construit par la fusion des deux chaînes:



par l’intermédiaire du pont>:

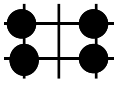


*figure 4*

Avec une notation macroscopique et à l'issue de cette initialisation des groupes de pierres, la position du diagramme 1 est la suivante:

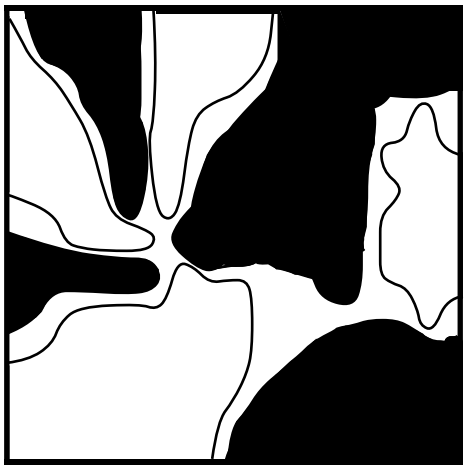


*diagramme 2*

## Une propriété intrinsèque: la base de vie

Les groupes définis au paragraphe précédent ont une propriété intrinsèque, la *base de vie*. La base de vie est un des attributs de la classe groupe.

La base de vie

La base de vie d'un groupe est constituée par l'ensemble de territoires contrôlés par ce groupe. Chaque territoire contribue à la base de vie suivant des heuristiques sur sa taille et sa forme [Davies 75a].

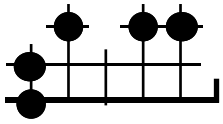
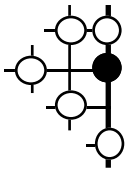
Une base de vie>:  et une base de vie< 

*figure 5*

## Des propriétés interactives, un premier niveau de synthèse.

Un groupe a des interactions amies avec des groupes amis synthétisées dans le concept d'*amitié*, des interactions ennemies avec des groupes ennemis synthétisées dans le concept d'*inimitié* et enfin des interactions vides synthétisées dans le concept de *vacuité*. L'amitié, l'inimitié et la vacuité sont des attributs de la classe groupe.

L'interaction amie

Une interaction amie relie deux groupes avec un ou plusieurs ponts*.

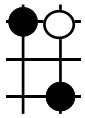
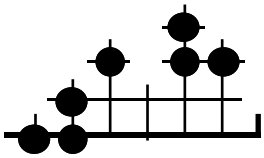
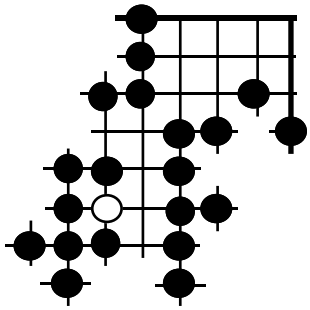
Le pont*  relie le groupe  et le groupe 

*figure 6*

L'amitié

L'amitié d'un groupe synthétise l'état de l'ensemble des interactions amies du groupe.
Si un groupe a plusieurs amis alors amitié>.
Si un groupe a un seul ami alors amitié*.
Si le groupe n'a pas d'ami alors amitié<.
Les deux groupes de l'exemple précédent ont une amitié*. Tous les autres groupes reconnus sur la position du diagramme 1 ont une amitié<.

L'interaction ennemie

Une interaction ennemie relie deux groupes ennemis avec les contacts. Une interaction ennemie n'a d'utilité que si les deux groupes ont une base de vie<. L'interaction ennemie contient alors l'information ludique qui spécifie quel groupe est le plus fort si un combat se déclenche entre les deux groupes.

Dans ce cas, il existe des heuristiques sur le nombre de libertés et la base de vie des deux groupes qui permettent de donner l'état de l'interaction ennemie [Davies 75b]. Nous donnons des exemples d'interaction ennemie du diagramme 1: interaction ennemie>, interaction ennemie<, interaction ennemie*, interaction ennemie0 ou interaction ennemie?

exemples:

