自己組織系としての計算システム
— ソフトウェア研究への 2 つの提案 —

金田　泰　　(日立製作所中央研究所)*

概要：**ソフトウェアを開発するのは不完全な人間であり，また今日では人間社会と密に結合された大規模・複雑な開放系が開発されていることをかんがえると，閉じた "完全な計画" を要求する従来のソフトウェア開発法や理論にかわる自己組織的な計算システムの開発法や理論を構築することが必要である．そこで，この報告では「プログラムなしの計算をめざそう」，「計算システムを自己組織系としてみよう」というソフトウェア研究への 2 つの提案をする．これらの提案の実現に必要なミクロ・モデルとマクロ・モデルについて説明し，前報でしめした計算モデル CCM を前者の例として位置づけ，後者の例として計算を確率過程としてみるマルコフ連鎖モデルをしめす．**

1. はじめに

従来のソフトウェア開発法やそのための理論は，基本的には "完全な計画" すなわち完全なプログラムや完全な仕様が記述可能であることを前提にしていたといってよいであろう．しかし，金田 [Kan 92] でものべたように，このような方法や理論には 2 つの問題がある．

第 1 は，人間は不完全な存在だということをわすれているという問題である．すなわち，プログラム開発につかわれるプログラミング言語などのトゥールは人間のわずかなあやまりすなわちミスタイプやバグをも許容しないし，論理にもとづくプログラムの理論はひとつでもバグのあるプログラムに対して「無意味」という烙印を押してしまう．対象が小規模なプログラムならばこのようなトゥールや理論でも十分やくにたつだろうが，複雑な大規模ソフトウェアにおいてはそうはいかない．人間が不完全な存在だということをみとめるならば，人間がおかすあやまりについて，ソフトウェア開発トゥールはもっと配慮すべきであり，そのためにはすくなくともなんらかのかたちで人間の不完全さを理論のなかであつかえるようにしなければならない．

第 2 は，開放系 (open system) については本質的に完全な仕様の記述ができないという問題である．構成要素が決定論的に動作し，あらかじめきめられた範囲の入力だけがあつかわれる閉鎖系 (closed system) については，原理的には完全な仕様を記述することができる．しかし，人間社会や自然をシステムの一部としてくみこんでいるシステムや，それとインタフェースされている開放系においては，そうではない．なぜなら，人間や自然は非決定論的・非合理的に動作するために，予測できない部分がのこるからである．

金田 [Kan 92] は上記のような問題から出発し，計算システムの自己組織化によってこれらの問題を解決しようとかんがえた．そこで，まずその第 1 歩として「自己組織系とはなにか」をかんがえた．そして，自己組織系の記述をめざした計算モデルとして化学的キャスティング・モデル (chemical casting model, CCM) を提案し[注1]，例題をしめした．しかし，金田 [Kan 92] においてはソフトウェアに関する問題や自己組織化とモデルとのあいだにあるギャップをうめることができなかった．すなわち，CCM にもとづいてどのようにソフトウェアを開発するのか，どのようにしてどのような自己組織化をめざすのか，どうすれば問題の解決につながるのか，などがあきらかにされないままだった．この報告はこのようなギャップを部分的にうめることをめざしている．

自己組織的な計算システムを実現するまでのみちのりはとおい．この目的のためには CCM だけでなく，さまざまな面からの研究が必要だとかんがえられる．そこで，この報告ではソフトウェア研究への 2 つの提案というかたちで主張をのべることにする．すなわち，「プログラムなしの計算をめざそう」，「計算システムを自己組織系としてみよう」という提案である．これらの提案の実現に必要な 2 つのモデルすなわちミクロ・モデルとマクロ・モデルについて説明し，CCM をミクロ・モデルの例として位置づける．また，マクロ・モデルの例として，計算を確率過程としてみるマルコフ連鎖モデルをしめす．

Computational Systems as Self-Organizing Systems,
Central Research Laboratory, Hitachi, Ltd.,
Yasusi Kanada
* 現在，新情報処理開発機構つくば研究センタに出向中．

[注1] 金田 [Kan 92] はこのモデルを化学的プログラミング・モデル (chemical programming model, CPM) とよんでいた．

2. 提案 1「プログラムなしの計算をめざそう」

2.1 提案の内容と提案理由

最初の提案は，プログラムなしの計算をめざそう，あるいはプログラムしないようにしようということである．その意味は，いわゆる “プログラム” というものをつかわずに，人間がもとめるなんらかの結果をみちびく “計算” ができるようにしよう，つまり計算の完全な計画すなわち決定論的なプログラムをつくらずにコンピュータをふくむシステムをつかえるようにしようということである[注2]．ただし，いますぐプログラムなしの計算ができるわけではないから，そのような方向にむけて研究しようということである．

なぜ上記のような提案をするかというと，その理由はすくなくとも 3 つある．第 1 の理由は，すでにのべたように完全な計画すなわち仕様やプログラムはそもそも記述不可能だということである．すなわち，まず，バグのないプログラムをつくることは，しょせん人間には無理だとかんがえられる．もちろん，明確な仕様があたえられている問題をとくための小規模なプログラムであれば，完全なプログラムをつくることは夢ではない．しかし，1 万ステップ程度のプログラムにおいてもそれはすでに困難である．また，オペレーティング・システムやオンライン・システムをはじめとする開放系である大規模ソフトウェア・システムにおいては完全な仕様を記述することが本質的にできない．そのような不可能なことを技術の進歩によって実現することをめざすよりも，不完全な計画からでもうまく計算できるようなしかけをつくることのほうが，より現実的な目標ということができるだろう[注3]．

プログラムなしの計算を提案する第 2 の理由は，決定論的な “プログラミング” では “どうでもよい” ことまできちんときめなければならないが，このようなむだな努力はさけたいということである．例として，LSI の配線プログラムの開発についてかんがえてみよう．配線プログラムの開発においては，そのプログラムが配線する順序を，可能なさまざまな順序のなかからえらんできちんときめなければならない (**図 1** 参照)．基本的な配線法がおなじでも配線順序によって最適解にどれだけちかづけるかが変化するから，それはまったくどうでもよいこととはいえない．しかし，どのような順序で配線するのが最適であるかはわからないままにプログラムをつくるから (あるいは，つくるとすれば)，その意味では配線順序は “どうでもよい” ことだといえる．現在のプログラミングにおいては，本質的なことだけでなく，このような “どうでもよい” ことまできちんときめなければならず，わずらわしい．

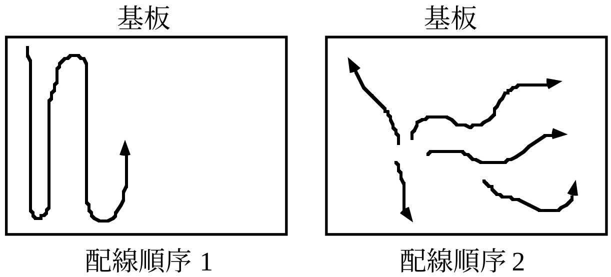


図 1　LSI 配線におけることなる配線順序の例

プログラムなしの計算を提案する第 3 の理由は，“プログラミング” は計算システムの**自己組織化**能力をころしているとかんがえられるということである．「自己組織化」ということばを生物における自己組織化たとえば生物の発生過程における形態形成のようなはたらきという意味でつかうとすれば，計算システムが自己組織能力をもちうるという点に関しては異論がおおいとかんがえられる．しかし，著者は生物だけでなく非平衡熱力学系などにおいても自己組織化がおこっている [Pri 77, Pri 84] のだから，それがコンピュータにおいておこりえない理由はないとかんがえている．そして，コンピュータにおいて自己組織化をおこすためには，(決定論的な) “プログラミング” をやめることが必要だとかんがえる．なぜなら，決定論的な世界では自己組織化はおこらないからである．なぜなら，自己組織化というのは**自発的・自律的**な行為であり，非決定性があることが必須だからである [Kan 92]．決定論的な世界でおこる組織化は他者によって完全に制御されたものであり，したがって自己組織化ではない．コンピュータのばあいには，間接的ではあるがプログラムによる決定論的な組織化を制御しているのは人間である．