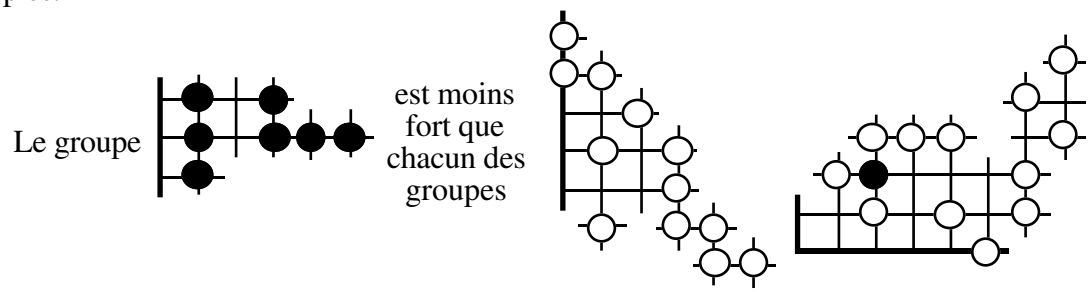



*figure 7*

L'interaction ennemie entre les groupes:

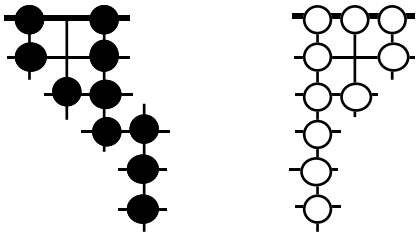


*figure 8*

est une interaction ennemie 0 (cf notations; en français, cela signifie qu'aucun des deux groupes ne peut attaquer l'autre sans se mettre en situation vulnérable).

L'inimitié

L'inimitié d'un groupe synthétise l'état de l'ensemble des interactions ennemies du groupe de la manière suivante:

Si au moins une interaction ennemie est * alors inimitié*.
Si pas d'interaction * et au moins une interaction ennemie est 0 alors inimitié0.
Si pas d'interaction * et pas d'interaction 0 et au moins une interaction ennemie > alors inimitié>
Si toutes les interactions ennemies sont < alors inimitié<.
Sinon interaction ennemie?

Le groupe noir de la figure 7 a une inimitié <.
Les groupes de la figure 8 ont une inimitié 0.

L'interaction vide

Une interaction vide relie un groupe et un espace vide.

La vacuité

La vacuité d'un groupe synthétise l'état de l'ensemble des interactions vides du groupe.
Si un groupe a plusieurs interactions vides alors vacuité>.
Si un groupe a une interaction vide alors vacuité*.
Si le groupe n'a pas d'interaction vide alors vacuité<.

## La santé: un deuxième niveau de synthèse

L'ensemble des propriétés d'un groupe sont synthétisées à un deuxième niveau dans la propriété de *santé*. La santé est un des attributs de la classe groupe.

La santé

La synthèse des propriétés intrinsèques et interactives du groupe est faite de la manière suivante:
Si base de vie> alors santé>
Si base de vie< et amitié< et inimitié< et vacuité< alors santé<
Si base de vie< et amitié< et inimitié0 et vacuité< alors santé0
Si base de vie* ou amitié* ou inimitié* ou vacuité* alors santé*
Sinon santé?

Les groupes:

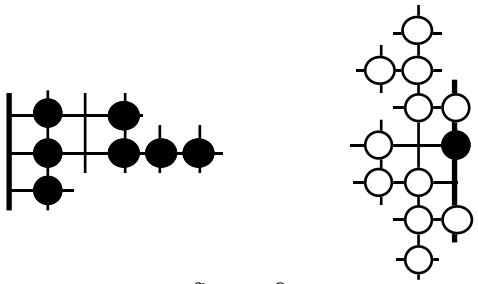
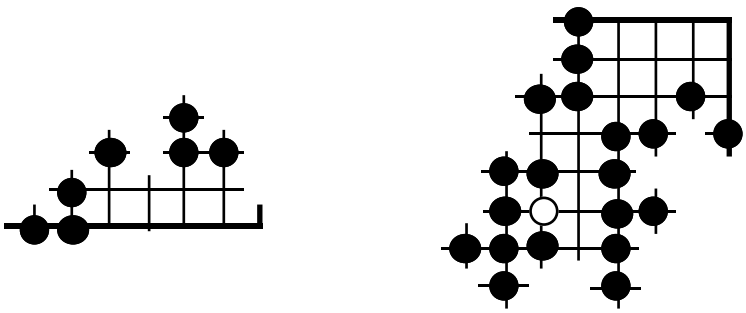


*figure 9*

ont une base de vie<, une vacuité <, une amitié< et une inimitié<, donc ils ont une santé<. Ils sont "morts" dans le jargon des joueurs de Go.