注2 上記のことからわかるように，ここでいう “計算” はそれじたいが非決定論的なものである．“計算” は基本的には人間の要求や希望をみたすようにおこなわれるが，プログラムによって完全に制御されているわけではないので，かならず人間の希望どおりになるとはかぎらない．この問題への対策については後述する．

注3 現在われわれが手にしている道具をつかっても，この目標をある程度実現することはできる．西垣 [Nis 88, Nis 90] はソフトウェア工学や AI が合理的なものをもとめながら，実際には人間の非合理性，あいまいさ，情念などを反映したソフトウェアがつくられていることを指摘している．しかし，“もっとうまくやる” ためには，“不完全な計画” にもとづく計算の理論が必要だろう．

この点について，さきほどの LSI 配線問題を例にとって，もうすこしかんがえてみよう．配線の順序をプログラマがきめることは単にプログラマにとってわずらわしいだけでなく，よりよい順序を排除することによって自己組織的な最適化の余地を排除してしまっている．その順序がプログラマが熟考をかさねることによってはじめてえられたものならば自己組織化によってよりよい順序がみいだされる可能性はひくいだろうが，“どうでもよい” こととしてきめられたならばその可能性はたかいといえるだろう．“プログラミング” は，このような自己組織的な最適化の可能性をころしてしまっている．

2.2 提案実現に必要な 2 つのモデル

ところで，プログラムなしの計算の価値をみとめるとしても，すでにのべたようにそれはただちに実現可能ではない．したがって，それをどうやって実現するかが問題になる．そのためには，著者はつぎのような 2 つのモデルを確立するとともに，それらをうまくむすびつけることが必要だとかんがえる．

第 1 のモデルは**ミクロ・モデル**である．これは，部分的な知識だけでソフトウェアが実行できる，あるいはシステムをうごかすことができるしかけ，すなわち計算モデルである．計算システムは人間がつくるものであるから，それを動作させるためにこのようなモデルをつくることが必要であることをとくに説明する必要はないだろう．ただし，ここで “計算モデル” といっても，“計算” の概念が従来の “完全な計画” にもとづく計算とはちがっているから，従来の意味の計算モデルとはちがっている．ところで，この「部分的な知識だけで」ということばはさまざまな意味に解釈しうるが，ここでは「非決定的な部分をふくんでいても」という意味だと解釈する．著者の研究においてはミクロ・モデルとして化学的キャスティング・モデルを使用しているが，それについては第 3 章でのべる．

第 2 のモデルは**マクロ・モデル**である．これは，システムがのぞみどおりに動作しているかどうかを**観測**し，**制御**するしかけである．マクロ・モデルが必要な理由は，ミクロ・モデルは不完全なモデルであるから，それだけでは計算が人間の意図したとおりにおこなわれる保証がないからである．意図がうまく実現されたかどうかは観測によって確認されるべきである．また，観測の結果として計算が意図したとおりにおこなわれていないことがあきらかになれば，それをのぞましい方向に修正するために制御することが必要になる．著者の研究においてはマクロ・モデルとして確率過程にもとづくモデルを使用しているが，それについては第 4 章でのべる．

これらの 2 つのモデルは，すでにのべたことからもある程度わかるように相補的，すなわち両方とも不可欠である．相補的であるという理由を 3 つあげることにする．第 1 の理由は，システムのミクロな状態のなかには観測と制御には不要な情報すなわち本質的な情報をとらえるうえではじゃまになる微細な情報がおおいから，観測と制御のためにミクロ・モデルをつかうことはできず，マクロ・モデルが必要だということである．

第 2 の理由はより重要だが，それはミクロ・モデルからはシステムのマクロな状態やふるまいは予測できないから，ミクロ・モデル以外にマクロ・モデルが必要だということである．計算システムとの直接の関係はないが，物理学から例をとると，物理学にはミクロな世界とマクロな世界とをつなぐ統計物理学という分野がある．そこではミクロ・モデルとしては力学，マクロ・モデルとしては熱力学があり，統計力学がこれらをつないでいるということができるだろう．マクロ・モデルである熱力学におけるマクロな量たとえばエントロピーは多数の分子の関係として定義されるから，個々の分子のうごきを説明するミクロ・モデルである力学によっては説明されない．また，熱力学における第 2 法則すなわちエントロピー増大の法則が力学だけからはうまく説明できないことはよくしられた事実である．

第 3 の理由は，マクロ・モデルだけではシステムの詳細な動作を決定できないということである．これは，マクロ・モデルにおいては捨象されている部分があるからあきらかであろう．

ところで，物理・化学系などの自己組織化の理論として Haken らによるシナジェティクス [Hak 78, Hak 83] や Prigogine らによる散逸構造理論 [Pri 77] などがある．これらの理論はもともと計算システムには直接関係がないが[注4]，これらの理論においてもミクロなレベルとマクロなレベルとがあつかわれ，それらの “からみあい” から自己組織化がおこるこ

[注4] Haken らはシナジェティクスの計算システムへの適用もかんがえている [Hak 83] が，その対象はおもに並列処理におけるハードウェアの自己組織化である．この研究がめざす記号処理における自己組織化のためには，Haken らのとは質的にちがう理論が必要だとかんがえられる．なぜなら，自己組織化対象がおかれる空間の構造がまったくことなる (たとえば Haken らがあつかっているのは連続系であり，われわれの対象は離散系である) からである．

とがしめされている．また清水 [Shi 90, Shi 92] は生物における自己組織化を理論化しようとしているが，生物においてはこのような 2 つのレベルのあいだのフィードバックとフィード・フォワードによってつくられる “ホロニック・ループ” が多層的に存在し，自己組織化をおこなっていると主張している．計算システムにおいても，定性的にはこのようなミクロなレベルとマクロなレベルとのあいだのからみあいから自己組織化がおこるといえるであろう．

これらの 2 つのモデルをうまくくみあわせることによって，適応的な計算システムを構成することが将来の研究目標になるとかんがえられる．この点については次章でもうすこしくわしくのべる．

3. 提案 2「計算システムを自己組織系としてみよう」

3.1 提案の内容と提案理由

第 2 の提案は，“計算” をおこなうシステムを自己組織系として，あるいは単にシステムとしてみようということである．つまり，コンピュータをふくむ “システム” やソフトウェアをシステム論的にとらえようということである．計算をおこなうのが計算システムまたはコンピュータ・システムであるならば，それをシステムとしてみるのは当然のことだとみえるであろう．しかし，コンピュータ科学においては，通常, システムということばがサイバネティクス [Wie 61] や一般システム理論 [Ber 68] などのシステム理論におけるような意味ではつかわれていない．そのことは，これらのシステム理論において非常に重要なフィードバックの概念がコンピュータ科学においてはつかわれないことからもわかるであろう[注5]．

計算をシステムとしてみようという提案をする理由は 2 つある. 第 1 の理由は上記の提案のなかの「自己組織」の部分には直接関係がないが，計算の観測と制御のための基礎理論はシステム理論にほかならないということである．第 1 の提案における “プログラミング” によらない “計算” においては，計算の観測と制御のためのしかけが必要であるということはすでにのべた．計算プロセス (過程) の観測と制御のための理論というのはまだ存在しないとかんがえられるが，工業プロセスやその他のさまざまなプロセスの観測と制御のための理論はすでにあり，その基礎となっているのはシステム理論である．したがって，計算プロセスの観測と制御のためにもシステム理論的なもののみかたが基礎になり，とくに前述のマクロ・モデルは計算のシステム理論からうみだされるとかんがえられる．ミクロ・モデルによって支配されたプログラムによって部分的に制御されたシステムは，それだけでは人間の意図する計算をおこなわないかもしれない．しかし，さらにマクロ・モデルにもとづいて観測とその結果にもとづく制御を外部からうけることによって，意図されたとおりの計算をおこなうことができると期待される[注6]．

計算の観測と制御において重要なことのひとつは (自己) 安定性である. システム哲学者 Laszlo [Las 72] は自然のシステムの重要な特性のひとつは自己安定性だとしているが，システム理論にもとづいてつくられた人工のシステムたとえば工業プロセスの制御システムなどにおいても自己安定性は不可欠である. ところが，計算システムには自己安定性がないとかんがえられる．些細なバグが致命的な結果をまねくのは，この性質がないためだとかんがえられる．自己安定性がないのは，計算システムがシステム理論にもとづいておらず，システム理論においては常識となっている負のフィードバックといった概念を適用することができないためだとかんがえられる．

第 2 の理由は上記の提案のなかの「自己組織」の部分に直接関係するが，これまでころされていた自己組織化能力を計算システムに期待するならば，計算システムを自己組織系としてみることが重要だということである．Laszlo は自然のシステムの重要な特性のひとつは自己組織性だとしている [Las 72][注7]. 閉鎖系においては，あらかじめ最適な点をもとめておいて，システムの状態がそこからはずれれば負のフィードバックによって最適点にもどすようにすればシステムをのぞましい状態にたもつことができる. しかし，開放系においては外部環境からどのような入力や間接的な影響があるかがわからないため，あらかじめどこが最適点になるかがわからない．したがって負のフィードバックによる制御だけではシス

[注5] 西垣 [Nis 88] はコンピュータの性能の制御に自動制御理論をつかおうとしたが失敗した，プログラムとはフィードバックのきかないシステムなのだとかいている．

[注6] このばあい，もしもマクロ・モデルがまちがっていれば意図された計算をおこなうことはやはりできないが，マクロ・モデルのほうがミクロ・モデルより直接的に人間の意図を反映したものであれば，このような可能性ははるかにすくないとかんがえられる．

[注7] Laszlo [Las 72] は，ここでとりあげた自己安定性と自己組織性のほかに自然のシステムがもつ特性として全体性 (非還元性) と階層性をあげている．彼の理論は計算のシステム理論を構築するうえでおおきな示唆をあたえている．

テムをのぞましい状態にたもつことができない．たとえば，システムがこれまでとどまっていた安定点からゆらぎによってはずれることによってよりよいものにめぐりあったときには，ゆらぎに対して正のフィードバックがかかるような機構 (自己組織性のひとつのかたち) が必要であろう．生物や社会システムなどの自然のシステムは，このような状況に対処できる自己組織性をもっている．計算システムにおいても，それが開放系であるならば，なんらかのかたちで自己組織性をもつことは必須であろう．現在はシステムの構成要素としての人間が自己組織化すなわちシステムの改善や再構築をおこなっているが，それだけでは十分には対処できない．計算システムを構成するソフトウェアにも自己組織化の能力がそなわっているべきであろう．このように外部からの入力や影響に適応して動作する計算システムを**適応的な計算システム**とよぶことができるだろう．

すでにのべたように，自己組織化はミクロなレベルとマクロなレベルのからみあいからおこる．それを説明するのが計算のシステム理論のはずである．すなわち，計算のシステム理論は物理，生物，社会学などの自己組織化の理論からまなんで，前述のミクロ・モデルとマクロ・モデルのあいだにおこりうるからみあいをあきらかにし，適応的な計算システムを構成するための基礎をあたえるであろう．

3.2 自己組織系のひとつのモデル

計算のシステム理論はまだどのような理論になるのかほとんどわからない．しかし，この理論をつくっていくための作業仮説として，自己組織系のひとつのモデルをしめそう．このモデルは金田 [Kan 92] がすでに提案しているものだが，ここでも必要最小限の説明をくわえておく．

自己組織系の基本モデルとして**図** 2 のようなものをかんがえることができる．このモデルがあらゆる自己組織系にあてはまると主張することはできないだろうが，われわれが対象としようとしている情報の自己組織化をおこなうシステムをはじめとして，散逸構造をうみだす熱力学系やその他のさまざまな自己組織系のモデルとなっているとかんがえられる．

このモデルにおいて，システムは時間とともに変化する．システムに対して，秩序の指標としての**エントロピー**または (大域) **秩序度**が定義されている．エントロピーが定義されているばあいには，それは時間とともに減少する．秩序度が定義されたばあいには，それは時間とともに増加する．熱力学系のばあいには，第 2 法則をみたしつつエントロピーを減少させるためには，エントロピーを外界にすてなければならない．したがって，システムは**開放系**でなければならない．このような法則が存在しないシステムにおいては，(この要請だけをかんがえるかぎりは) かならずしも開放系でなくてもよいであろう．

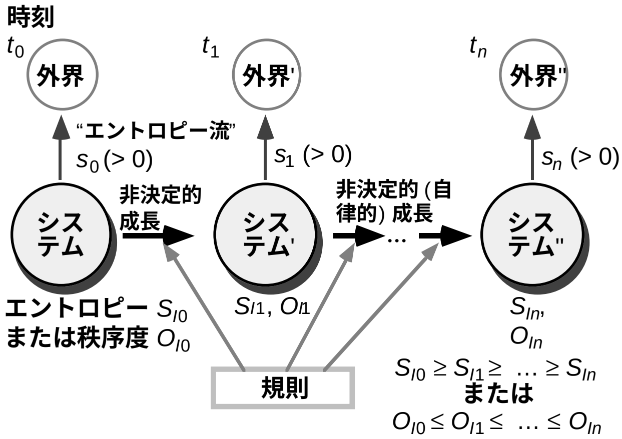


図 2　**自己組織系のモデル**

また，ここにはシステムの変化のしかたを支配する規則あるいは法則が存在するはずである．ここで規則あるいは法則といっても，それはシステムの動作を決定論的にきめるものではない．この規則・法則が課する制約のもとでもシステムは非決定的・自律的に動作し，自己組織化する．したがって，規則・法則にもとづく動作という面を中心にとらえれば自己組織系を動力学系 (dynamical system) としてとらえることができるだろうし，非決定的な動作という面を中心にとらえれば確率過程としてとらえることができるだろう．散逸構造をうみだす熱力学系のような自己組織系を解析するには，動力学系としての面と確率過程としての面を両方とらえる必要があることがしめされている [Pri 77, Hak 78].