Les groupes:



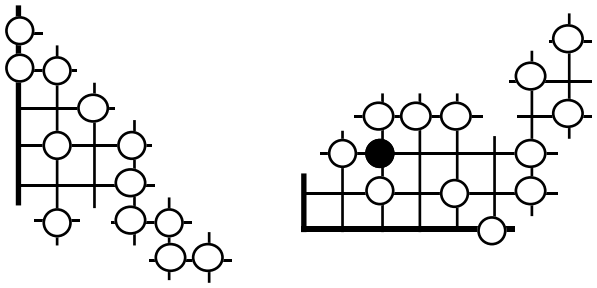


*figure 10*

ont une base de vie>, donc ils ont une santé>. Ils sont “vivants sur place” dans le jargon des joueurs de Go.

Les deux groupes de la figure 10 ont une base de vie<, un encerclement>, une amitié< et une inimitié0 donc ils ont une santé0. Ils sont “vivants par seki” dans le jargon des joueurs de Go.

**Si la santé d'un groupe est négative, une catastrophe se produit.**

Quand une catastrophe se produit, les groupes ennemis "mangent" le groupe mort en fusionnant autour de lui pour former un groupe plus gros. Nous avons utilisé le terme "catastrophe" parce qu'à ce moment de la partie, la description du goban, le score de la partie (et la psychologie des deux joueurs !) subissent une discontinuité sensible.

Exemples:

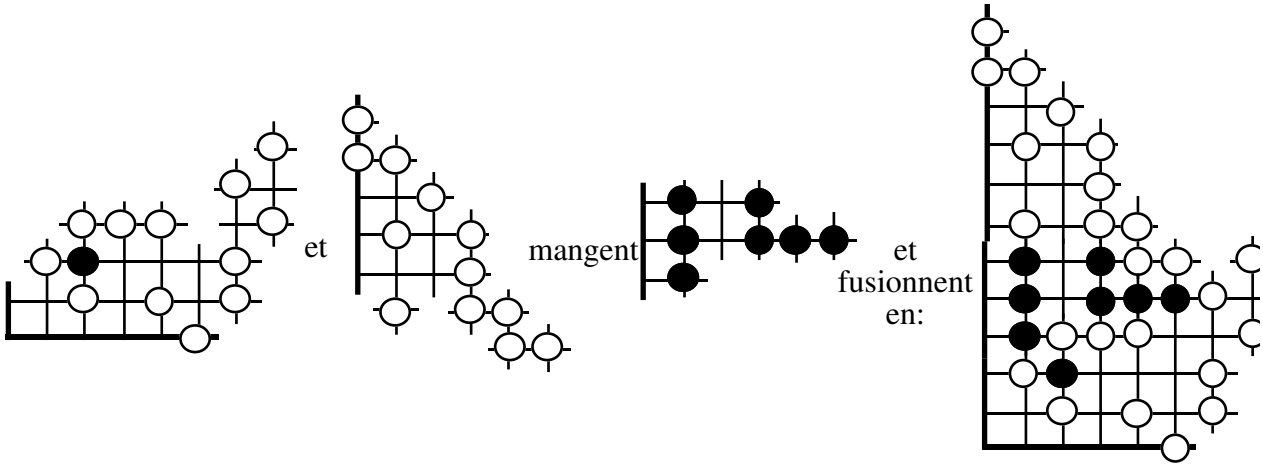



*figure 11*

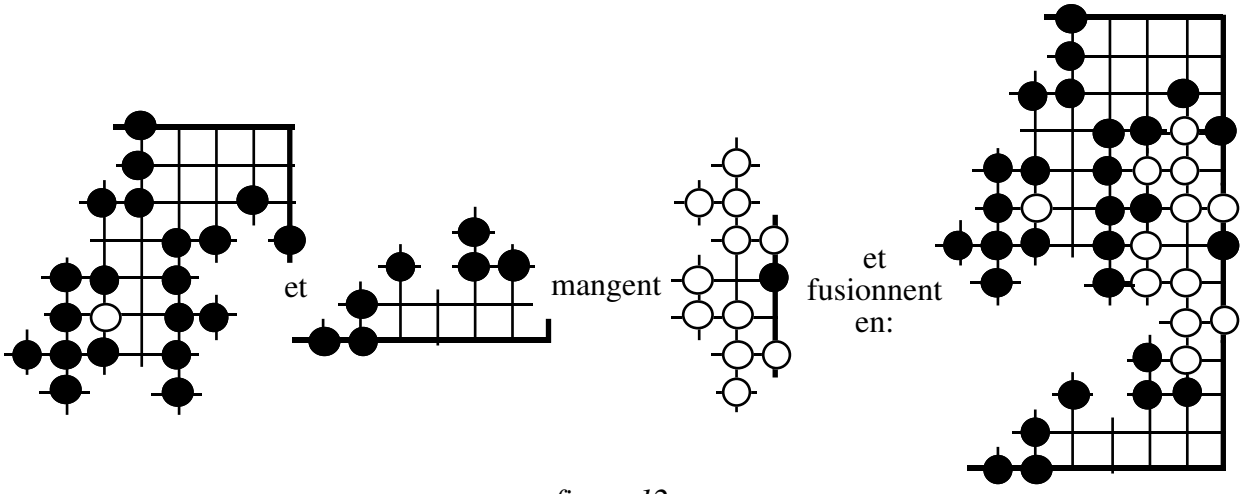



*figure 12*

Les deux catastrophes peuvent être aussi décrites en notation visuelle macroscopique:

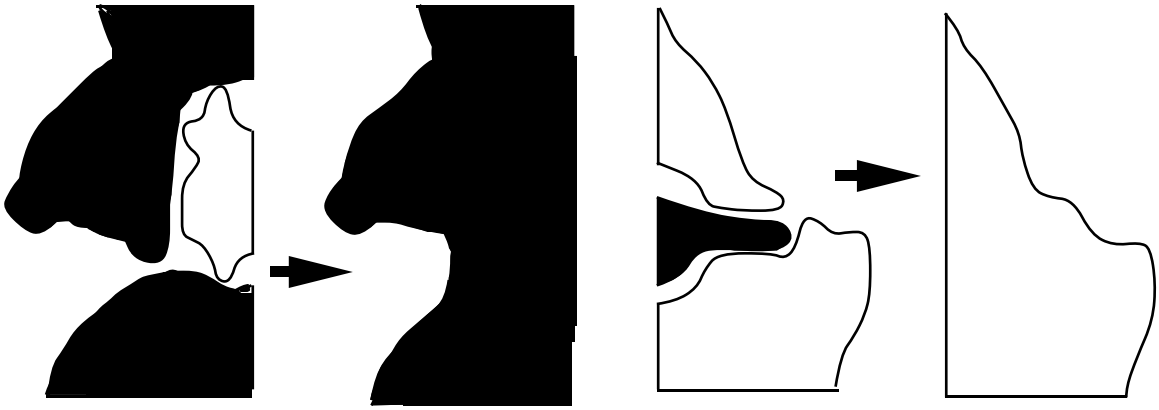


*figure 13*

Tant que des catastrophes se produisent le modèle fait fusionner les groupes. Le nombre de groupes diminue à chaque catastrophe pour s'arrêter lorsqu'aucun groupe n'a de santé<.
Après le déclenchement des catastrophes, l'interprétation du goban est la suivante:

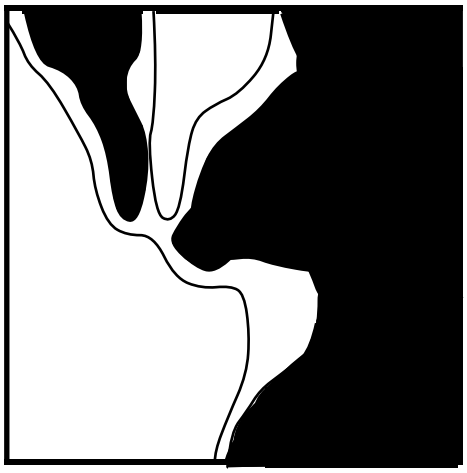


*diagramme 3*

## CONCLUSION

Nous présentons les difficultés de modélisation que nous avons rencontrées et une analogie entre le jeu de Go et certains systèmes naturels étudiés en Physique. Enfin, nous donnons l'état actuel du système Mood Indigo.