4. ミクロ・モデルの例 — 化学的キャスティング・モデル (CCM)

この章では，前章でしめした 2 つの提案を具体化するために必要なミクロ・モデルとして，化学的キャスティング・モデル (chemical casting model, CCM) について説明する．このモデルは金田 [Kan 92] で提案した化学的プログラミング・モデルを改名したものだが[注8]，ここでは，[Kan 92] において明確に説明しなかったことに重点をおいて説明する．ミクロ・

[注8] Programming ということばを追放するため，それにかわることばとして casting をえらんだ．Cast という単語には「さいころをなげる」，「計算する」，「配役をきめる」などの意味があり，よりふさわしいとかんがえた．

モデルとしてはもっとべつのものもかんがえられるが，現在のところこのモデルがもっとも上記の提案で要請されているものにちかいとかんがえられる．

4.1 CCM の概要

CCM は化学反応系とのアナロジにもとづく計算モデルである．どのようなアナロジにもとづいているかは，以下の説明でもわかるだろうが，くわしくは金田 [Kan 92] を参照されたい．CCM は不完全な (非決定的な) 計画にもとづく計算のためのミクロ・モデルであり，前章でのべた自己組織系のモデルにももとづいている．

CCM の構成要素についてかんたんに説明する．CCM は chemical abstract machine [Ber 90] と同様にプロダクション・システムにもとづくモデルである．プロダクション・システムにおける作業記憶は CCM においても**作業記憶**とよぶ．すなわち，CCM が作用するデータは作業記憶にふくまれる．そして，プロダクション・システムにおける規則ベースすなわちプログラムに相当するものをキャスタとよぶ．CCM は不完全な計画にもとづく計算のためのモデルなので，完全な計画を意味するプログラムということばのかわりに，**キャスタ**ということばを使用する．キャスタの自己組織化すなわち自律的なかきかえというのも興味ある問題だが，いまのところはキャスタはユーザによって記述され，そのままのかたちでつかわれるとかんがえる．したがって，当面は自己組織化の対象となるのはデータだけである．

作業記憶にふくまれるべき**オブジェクト**あるいは**データ**としては，つぎのようなものがある (**図** 3 参照)．**原子**はデータの単位であり，内部状態をもつ．原子にはデータ型があり，それを**元素**ともよぶ．原子どうしを**リンク**によって結合することができ，結合された全体を**分子**とよぶ．リンクは無向でも有向でもよい．無向のリンクは化学結合に似ているが，化学結合には有向のリンクに相当するものはない．また，リンクにはラベルをつけることもできる[注9]．分子どうしをリンクのようなもので結合した超分子あるいは超超分子のような階層構造をかんがえることもできるが，いまはかんがえない．

キャスタは反応規則と局所秩序度とで構成される．**反応規則**はシステムの局所的な (ミクロな) 変化のしかたをきめる規則であり，ユーザにより定義される．ここで「局所的」ということばは，その反応規則によって参照される原子数がすくないということを意味する[注10]．反応規則は前向き推論によるプロダクション規則として記述される．したがって，つぎのようなかたちをしている．

$$\text{LHS} \rightarrow \text{RHS}.$$

反応規則は化学反応式に相当するものだといえる．反応規則の例は次節および金田 [Kan 92] にあげた．後述するグラフ彩色問題や [Kan 92] でしめした N クウィーン問題などをはじめとするおおくの単純なシステムにおいては反応規則は 1 個だけ存在するが，複数の変化のしかたをみとめるより複雑なシステムにおいては複数個の反応規則が存在する．

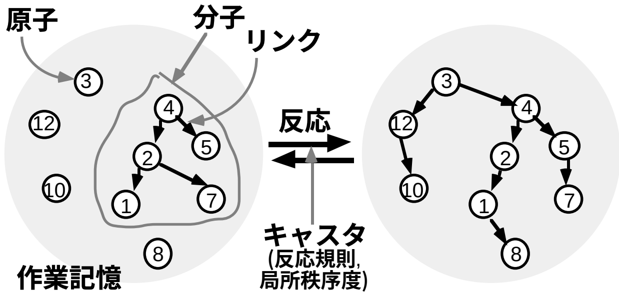


図 3　**化学的キャスティング・モデルの構成要素**

局所秩序度は局所的な “組織化” あるいは “秩序化” の程度をあらわす量であり，作業記憶の局所的な状態が “のぞましい” ほどおおきな値をとるように，ユーザにより定義される．局所秩序度の定義のしかたとしては自己秩序度と相互秩序度とがあるが，これらについては金田 [Kan 92] が説明している．前記の N クウィーン問題のキャスタは相互秩序度を使用しているが，以下の説明においては，かんたんのため自己秩序度だけをかんがえる．自己秩序度は元素ごとに定義され，規則の適用時に原子ごとに計算される．ただし，その値は当該原子の内部状態だけでなく，その原子からでるリンクがつながったさきの原子の状態にも依存しうる．

反応はつぎの 2 つの条件をみたすときにおこる．反応規則の左辺 LHS および右辺 RHS には原子とマッチする 1 個または複数個のパタンがあらわれるが，第 1 の条件は左辺にあらわれるすべてのパタンのそれぞれにマッチする原子が存在することである．

反応がおこるとこれらの原子は消滅して，そのか

[注9] ひとつの原子から複数の無名のリンクまたは複数の同名のリンクをはることをゆるすと，興味ある動作が実現される．実際，後述のグラフ彩色の問題ではこのようなリンクを使用している．

[注10] CCM においては，化学反応系のように (物理的な意味での) 距離の概念が導入されていないから，局所的ということばは距離がちかいということを意味しない．

わりに右辺にあらわれる原子が生成される[注11]．ただし，左辺と右辺とに対応する原子があらわれるばあいは，その原子は生成・消滅するかわりにかきかえられる[注12]．このような規則とそれにあらわれる (左辺および右辺の) パタンにマッチするすべての原子との組を**インスタンス**とよぶ．ひとつのインスタンスがふくむ原子のうち，反応前に存在するものすなわち左辺にあらわれるものの局所秩序度の総和を“反応前のインスタンス秩序度”，反応後に存在するものすなわち右辺にあらわれるものの局所秩序度の総和を “反応後のインスタンス秩序度” とよぶ．反応後のインスタンス秩序度をあらかじめ計算したものが反応前のインスタンス秩序度よりおおきいとき，すなわち反応によって局所秩序度の和が増加する時だけ反応がおこるというのが第 2 の条件である．

そして，いずれかのインスタンスについて上記の 2 条件がみたされているかぎり，反応はくりかえしおこる．これらの条件をみたすインスタンスが存在しなくなると実行は中断される[注13]．

ただし，一般には上記の 2 つの条件をみたすインスタンスは複数個存在する．条件をみたすインスタンスが複数個生成される原因としては，ひとつの規則の条件部をみたす原子の組が複数個存在するばあいと，複数の規則についてその条件部をみたす原子の組が存在するばあいとがある．いずれのばあいでも，これらのインスタンスのうちのいずれがどのような順序で，あるいは並列に反応するかは非決定的である．いいかえると，反応の順序はシステムが自律的にきめる．したがって，反応を完全に制御することはできない．

しかし，これをある程度は制御することができないと，のぞんだ計算を実現できないばあいがある．そのために，**スケジュリング戦略**というものがさだめられる．ユーザはそれを指定することによってインスタンスの選択順序を制御し，反応の順序を部分的に制御することができる．スケジュリング戦略にはインスタンスを系統的に選択する**系統的戦略**[注14]と，ランダムに選択する**ランダム戦略**とがある．スケジュリング戦略としてこれ以外のものもかんがえられるが，前記のものもふくめて金田 [Kan 92] がくわしくのべているので，ここでは説明しない[注15]．

ところで，反応規則も局所秩序度も局所的に (ミクロに) 作用するということをのべた．局所性は不完全性の一種だとかんがえることができるが，CCM はこのような不完全な “計画” による大域的な組織化をめざす (すなわちマクロな，より完全なものをめざす) 計算モデルである．

4.2 CCM による例題

この節では，CCM にもとづくキャスタの例として，グラフ彩色問題のキャスタをとりあげる．まず問題を説明する．グラフ彩色問題は，グラフの頂点をあらかじめきめられた数の色たとえば 4 色にぬりわけるという問題である．ただし，ここでグラフの隣接頂点が同色にならないようにぬらなければならない．この問題は，グラフの頂点を地図の領域に対応させ，グラフの辺を地図の領域境界と対応させることによって地図の彩色問題と対応づけることができる．すなわち，おなじキャスタで地図の彩色問題をとくことができる．たとえば，**図** 4 にしめす 5 頂点からなるグラフを彩色する問題は，そのグラフにかさねてえがいた 5 領域からなる地図の彩色問題と等価である[注16]．

つぎに CCM によってこの問題をとくためのデータ構造について説明する．グラフの頂点および辺をそれぞれ原子としてあらわす．したがって，ここに

[注11] したがって，右辺にあらわれるパタンは，原子の生成に必要な情報をすべてもっていなければならない．

[注12] したがって，反応規則には記述されていないその原子へのリンクがもしあったとしても，そのリンクは保存される (dangling link とはならない)．リンクの存在および秩序度に関する条件により，CCM における反応は通常のプロダクション・システムにくらべて意味が複雑化している．とくに，後述のように反応前に右辺のインスタンス秩序度を評価する必要があることから，並列論理型言語における “コミット” と同様の困難な問題がおこる．きちんと意味を定義することは今後の課題である．なお，左辺の原子と右辺の原子は，ラベルをつけることによって対応づけられる (図 6 参照)．

[注13] 金田 [Kan 92] がのべているように，実行が中断されたあとでも条件をみたすインスタンスが生成されると，ふたたび反応がおこる．

[注14] 金田 [Kan 92] は決定的戦略とよんでいるが，まぎらわしいので系統的戦略というなまえにあらためた．

[注15] 金田 [Kan 92] ではのべなかったことを，すこしつけくわえておこう．系統的戦略を使用するとリミット・サイクル (ダイナミック・ループ) におちいるばあいがある [Kan 92] が，ランダム戦略においてはこのようなことがないので，そのほうがあつかいやすいであろう．それから，スケジュリング戦略の自己組織化というのも，ここではかんがえないが興味ある研究課題である．

[注16] なお，CCM においてはシステムの動作中に原子を追加したり削除したりすることができる．本来，自己組織化の研究はこのような開放系をあつかうはずだが，ここではシステムの外部からデータがあたえられるのはシステムの動作前にかぎることにする．

は定義はしめさないが，vertex および edge という 2 個の元素を定義する．これにより，たとえば図 4 のグラフは**図** 5 のように表現される．なお，vertex 型の原子は内部状態として色をもつ．図 5 では C1, C2, C3, C4 が色をあらわしている．

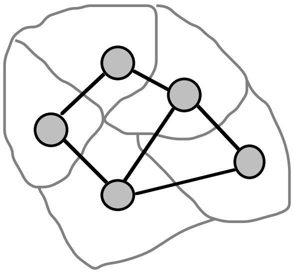


図 4　**グラフ彩色問題の例**

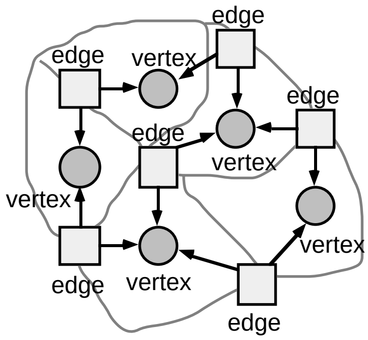


図 5　**図** 4 **の問題解決のためのデータ構造**

つぎに，グラフ彩色問題をとくためのキャスタをしめす．第 1 に，このキャスタを構成する唯一の反応規則の定義を視覚言語のかたちで**図** 6 にしめす[注17]．この反応規則は，隣接するひとくみの 2 頂点とそれらのあいだの辺をあらわす原子だけを参照して，一方の頂点の色をランダムにぬりかえる規則である．

この反応規則のより詳細な意味はつぎのとおりである．規則左辺には，原子にマッチする 3 個のパタンがふくまれている．それらのうちの 1 個は edge 型の原子にマッチし，のこりの 2 個は vertex 型の原子にマッチする．前者には edge，後者には vertex1 および vertex2 というラベルがつけられている．そして，その edge 型の原子から 2 個の vertex 型の原子への無名のリンクが存在する[注18]．したがって，この規則左辺にマッチすることができるのは，グラフの隣接する 2 頂点をあらわす vertex 型のデータと，それらをむすぶ辺をあらわす edge 型のデータである．反応によって，vertex2 にマッチした原子の内部状態がかきかえられる．すなわち，反応前にはその内部状態すなわち色は C2 だったが，それがランダムに生成された色 C3 にぬりかえられる．ここで C3 はあらかじめきめられた色のなかから選択される[注19]．ところで，C1, C2, C3 のあいだにはなんの制約も記述されていないから，それらはことなっていてもよいしおなじでもよい．

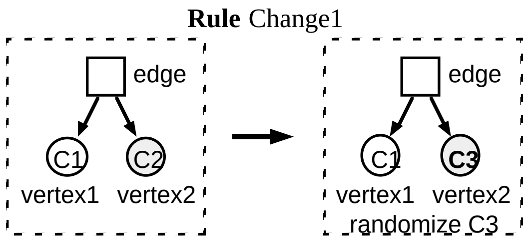


図 6　**グラフ彩色問題のための反応規則の例**

この反応規則を “不完全な計画” にもとづく計算という観点から分析する．この反応規則は，グラフが何個の頂点と何個の辺によって構成されていても上記の 3 点だけを参照するという意味において (空間的に) 局所的であり，その意味で “不完全” だといえる．また，この反応規則の適用すなわちひとつの反応は問題をとく手順の 1 ステップを構成するが，それをどうつなぎあわせて手順を構成するかは非決定的であるから，この反応規則は手順を規定してはいない．したがってこの反応規則は時間的に “局所的” であり，その意味でも “不完全” だいえる．

第 2 に，局所秩序度の定義をしめす．このキャスタには 2 種類の元素がつかわれるが，このうち vertex に対しては局所秩序度を定義しない．局所秩序度が定義されていない元素の原子の局所秩序度は 0 とみなされる．元素 edge の局所秩序度定義はつぎのように定義する．

$$o_{edge}(x) = \begin{cases} 1 & \textbf{if}\ x.vertex[1].color \neq x.vertex[2].color \\ 0 & \textbf{otherwise} \end{cases}$$

この定義は，辺の両端の頂点が同色ならば 0，そうでなければ 1 という意味である．すなわち，元素 edge の局所秩序度は “のぞましい” 状態においてよりたかい値をとる．この定義の $x.vertex[1].color$ および $x.vertex[2].color$ はそれぞれ edge 型のデータ x からでる最初のリンクおよび 2 番めのリンクがさす vertex 型のデータの *color* という内部状態すなわち

[注17] 図 6 の反応規則の記述においては，わかりやすさを重視して厳密さを犠牲にしていることをことわっておく．

[注18] このリンクは無名なので，ある edge 型のデータからでるリンクは規則左辺のどのリンクともマッチしうる．したがって，そのリンクによって結合された vertex 型のデータは，規則左辺のどのパタンともマッチしうる．

[注19] C3 はランダムに生成しているので，C3 として C1, C2 と同一の色が選択されることもある．しかし，そのばあいには反応規則の適用によってインスタンス秩序度が増加しないため，反応規則は適用されない (後述の局所秩序度の定義を参照)．このように，秩序度に関する条件があることによって，キャスタの記述が簡潔になっている．

色を意味している．“不完全な計画” にもとづく計算という観点からみると，この局所秩序度は当該の辺が接続している両端の頂点をあらわす原子を参照するが，それ以外の原子を参照しないという点において局所的であり “不完全” だといえる．

上記のキャスタを実行させると，局所秩序度がたかい値をとる方向に，つぎつぎに反応がおこる．しかし，ひとつの反応はそれにかかわる辺の周辺の辺の局所秩序度を低下される可能性がある．したがって，解にむかって直線的に動作するわけではないし，かならず解に到達するともいえない (この点については次章で再度論じる)．そこで，CCM にもとづく言語 SOOC-92 (Self-Organization-Oriented Casting または Computing)[注20] とその処理系を使用して，スケジュリング戦略としてはランダム戦略をつかって，上記のキャスタのふるまいをしらべた．SOOC-92 はテキスト・ベースの言語である．

前記のキャスタはすなおにコーディングしても動作させることができる．しかし，ここではそれを改良したキャスタを米国本土 48 州の地図の 4 色ぬりわけ [Tak 92] に適用した結果をしめす[注21]．なお，米国本土 48 州の地図を対応するグラフに変換すると 48 頂点，106 辺のグラフになる．

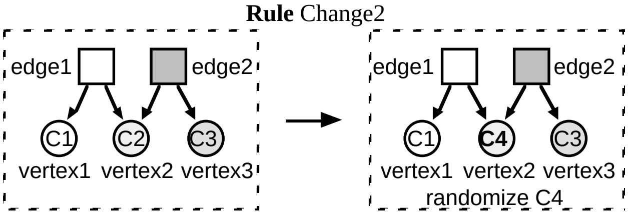
総反応回数，規則左辺のマッチング回数 (不成功におわるばあいもふくむ)，および実行時間を SUN4 の KCL (Kyoto Common Lisp) 上の SOOC-92 処理系において測定したところ，その平均値はそれぞれ 585 回，21873 回，4.18 秒となった[注22]．

[注20] 金田 [Kan 92] においてはこの言語・処理系を SOOP (Self-Organization-Oriented Programming) とよんでいたが，“完全な計画” を意味する Programming ということばを追放するために，それを Computation にあらためた．

[注21] このキャスタにおいては辺から頂点へのリンクが単方向のポインタとして実装される．ところが，これでは規則左辺のマッチングにかかる時間がながくなる．そのため，頂点から辺へのリンクをもうけることによって双方向のポインタにし，それによって高速化をはかったという点がおもな改良点である．

[注22] 発表においては，図 6 の規則では 17 頂点，46 辺の例題の解をもとめるまでに 1000 回以上の反応を要すると報告したが，これはあやまりだった．ただし，改良された非局所的な反応規則 (下図) のほうがすくない反応回数で解がもとめられるという点は，報告のとおりである．

Rule Change2



最後に，上記の例題を 2 つの提案との関係において論じよう．この例題のキャスタはデータを局所的に参照するだけで，また手順を構成する 1 ステップをあたえるだけで問題解決をはかろうとしていた．そしてこの問題においては，上記のかんたんなキャスタにおいて一応は目的を達することができた．したがって，提案におけるマクロ・モデルにもとづく観測・制御の必要はとくにないといえる．このように制御なしで目的を達することができたのは，反応規則として適切なものをえらんだという理由もあるが，最大の理由は局所秩序度として適切なものをえらんだことである[注23]．もし局所秩序度として不適当なものをえらんだときは，計算の方向を修正するためにマクロ・モデルがやくにたちうると予想される．ただし，いまのところこの予想を実証するような例題はみつかっていない．また，計算の高速化をめざすなら，上記のキャスタについてもマクロ・モデルにもとづく観測・制御が有効でありうるとかんがえられる[注24]．この点については次章でさらにのべる．

5. マクロ・モデルの例 — 確率過程にもとづくモデル

この章では，第 3 章でしめした 2 つの提案を具体化するために必要なマクロ・モデルの例について説明する．5.1 節では，このマクロ・モデルの説明に必要な大域秩序度という量を定義し，それを例題について実際に観測した結果をしめす．そして，5.2 節でマルコフ連鎖にもとづくマクロ・モデルを説明する．ただし，このモデルは未完成である．

5.1 大域秩序度の定義とその観測値

局所秩序度の作業記憶全体にわたる和を大域秩序度という．グラフ彩色問題のばあいは，頂点すなわち vertex 型のデータの局所秩序度が 0 なので，すべての辺すなわち edge 型のデータの局所秩序度の和が大域秩序度となる．したがって，辺の総数を N とすると，大域秩序度 O の値の範囲は $0 \le O \le N$ とな

[注23] グラフ彩色問題においては，すべての辺について辺の局所秩序度が 1 であるということが解の必要十分条件になっている．すなわち局所秩序度の定義が問題の仕様をあたえているから，問題をとくことができたのである．このことは，金田 [Kan 92] でしめした N クウィーン問題 (これも制約充足問題) のプログラムについてもいえる．

[注24] たとえば，観測結果にもとづいて，反応規則をまえの脚注でしめした Change2 にかきかえるというような “制御” をおこなうことは，具体的な方法はここではしめさないが，可能であろう．

る．大域秩序度が最小値 0 をとるのはすべての頂点が同一色のばあいであり，最大値 N をとるのは解がえられた状態である．米国地図の彩色においては，辺数が 106 なので O の最大値は 106 である．

SOOC-92 による米国地図の彩色のひとつの実行過程において，大域秩序度の値を反応がおこるごとに実測した結果を**図** 10 にしめす．初期状態においてはすべての頂点に同一の色をあたえている．この図から，つぎの 2 つのことがわかる．

第 1 に，CCM にもとづくグラフ彩色においては大域秩序度が単調に増加しないことが容易にわかる．これは，4.2 節でもふれたように，反応によってある頂点が隣接頂点とことなる色にぬりわけられても，その頂点の他の複数の隣接頂点とおなじ色にぬられることがあり，このようなばあいには大域秩序度が減少するからである．キャスタのなかには，このように局所秩序度の増加がばあいによって大域秩序度の減少をひきおこす**競合型**のキャスタと，局所秩序度が増加するときには大域秩序度が減少することがない**協調型**のキャスタとがある．金田 [Kan 92] でしめした N クウィーン問題のキャスタも競合型である．

第 2 の点は第 1 の点ほどあきらかではないが，図 10 をみると大域秩序度の変化はランダムにみえるから，確率過程とみなせるであろう[注25]．しかも，この確率過程は実行開始からしばらくは非定常性がつよく，その後定常にちかくなっていることがよみとれる．図 10 のばあいには反応回数が 100 回くらいまでは非定常性がつよいようにみえる．ただし，反応回数が 100 回をこえても反応により解に到達するばあいとそうでないばあいとが存在するから，真の定常状態に達してはいない．真の定常状態は解に到達した状態である．したがって，CCM にもとづく計算過程においては，つぎのような 3 つの状態がこの順にあらわれるという仮説をたてることができる．

(1) **強非定常状態**

反応ごとに確率分布が変化する状態．

(2) **準定常状態**

反応ごとに解状態すなわち大域秩序度が最大の状態の確率が増加するが，それ以外の状態のわりあい (条件つき確率) は変化しない状態．

(3) **停止状態 (定常状態)**

大域秩序度が最大の状態の確率が 1 である状態．ランダム戦略を使用しているばあいには有限時間でこの状態が実現されることはない (すなわち計算時間に上限はない)．したがって，この状態は $t \to \infty$ の極限として存在する．

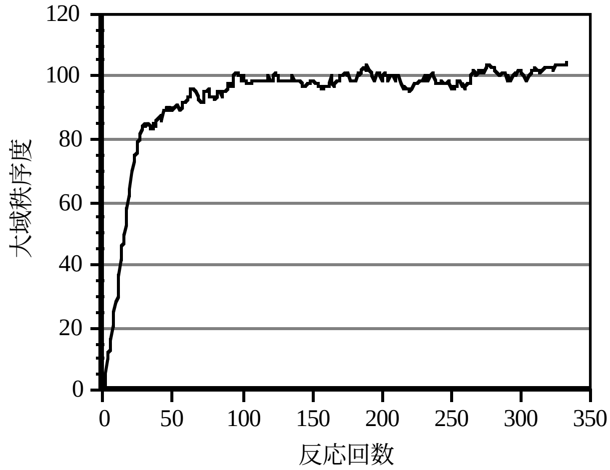


図 10　グラフ彩色における大域秩序度の実測値例

5.2 マルコフ連鎖にもとづくマクロ・モデル

CCM の計算における大域秩序度の時系列をマルコフ連鎖とみるマクロ・モデルを構成した．このモデルは，大域秩序度の時系列を観測してそれをなんらかのかたちで反応の制御にいかすことをめざしている．以下このモデルについて説明するが，まずやや一般的な説明からはじめる．

このモデルにおいては，反応がおこるごとに変化する作業記憶の状態を確率過程とみる．そのためにまず，初期状態において時刻を 0 とし，反応がおこるたびに時刻が 1 ずつすすむとみなす．そして，時刻 t における作業記憶の状態をあらわす適当なマクロな変数 $X(t)$ を確率変数とみなす． $X(t)$ がとる値として実数値や数値以外のものをかんがえることもできるが，ここでは非負の整数値にかぎられると仮定する[注26]．また，$X(t)$ に関してマルコフ性がなりたつことを仮定する．すなわち，$X(t)$ が i という値をとる確率を $p(X(t) = i)$ ($\sum_{i=0}^{I} p(X(t) = i) = 1$ がなりたつ) とし，$p(X(t) = i)$ ($i = 0, 1, \dots, I$) を要素とするベクトルを $\mathbf{p}_t$ とするとき，$\mathbf{p}_t$ と $\mathbf{p}_{t+1}$ とのあいだにつぎのような関係がなりたつと仮定する．

$$\mathbf{p}_{t+1} = \mathrm{T}\,\mathbf{p}_t .$$

ここで遷移行列 T は I 行 I 列の行列であり，その値は時刻にはよらないとする．T の固有値を $\lambda_0, \lambda_1, \dots, \lambda_I$ ($\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_I$) とすると，$\lambda_0 = 1$ がなりたつ

[注25] 本来は，確率過程とみなせることを証明する必要があるが，それはここでは省略する．

[注26] このような仮定をおくと，もはや $X(t)$ が実数値をとるばあいは同様のやりかたではあつかえなくなる．しかし，$X(t)$ が離散値であるかぎりはそれを非負の整数値に写像できるから，その意味では一般性をうしなっていない．

[Mor 79]．また，T^n はつぎのように表現することができる [Mor 79].

$$T^n = T_0 + \lambda_1^n T_1 + \lambda_2^n T_2 + \ldots + \lambda_I^n T_I .$$

$\mathbf{p}_t = T^t \mathbf{p}_0$ かつ $\lambda_2, \ldots, \lambda_I < 1$ がなりたつから，$t \to \infty$ とすれば $\lambda_1^t, \ldots, \lambda_I^t \to 0$ となり，したがって $T^t \to T_0$ となる．このような時刻 t における状態が停止状態である．また，$\lambda_2, \ldots, \lambda_I$ は 1 より十分にちいさいが λ_1 が 1 に十分にちかいというばあいには，t が十分おおきな値をとるときに $T^t \approx T_0 + \lambda_1^t T_1$ がなりたつ．このような時刻 t における状態が準定常状態だとかんがえられる．

このような仮定のもとで X として大域秩序度をとり，SOOC-92 による大域秩序度の実測値をモデルにあてはめてグラフ彩色問題およびエイト・クウィーン問題のキャスタの実行における T の値を推定すると，大域秩序度の変化をうまく説明できるモデルがえられた．紙数がたりないため T の推定法はべつの機会にしめす．これ以下の計算はグラフ彩色問題についてはまだ部分的にしかおこなっていないため，これ以降はエイト・クウィーン問題の解析結果をしめす (ただし，これまでにわかっている範囲ではグラフ彩色問題も同様の性質をしめしている)．この問題のばあい，推定された T からその固有値をもとめると $\lambda_1 = 0.986$, $\lambda_2 = 0.5$, $\lambda_3 = 0.2$, … となった．これにより上記の条件の成立がたしかめられ，したがって準定常状態の存在がたしかめられたといえる．

このモデルが実測結果とよくあっているということは，つぎのようにしてもたしかめることができる．大域秩序度 (global order degree, GOD) の実測値から $p(X(t) = i)$ の値を直接推定したものを**図** 11 にしめし，マルコフ連鎖モデルから推定した $p(X(t) = i)$ の値を**図** 12 にしめす[注27]．これらを比較すると，時間 (反応回数) のスケールにちがいがあり，また時間が 16 以下の部分すなわち $\lambda_2, \lambda_3, \ldots$ 以下の固有値に支配されている部分にはちがいがあるが，それ以外はよく一致している．したがって，マルコフ連鎖モデルじたいは適切だが，固有値の推定誤差がおおきいとかんがえられる[注28].

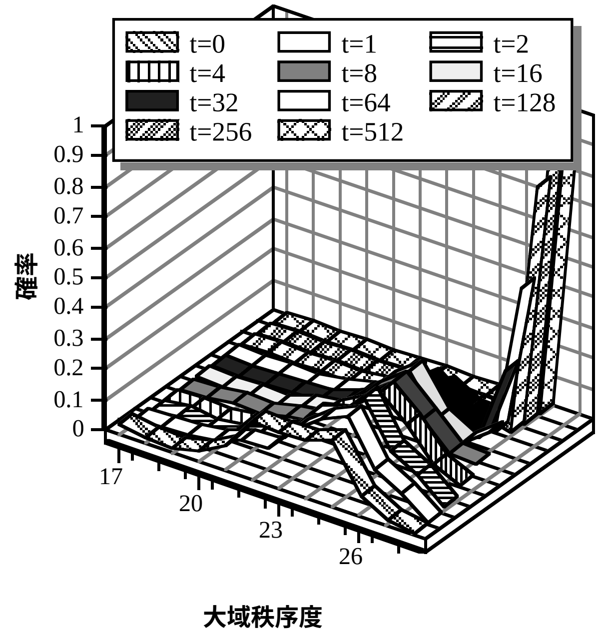


図 11　エイト・クウィーン問題の計算過程における実測値から直接推定した大域秩序度の分布の変化

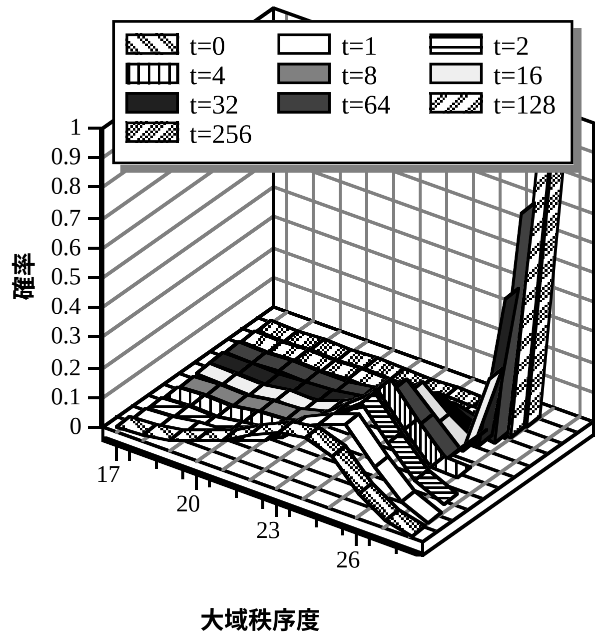


図 12　マルコフ連鎖モデルから推定したエイト・クウィーン問題の大域秩序度の分布の変化

ところで，上記のモデルにはまだつぎのような問題がある．第 1 に，推定すべきパラメタがおおすぎる．T のすべての要素を推定する必要があるから，グラフ彩色問題のばあいには，辺の数を N とすると $(N+1)^2$ 個のパラメタを推定することになる (このなかには実際には推定できないものもあるが)．したがって，パラメタ推定のために多量の測定データを必要とする．第 2 に，これはもっと重大な問題だが，このような観測によってえられたモデルをどのようにして制御にいかせばよいかは，まだほとんどわか

[注27] ここで，実測時には初期状態はランダムに生成し，またモデルにおいてもそのことを仮定している．

[注28] 時間のスケールに差があるのは λ_1 の推定誤差のためであり，$t \leq 16$ におけるふるまいに差があるのは実測値がすくないための実測値からの推定の誤差と，マルコフ連鎖モデルにおける $\lambda_2, \lambda_3, \ldots$ の推定誤差のためだとかんがえられる．

っていない[注29]．自己組織化はおろか，自己安定化をどのようにあつかえばよいかもわかっていない[注30]．ただ，すくない測定データからモデルのパラメタが推定できれば，4.2 節でふれたような計算効率の向上のためにこのモデルをいかすことは可能だろう．これらの問題の解決は今後の課題である．

6. むすび

この報告では，今後のソフトウェア研究に関して「プログラムなしの計算をめざそう」，「計算システムを自己組織系としてみよう」という 2 つの提案をした．そして，これらの提案の実現に必要なミクロ・モデルとマクロ・モデルについて説明し，金田 [Kan 92] において提案した計算モデル CCM を前者の例として位置づけ，後者の例として計算を確率過程としてみるマルコフ連鎖モデルをしめした．これらのモデルはかならずしも上記の提案を実現するための最終的なものではないが，研究の出発点として適当なものだとかんがえている．

CCM およびマルコフ連鎖モデルはまだ自己組織的な計算システムの構築のために十分ではないので，今後さらにこれらを発展させる必要がある．また，上記の提案を実現するという目的のために CCM やマルコフ連鎖モデルとはことなるよいよいモデルがないかどうか，検討する必要がある．詳細な課題については本文中でのべたので，くりかえさない．

謝辞

つぎの方々に感謝したい．日立製作所日立研究所の 坂東 忠秋 氏が同中央研究所在勤時に報告者に対して自己組織化の研究を提案することをすすめていただいたことが，この研究をはじめるきっかけとなった．また，同中央研究所の 小島 啓二 氏の，局所的な情報にもとづいて計算するだけでなく大域的な制御をはたらかせなければ目的を達することはできないだろうという指摘が，マクロ・モデルにもとづく観測と制御のかんがえかたにつながった．さらに第 33 回プログラミング・シンポジウムにおいては，おおくの方々に議論していただいた．

[注29] 性能のよい反応規則やスケジュリング戦略と性能のわるいそれらとをモデルのうえで比較することによって，性能を制御するためのいとぐちをつかむことができるとかんがえられる．実際にこの方向の研究をすでにおこなって成果もえられているので，つぎの機会に報告する．しかし，のぞましくない結果をうみだすキャスタを制御してのぞましい方向にむかわせるための方法については，まだまったくわかっていない．

[注30] ただし，安定性については金田 [Kan 92] においてある程度考察している．

参考文献

[Ber 68] フォン・ベルタランフィ: *一般システム理論* (長野 敬, 太田 邦昌 訳), みすず書房, 1973.

[Ber 90] Berry, G., and Boudol, G.: The Chemical Abstract Machine, *Proc. 17th Annual ACM Symposium on Principles of Programming Languages*, pp. 81–94, 1990.

[Hak 78] ハーケン, H. : *協同現象の数理* (小森・相沢 訳), 東海大学出版会, 1980.

[Hak 83] ハーケン, H. : *シナジェティクスの基礎* (小森・相沢 訳), 東海大学出版会, 1986.

[Kan 92] 金田 泰 : コンピュータによる自己組織系のモデルをめざして, *第 33 回プログラミング・シンポジウム報告集*, 1992.

[Las 72] ラズロー, E. : *システム哲学入門* (伊藤 重行 訳), 紀伊國屋書店, 1980.

[Mor 79] 森村 英典, 高橋 幸雄 : *マルコフ解析*, OR ライブラリー 18, 日科技連, 1979.

[Nis 88] 西垣 通 : *AI — 人工知能のコンセプト*, 講談社現代新書, 1988.

[Nis 90] 西垣 通 : 秘術としての *AI 思考*, 筑摩ライブラリー, 1990.

[Pri 77] ニコリス, G., プリゴジーヌ, I. : *散逸構造* (小畠・相沢 訳), 岩波書店, 1980.

[Pri 84] プリゴジン, I., スタンジェール, I. : *混沌からの秩序* (伏見 康治 他訳), みすず書房, 1987.

[Shi 90] 清水 博 : *生命を捉えなおす 増補版*, 中公新書, 中央公論社, 1990.

[Shi 92] 清水 博 : *生命と場所*, NTT 出版, 1992.

[Tak 92] Takefuji, Y. : *Neural Network Parallel Processing*, Kluwer Academic Publishers, 1992.

[Wie 61] ウィーナー, N. : *サイバネティクス (第 2 版)* (池原, 彌永, 室賀, 戸田 訳), 岩波書店, 1962.

質疑応答 (一部省略)

電総研 佐藤 計算量の期待値は予測できるか.

A マルコフ連鎖とみなせるという仮定のもとでは平均時間は予測できる．ただし，解がえられていない確率が 0 にはならず，最大値は有限でない．

NTT **竹内** 彩色問題の規則で辺数をふやすとどれだけ効果があがるか.

A まだ効果はたしかめられていない (発表以降にわかった点について 4.2 節脚注を参照).

佐藤 部分解が成長して解になるのか.

A 大域秩序度が解にちかい点 (局所最適解など) をランダムに探索して真の解に達するようだ.

佐藤 なぜ初期状態をランダムにしなかったか.

A グラフをかいたときにおもしろくないからだ (笑.「どういう確率過程かがわかりにくいという意」).