**La principale difficulté a été d'identifier et de placer les concepts les uns par rapport aux autres.**

La clarté recherchée pour présenter le modèle cache les difficultés rencontrées pour y aboutir.

La première difficulté pour élaborer un modèle du jeu de Go a été d'amorcer le processus de modélisation. Nous avons mis dans le modèle quelques connaissances de base dont parlent les joueurs de Go. Nous avons implémenté ce premier modèle. Nous avons comparé les résultats du modèle avec ceux de joueurs de Go. Si les résultats du modèle concordaient avec ceux des joueurs humains nous avons supposé que le modèle était valide (Ce qui ne voulait évidemment pas dire que le fonctionnement interne du modèle était le même que celui du joueur humain!), sinon nous avons modifié le modèle pour que les résultats concordent. Nous avons répété ce processus un grand nombre de fois.

*Par approximations successives nous avons remplacé, supprimé, créé et affiné les concepts et surtout nous les avons replacés les uns par rapport aux autres pour arriver à une structure qui permette d'exprimer simplement les règles qui animent les objets.*

Au départ nous ne pensions pas explicitement implémenter des concepts tels que: amitié, inimitié, vacuité, santé, ou catastrophe. C'est le souci de modularité et l'obligation de pouvoir modifier facilement notre programme qui nous a amené à cette découpe des concepts.

**Une similarité entre deux échelles de grandeur ou comment la physique conforte notre modèle**

Voici une figure quelconque, courante au jeu de Go:

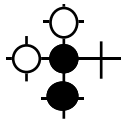


*figure 14*

A première vue, nous voyons des pierres de Go. Mais que se passe-t-il si nous regardons la figure 14 à l'échelle des groupes, c'est-à-dire si nous imaginons que chaque rond représente non pas une pierre mais un groupe de pierres ?

A cette échelle, le trait qui relie deux ronds représente alors une interaction entre deux groupes. Nous identifions alors des groupes , des interactions amies , des interactions ennemies et des interactions vides comme dans notre modèle. La figure 14 peut être vue indifféremment à l'échelle des pierres ou à l'échelle des groupes. Nous observons donc une similarité entre deux échelles caractéristiques du jeu de Go : l'échelle des pierres et l'échelle des groupes. Une propriété caractéristique de nombreux systèmes physiques ou naturels soumis à des contraintes est d'évoluer vers des états critiques où l'on peut observer des similarités entre différentes échelles [Wilson 75].

*Cette similarité nous conforte dans l'idée que les concepts identifiés par notre méthode expérimentale d'approximations successives sont ceux que nous devions identifier.*

## Intégration du modèle dans notre programme de Go MOOD INDIGO

Ce modèle est utilisé dans le programme de Go Mood Indigo écrit en C++ qui joue des parties complètes sur 19-19. Mood Indigo joue un coup entre cinq secondes et une minute sur des Sun SparcStation10. C++ offre l'avantage d'être rapide à l'exécution même si ses "fonctionnalités" objet sont assez rudimentaires.

Depuis six moix, Mood Indigo a joué plus de cinquante parties avec Gogol, un autre programme de Go développé actuellement dans notre équipe par Tristan Cazenave. Ces nombreuses parties ont permis de détecter et corriger de nombreuses faiblesses, de valider notre modèle et d'améliorer considérablement le niveau de ces deux programmes.

Mood Indigo a aussi été testé contre les deux meilleurs programmes américains jouant sur IGS, le serveur international de Go. Contre Many Faces of Go (16ème kyu), il a joué une partie équilibrée et perdu deux parties. Contre Poka (18ème kyu), il a joué une partie équilibrée. Nous pensons que le niveau et le style combatif de Mood Indigo sont essentiellement dûs au modèle que nous avons présenté dans cet article.

Mood Indigo a joué également contre de nombreux joueurs humains. La nature chaotique des parties de Go rend difficile l'estimation des niveaux faibles. Cependant, nous estimons que le niveau de Mood Indigo est de 20ème kyu.

## Remerciements

Nous remercions Jacques Pitrat pour la direction de cette recherche, Valérie Sion pour ses conseils indispensables, Tristan Cazenave pour son programme de Go Gogol, Patrick Ricaud, Zahia Guessoum, Mohammed Ramdani, Maria Rifqi et Hugh Bellemarre pour leurs multiples relectures de cet article.

## REFERENCES

[Berlekamp 91] - E. Berlekamp - Introductory overview of mathematical go endgames - Proceedings of symposia in applied mathematics - 43 - 1991

[Berlekamp 94] - E. Berlekamp, D. Wolfe - Mathematical Go: Chilling gets the last point - A. K. Peters - ISBN 1-56881-032-6 , 1994

[Boon 91] - M. Boon - Overzicht van de ontwikkeling van een Go spelend programma - Afstudeer scriptie informatica onder begeleiding van prof. J. Bergstra - 1991

[Bouzy Victorri 92] - B. Bouzy, B. Victorri - Go et Intelligence Artificielle - Revue de l'AFIA n. 10 - Juillet 1992

[Chen 89] - K. Chen - Group identification in Computer Go - Heuristic programming in Artificial Intelligence 1 - D.N.L. Levy &Beal D.F. (éds) Ellis-Horwood Prenctice-Hall 1989

[Conway 82] - J. Conway, E. Berlekamp, R. Guy - Winnings ways - tome 1 - Academic press - 1982

[Davies 75a] - J.Davies - Life and death - The Ishi Press, Tokyo - 1975

[Davies 75b] - J.Davies - Tesuji - The Ishi Press, Tokyo - 1975

[Ericsson] - K.A. Ericsson H.A. Simon - Protocols analysis, verbal reports as data - MIT Press - Cambridge, Massachussetts, London, England

[Hamann 85] - Hamann C.M. - Chronologie der programierung des japonischen Brettspiels Go, eine herausforderung an die Kunstliche Intelligenz - Angewandte informatik, 27,12, 1985

[Hsu 90] - F.Hsu, T.Anantharaman, M.Campbell, A.Nowatzyk - A Grandmaster Chess Machine - Scientific American, vol. 263, n.4 - October 1990

[Vermersch91] - Vermersch P. - Questionner l'action: l'entretien d'explicitation - Psychologie française, numéro spécial "Anatomie de l'entretien", 35-3, 227-235 - 1991

[Wilson 75] - K. G. Wilson - The Renormalization Group: Critical Phenomena and the Kondo Problem - Reviews of Modern Physics- vol. 47 - n°4 - pp-773-840 - 1975

[Wolf 92] - T. Wolf - Tsumego with Risiko - Workshop on Computer Go - Cannes 1992

## ANNEXE: LA TAXONOMIE DES CLASSES ET LEUR DESCRIPTION

Une présentation exhaustive des méthodes associées à chaque classe serait trop lourde. Aussi nous ne les présentons pas. Cependant, il est évident que la factorisation des méthodes dans les classes les plus générales a guidé la découpe des classes de la taxonomie ci-dessous.

### La taxonomie des classes

Chaque tabulation symbolise la relation d'héritage.

*Objet-reconnu*
- *Groupe*
- *Jeu*
  - *Attribut-Groupe*
    - *Attribut-Intrinseque*
      - *Base-De-Vie*
    - *Attribut-Interactif*
      - *Amitié*
      - *Inimitié*
      - *Vacuité*
    - *Attribut-Synthétique*
      - *Santé*
  - *Interaction*
    - *Interaction-Amie*
    - *Interaction-Ennemie*
    - *Interaction-Vide*

### La description des classes

*Objet-reconnu*
- *couleur*
- *ensemble-intersections*

*Jeu : Objet-reconnu*
- *résultat*
- *ensemble-coups-attaque*
- *ensemble-coups-défense*

*Groupe : Objet-reconnu*
- *base-de-vie*
- *amitié*
- *inimitié*
- *vacuité*
- *santé*

*Base-De-Vie : Attribut-Intrinseque*
- *ensemble-territoires*
- *ensemble-yeux*

*Attribut-Interactif : Attribut-Groupe*
- *ensemble-interactions*

*Interaction : Jeu*
- *groupe-un*
- *groupe-deux*

ERROR: undefined
OFFENDING COMMAND: ÿ

STACK: