自律分散的に自己診断可能なマルチエージェントシステム

溝口博三*・下川俊彦*
吉田紀彦**

Multi-Agent System for Autonomous Decentralized Self-Diagnosis

by

Hiromi MIZOGUCHI*・Toshihiko SHIMOKAWA*
and Norihiko YOSHIDA**

It is desired to construct a mechanical system that works for a long time without any human support. But it is usually considered to be difficult to make the decentralized self-organizing autonomous system fault tolerant. In this paper, we propose to apply the theories of self-diagnosable systems for computer networks to this problem. As an example, we apply the highly structured one-stept-permanent fault diagnosable system, proposed by Kohda, to a decentralized self-organizing autonomous robotic system that makes up a circle. The result of simulation shows the usefulness of the proposed method.

1．始めに

近年，インターネットが急激に普及してきている。大きな要因として，コンピュータ技術の進歩による小型高性能化と，ネットワークの整備が挙げられるであろう。これにより，これからの社会がネットワークを基盤とする情報社会であることを容易に予想させる。ネットワークを基盤とすることで，リソースの管理，分配，利用は高度に分配されたネットワーク環境で行なうことが確実である。このような環境に対応する為のより優れた情報処理技術の出現が期待されている。

また，集中型処理から分散型処理への移行が進んでいる。これは，コンピュータの行なうべき問題が高度化かつ多様化するに従い，これまでの集中処理では対応できなくなってきているからである。1台の高性能コンピュータによって集中処理を行なうよりは，ネットワーク化とコンピュータの小型高性能化によって，複数のコンピュータによる分散・協調処理の方が効率が良くなってきている為であろう。1つの要素に全ての機能を集中するのでは無く，多数の要素に機能を分散させて，それらが1つの目的に向かって協力して働く様にすることによって，コスト，柔軟性，拡張性，および信頼性に優れたシステムを構築することが出来る。

マルチエージェントシステムとは，この様な協調分散環境における問題解決機構であり，ここでは個々の問題解決装置をエージェントと呼び，このエージェントに分散した問題を処理させることで，協調しながら問題を解いていくシステムである。

しかしながら，これらのシステムにおいては，これまで故障の発生について想定されて来ていなかった，ゆえに故障が生じるとシステム全体の秩序が崩壊し，人為的な支援無しにはその修復を行なうことは不可能であった。よってこの研究では，このような故障を発見し，発見した故障しているエージェントに必要な処

平成11年4月23日受理
*九州大学大学院システム情報科学研究科　福岡市東区箱崎
(Graduate School of Information Science and Electrical Engineering, Kyushu University, Hakozaki, Fukuoka)
**情報システム工学科 (Department of Computer and Information Sciences)

理を施すことを目的とする。

故障しているエージェントを発見する為のシステムとして，コンピュータネットワークシステムにおける自己診断可能システムの1つに，香田によって提案されている highly structured system[1] というものがある。highly structured system は，検査数最少で計算複雑度 O(|E|)(|E|は検査数) であるという特徴を持つ。本研究では，このシステムを応用することによって，分散的に故障エージェントを発見する方法を考える。また，これまでの研究では，検査結果の導出までしか考えられていなかったので，各エージェントへの検査結果の通知，故障した場合の対応に必要な処理(排除または修復) を施すことまでを行なう。

本論文では，第2章で自己診断可能システムの説明，第3章で自己診断可能システムをマルチエージェントシステムに適用した場合のシステム構成法を，第4章で実際の例題についての実験とシステムの評価を行ない。第5章で研究のまとめと今後の課題について述べる。

2．自己診断可能システム

2.1 PCM モデル

ディジタルシステムの複雑化，大規模化に伴い，システムの信頼性を確保することが次第に困難になり，自己診断可能なシステムの実現が望まれている。自己診断可能なシステムとは，それを構成しているユニットの中で故障しているユニットを検出し得るシステムである。

自己診断可能なシステムとしては，Preparata, Metze, Chien の提案による PMC モデルに基づく自己診断可能システム[2] が最近，再び注目されてきている。PMC モデルを用いて，自己診断可能システムの必要十分条件，構成方法，検査結果の集合から故障ユニットを検出する解析問題などの研究が盛んに行なわれているが，その中で，香田により highly structured system[1] というものが提案されており，ここでは，その説明を行なう。

自己診断可能システムは，それを構成しているユニットだけで，故障しているユニットを検出し得るシステムである。しかし，ユニットは自分自身を検査することができない。ゆえに，ユニットは他のユニットに自分自身の状態を検査してもらわなければ，自身の状態が正常か故障か分からない。ただし，検査ユニットが正常であれば，被検査ユニットの正常，故障状態に応じた検査結果を出力するが，検査ユニットが故障であれば，いずれの検査結果を出力するかは決まっていない。

自己診断可能システムは通常，ユニットの集合 V と検査の集合 E とで構成される有向グラフ G=[V, E]で定義される。Preparata らは，ユニットを節点に，検査を弧に対応させ，節点 v の集合 $V=v$ と弧 (v, u) の集合 $E=\{(v, u)\}$ とで定められる有向グラフ $G=[V, E]$ を用いて自己診断可能システムを表現した。弧 (v, u) の始点 v, 終点 u に対応するユニットは各々，検査ユニット，被検査ユニットと呼ばれる。さらにその検査結果 $a(v, u)$ を弧の重みに対応させた（図1）。検査結果の集合は症候群 (syndrome) と呼ばれる。また症候群が与えられたとき，システム中の故障ユニットを検出する問題は，解析問題と呼ばれる。

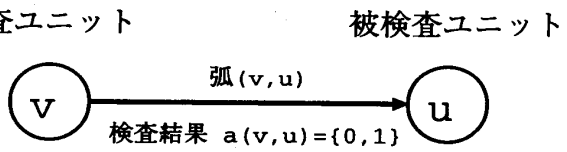

図1．PMC モデルにおけるシステムの基本構成

2.2 t-SD システムと t-OD システム

システムがt個までの多重故障を許す時，症候群から少なくとも1つの故障ユニットを検出しうるシステムを t 重故障逐次診断可能システム (t-fault sequentially diagnosable system; 略してt-SDシステム[3] といい，全ての故障ユニットを検出しうるシステムを t 重故障同時診断可能システム (t-fault one-step diagnosable system; 略して t-OD システム[4] という。

システム $G=[V, E]$ において，X を V の部分集合 ($X\subset V$)，$\bar{X}$ を X の補集合 ($\bar{X}=V-X$) とする。$X, \bar{X}$ を各々故障，正常ユニットの集合であるとする時の，起こり得るすべての症候群の集合を S(X) とし，その1つの症候群を s(X)(s(X) ∈S(X)) とする。s(X) を症候群としてもつ故障パターン（正常ユニットと故障ユニットを区別したものであり，故障ユニット数は t 個以下である）は一般に複数個存在する。そこで，L1，L0 を各々症候群より導出された故障，正常ユニット集合とし，これらを用いて故障パターンを表現することとする。s(X) に対し，自明な故障パターン

$$L_1=X,\ L_0=\bar{X} \tag{1}$$

が存在する。

定義1：任意の $s(X)$ ($1\leq|X|\leq t$) に対し，これを症候群として持つ故障パターンが，

$$L_1=X_i,\ L_0=\bar{X}_i(|X|\leq t, 1\leq i\leq l) \tag{2}$$

のように1個存在し，これ以上存在しない時

$$x \in \bigcap_{i=1}^{l} X_i \tag{3}$$

なる x が存在し，更に，$s(\phi)$ に対する故障パターンが自明な故障パターン

$$L_1 = \phi,\ L_0 = V \tag{4}$$

とし，これらを用いて故障パターンを表現することと以外に存在しない時，およびその時に限り，G は t-SD であるという。但し，$|X|$ は X の要素数であり，ϕは空集合である。

定義2：任意の $s(X)(|X| \leqq t)$ に対し，式1以外の故障パターンが存在しない時，およびその時に限り，G は t-OD であるという。

Hakami と Amin はシステム G が t-OD システムであるための必要十分条件を次のように与えている。

定理1：総ユニット数 n のシステム G が t-OD であるための必要十分条件は次の条件で与えられる。

1．$n \geqq 2t+1$
2．$\mathrm{Din}(v) \geqq t$(但し，Din(v) は v を検査するユニットの個数)
3．v の任意の部分集合 $X(|X| = n-2t+p,\ 0 \leqq p \leqq t)$ に対し，

$$|\Gamma(X)| > p \tag{5}$$

$$\Gamma(X) = \{v \in V-X \mid \exists x \in X, (x, v) \in E\} \tag{6}$$

なお，任意の2つのユニットが互いに検査し合わないシステム G が t-OD であるための必要十分条件は，上記1，2で与えられる。

2.3 highly structured system

highly structured system[1] とは，香田によって提案されている，自己診断可能システムであり，検査数最少で計算複雑度 $O(|E|)$($|E|$は検査数) であり，また，各ユニットの検査順位を問わない，各ユニット毎にローカルな情報で検査が可能，などの長所をもつ症候群解析法が提案されている。

システム $G=[V, E]$ の各ユニット $v(v \in V)$ が常に *Subsystem* $H(v\ ;\ \mu,\ \nu)$ を持つとき，システム G は highly structured system であるという（図2）。

ここで，矢印は検査，μは長さ2の検査本数，νは長さ1の検査本数を表す。この highly structured system に対して，次の定理2，3が証明されている。

定理2：システム $G=[V, E]$ が図2の Subsystem

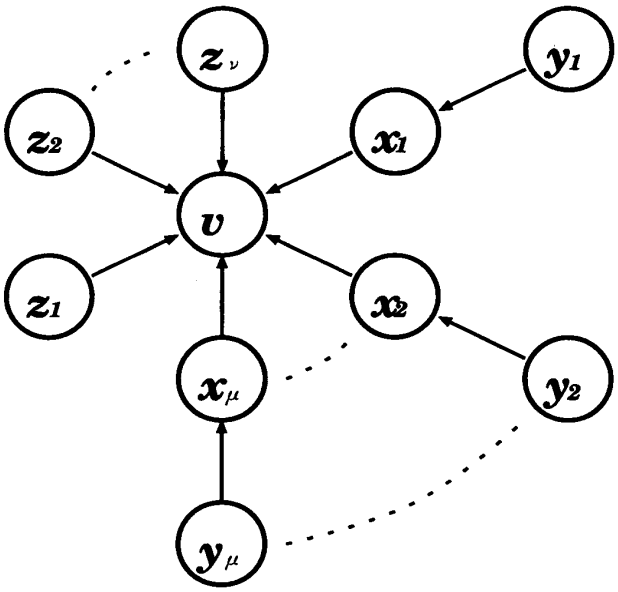

図2．Subsystem H(v ; μ, ν)

$H(v\ ;\ \mu,\ \nu)$ を部分システムとして持ち

$$\mu + \lfloor \nu/2 \rfloor \geq t \tag{7}$$

$\lfloor b \rfloor$は b を超えない整数　(8)

でかつ，v が任意のユニット $u \in V$ に到達可能であるならば，G は t-SD である。

定理3：システム $G=[V, E]$ が任意のユニット $v(v \in V)$ に対し，*Subsystem* $H(v\ ;\ \mu,\ \nu)$ をもち，かつ

$$\mu + \lfloor \nu/2 \rfloor \geq t \tag{9}$$

ならば，G は t-OD システムである。

また，t-OD システムは，検査数が nt 個，t-SD システムは検査数が n+t−1 個必要である。

図2の *Subsystem* $H(v\ ;\ \mu,\ \nu)$ において，与えられた症候群を s とする。この検査パターンを図3のように6つのタイプに分ける。

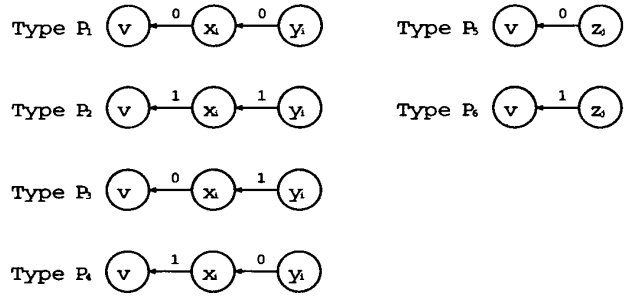

図3．Subsystem H(v ; μ,) の検査パターン

与えられた s に対し，Type Pi の検査パターンの数が pi 個であるとする（i＝1，…，6）。このとき，図3より，

$$p_1+p_2+p_3+p_4=\mu \tag{10}$$

$$p_5+p_6=\nu \tag{11}$$

$$\mu+\lfloor \nu/2 \rfloor \geq t \tag{12}$$

が成立する。又，

$$\delta=p_1+\lfloor \nu/2 \rfloor-(p_4+p_6) \tag{13}$$

とおく。このとき，次の定理4が症候群解析法として与えられている。

定理4：$\delta \geqq 0$ と v が正常であることは，同値である。

つまり，定理2または定理3を満足するシステムにおいて，図2の *Subsystem* $H(v;\mu,\nu)$ の v となれるユニットについて，定理3を適応することによって，v の状態が検査できる。

3.エージェントの自己診断

マルチエージェントシステムでは，複数のエージェントが互いに協調しながら，全体として1つの問題の解決，もしくは何かのバランス状態を構成する為のシステムである。しかしながら，従来のマルチエージェントシステムにおいては，故障の発生そのものが想定されていない為，故障が発生するとシステム全体の秩序が崩壊し，システムを正常に運営することが不可能となる。その際人為的な支援なしには，その修復を行なえなかった。もし，誤った情報処理，発信を行なうエージェント（故障エージェントと呼ぶ）がシステム内に存在する場合，少なくともその故障エージェントを同定，排除，または修復しない限りシステム全体の信頼性を維持することは困難となる。ゆえに，マルチエージェントシステムの高信頼化を実現するには，中央集権的な制御機構なしに，故障エージェントの同定，排除，または修復を行ない，システムの秩序を回復させる仕組みが必要不可欠であると言える。そこで，そのような仕組みの1つとして，故障エージェントに自己診断機能を持たせることを考える。各エージェントが自分自身の故障の有無を診断し，故障が生じていると診断される場合は，自動的に排除または修復，他のエージェントへの通知等のシステムの秩序を回復する為の処理を行なう。

マルチエージェントシステムをコンピュータネットワークシステムの一種とみなせば，第2章で述べた自己診断可能可能システムの応用が可能となる。ここで，自己診断可能システムでのユニットをマルチエージェントシステムのエージェントに対応させることによって，理論の導入を行なう。ただし，実際のマルチエージェントシステムでは他の全てのエージェントと協調しながらではなく，ある特定のエージェントとのみ協調して処理を行なうことが一般的である。また，状況の変化や必要に応じて，エージェントの追加や削除が行なわれることも考えられる。従って，全エージェントの数などの環境に柔軟に対応できる故障システムが必要となる。その為，システムを第2.3節で述べた highly structured system を応用することによって自己診断可能システムを構築する。しかし，この highly structured system をはじめとする自己診断システムは，検査の組合せや症候群解析法などは議論されているが，実際にどのユニットが検査結果を解析するのか定められていなかった。従ってこれまでは，検査結果集め，解析するサーバー的な存在が必要であった。そこでこの研究では，検査結果を解析するサーバー的な存在を排除し，分散的に故障エージェントの同定，排除もしくは修復を行なうシステムを構築する。

3.1 故障の種類

本研究での故障は，エージェントの内部情報の故障を考える。ここでの内部情報とは，故障システムの運営に関する内部情報を含まない。また，通信システムの故障もないものとして，研究を行なう。

3.2 システム構成

このシステムの中心的な存在として，第2.3節で述べた highly structured system の中の Subsystem H(v；μ，ν) を扱う。この Subsystem H(v；μ，ν) はこのシステムの中心のエージェントである v の正確な診断が可能である。

基本的な考え方として，まず故障エージェントを導出するのではなく，1つの正常なエージェントを導出する。まず，任意のエージェントを選び，これを *Subsystem* $H(v;\mu,\nu)$ の中心のエージェント v とする。v は，自身で *Subsystem* $H(v;\mu,\nu)$ の構築に必要な検査数だけ他のエージェントに検査を依頼する。検査結果を v に集めることによって，v は自身が正常か故障かを判断することができる。もし v が故障であるなら，あらたに任意のエージェントを選び，Subsystem H(v；μ，ν) の中心のエージェントとして，検査を行なう。正常なエージェントが導出できたなら，そのエージェントによる検査は正確な検査であると保証できるので，そのエージェントを起点として，他の

全てのエージェントの検査を行なう。これによって，全ての故障エージェントの導出が可能である。

この方法では，*Subsystem* $H(v\ ;\ \mu,\ \nu)$ の中心のエージェントとなる v が自分で *Subsystem* $H(v\ ;\ \mu,\ \nu)$ を構築し，その結果を解析することから，分散的に故障診断を行なうことが可能となる。

3.2.1 エージェントの条件

ここでのエージェントは以下の条件を満たしているものとする。

- 通信機能の故障はない
- 自身の故障の検査は出来ないが，他のエージェントから自分を検査してもらった結果から自分の状態を知る判断能力を持ち，その能力の故障はない。
- 自分以外の任意のエージェントとの通信が可能

3.2.2 構　成

このシステムにおいては，図4のように，全てのエージェントに共有のデータプールの存在を持つことにする。このデータプールは，全てのエージェントの情報（登録エージェントの存在，その状態，全体数）を持つ，また，故障検査，検査結果から状態の判断，*Subsystem* $H(v\ ;\ \mu,\ \nu)$ の構築等，システムの運営には関わらず，エージェント間の情報の授受のみを行なう。

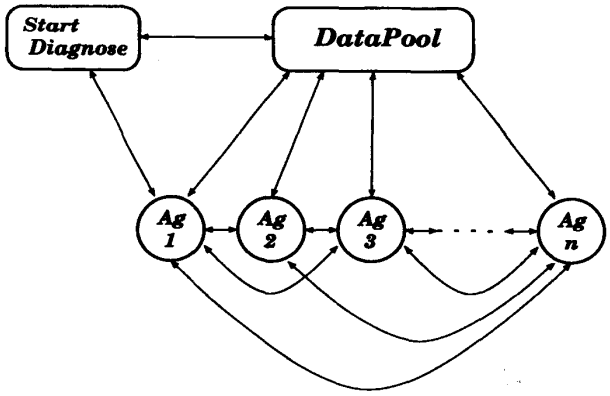

図4．自己診断可能システムの構成

データプールの設置の理由として，次のことが挙げられる。仮にデータプールの存在がない場合を考えてみると，各エージェントは，自分以外の他の全てのエージェントの情報を持たなくてはならない。このとき，エージェントの追加，削除を行なう場合，全てのエージェントに対して通信を行なわなくてはならないので，容易にエージェント数の変更が行なえない。ゆえに，データプールの設置によって，各エージェントの情報の一括管理を行ない，各エージェントは，このデータプールから必要な情報の送受信を行なうことにより，システムを実行する。

また，診断を開始する為の命令を出すプロセスが必要である。このプロセスを StartDiagnose と呼ぶことにする。StartDiagnose は，データプールから *Subsystem* $H(v\ ;\ \mu,\ \nu)$ の中心のエージェントになれるエージェントの情報をデータプールから取りだし，そのエージェントに診断開始を依頼する為のプロセスである。

3.2.3 システム構築方法

1．任意のエージェント v を決定（図5）

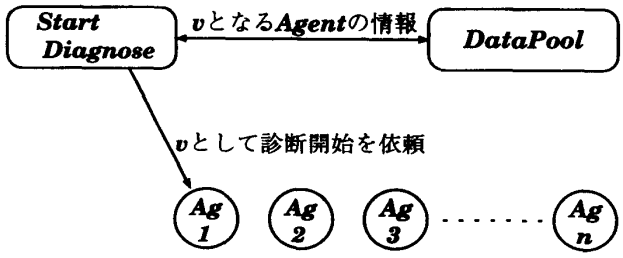

図5．任意のエージェント v を決定

- StartDiagnose が，データプールから登録エージェントの情報を受けとり，そのエージェントを *Subsystem* $H(v\ ;\ \mu,\ \nu)$ の中心のエージェントとして診断を始めるよう依頼する。

2．v が自らを中心とした *Subsystem* $H(v\ ;\ \mu,\ \nu)$ を構築（図6）

- 過去の検査結果を収得し利用する方法
- 過去の検査結果を利用しない方法

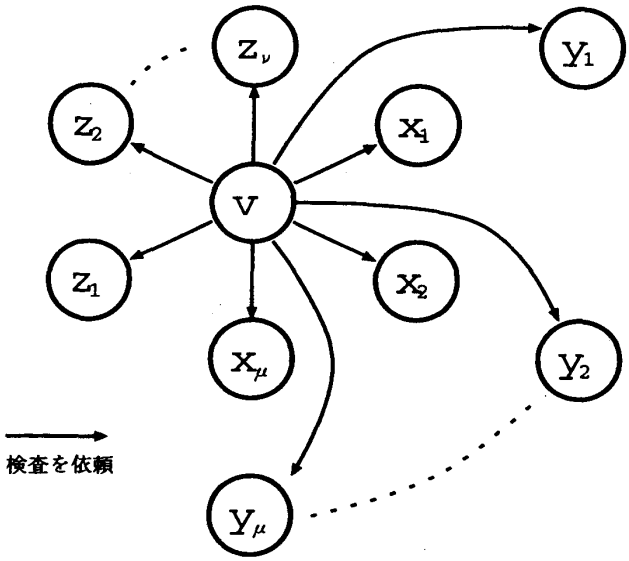

図6．vが自らを中心とした subsystem H を構築

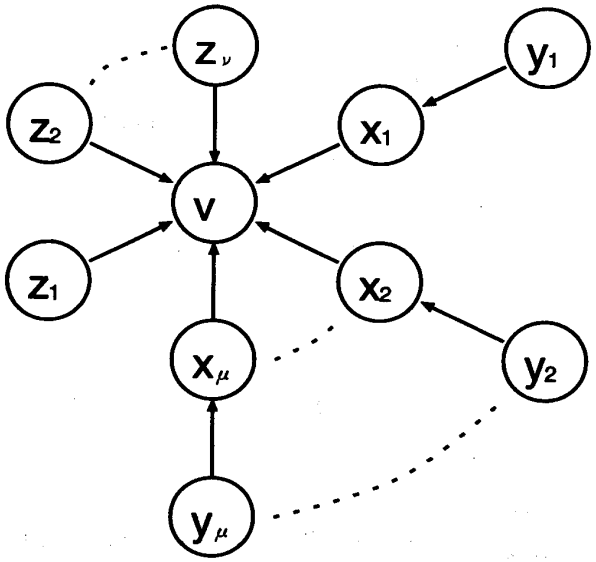

図7．検査を依頼されたエージェントが検査を実行

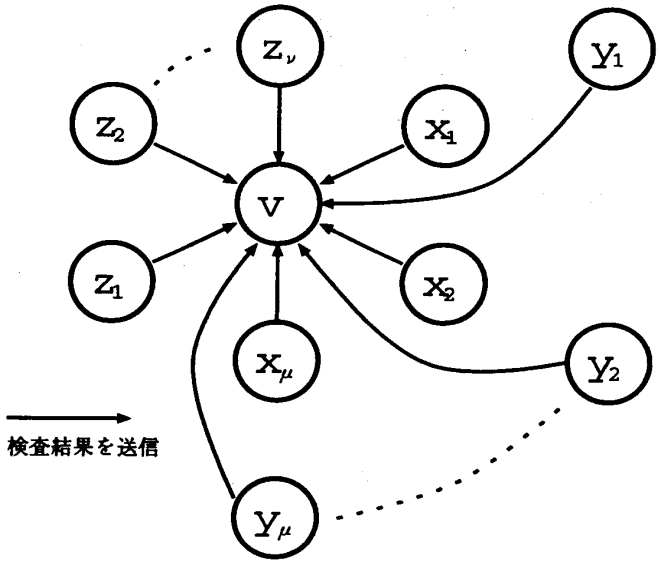

図8．検査結果をvに送信

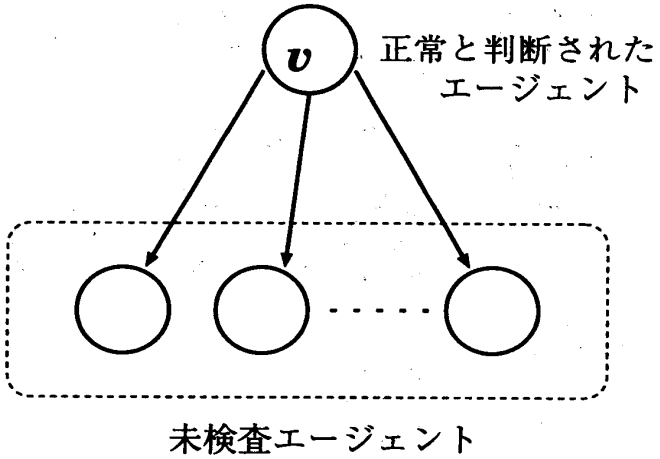

図9．vが他の全てのエージェントを検査

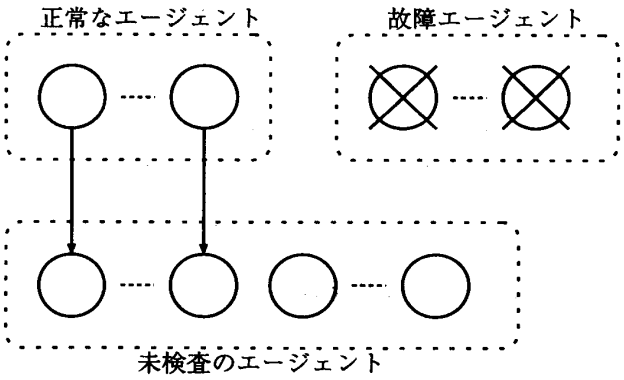

図10．正常と判断されたエージェント全てが，未検査のエージェントを検査

3．検査を依頼されたエージェントが検査を実行（図7）

4．検査結果をvに送信，v 自身が自身の状態を判断（図8）

5．v が正常なエージェントと診断されるまで上記1～4を繰り返す

6．v を起点として全ての未検査のエージェントを検査

- v が他の全てのエージェントを検査（図9）
- 正常と判断されたエージェント全てが、未検査のエージェントを検査（図10）
- 診断の終了は，データプールに登録されている全てのエージェントが正常または故障と判断された時点で終了する。

7．故障と判断されたエージェントは，故障の種類に応じた処理を施される

修復可能な故障　故障を修復し，正常なエージェントとして動作

修復不可能な故障　エージェントを停止し，データプールの登録を抹消する

3.2.4　検査数

このシステムでは，正常なエージェントを発見するまで *Subsystem* $H(v\,;\,\mu,\,\nu)$ を構築すればよい。また，正常なエージェントが発見されてから，（エージェント数－1）回の検査を行なうことによって，検査を終了する。ここでは，過去の検査結果を利用しない場合を考える。この検査数は，1つの *Subsystem* $H(v\,;\,\mu,\,\nu)$ の構築に以下の回数の検査を行なう。

$$2t = 2\left\lfloor \frac{n-1}{2} \right\rfloor \qquad (14)$$

さらに正常なエージェントを発見するまで *Subsystem* $H(v\,;\,\mu,\,\nu)$ を構築するので，故障エージェント数を

1個とする時，

$$B(l,k)=\frac{n-1}{n-1+k}\times\left\{2\left\lfloor\frac{n-1-l+k}{2}\right\rfloor+(n-(l-k-1))\right\}+\frac{k}{n-l+k}\times B(l,k-1) \quad (15)$$

$$B(l,0)=2\left\lfloor\frac{n-1-l}{2}\right\rfloor+(n-l+1) \quad (16)$$

とおくと，検査数 $C(l)$は，次式によって求められる。

$$C(l)=B(l,l) \quad (17)$$

さらに，平均検査個数は，故障の個数が i 個の時の確率を P(i)とすれば，次式によって求められる。

$$\sum_{i=0}^{t}P(i)C(i) \quad (18)$$

4．実験と評価

具体的な問題として以下の例においての故障を想定し，エージェントの自己診断可能システムが正常に動作するか確かめ，さらに通信回数，検査回数を調べた。

4.1 問題

4.1.1 自律分散的に円周を構成する群エージェント

具体的な例題として取り上げるのは，無秩序な初期状態からエージェントどうしが互いに情報を交換しつつ，中央集権的な制御機構なしに，自律分散的に円周という秩序的な形を形成するアルゴリズム[5]であり，ミルウォーキー大の鈴木によって考案された。この概要は次の通りである。

前提条件として，エージェントは次の情報および機能を持つ。

- 各エージェントは最終的な円の直径を知っている
- 自分と他のエージェントとの距離を算出する機能を持つ

そこで，以下の行動を行なう。

1．自分と他のエージェントとの距離を求め，誰が最遠エージェントおよび最近エージェントなのかを知る
2．最遠エージェントとの距離が円の直径より長い場合は，最遠エージェントに少し近付く
3．最遠エージェントとの距離が円の直径よりある割合だけ短い場合は，最遠エージェントから少し遠ざかる
4．最遠エージェントとの距離が円の直径にほぼ等しい場合には，最近エージェントから少し遠ざかる
5．すべてのエージェントが1～4を繰り返し，自分が4の状態になったとすべてのエージェントが判断したら終了する

4.1.2 故障の種類

第4.1.1節のシステムにおいて，次の故障を想定する。

1．円の直径情報の故障
- 他のエージェントから，正しい直径情報を受けとることにより，情報の修復が可能

2．自分と他のエージェントとの距離測定装置の故障
- 修復不可能

実際のシステムは，各エージェントが以下のような軌跡を描く。

- 全てのエージェントが正常（図11）
- 1つのエージェントが故障，その他全てのエージェントが正常（図12）

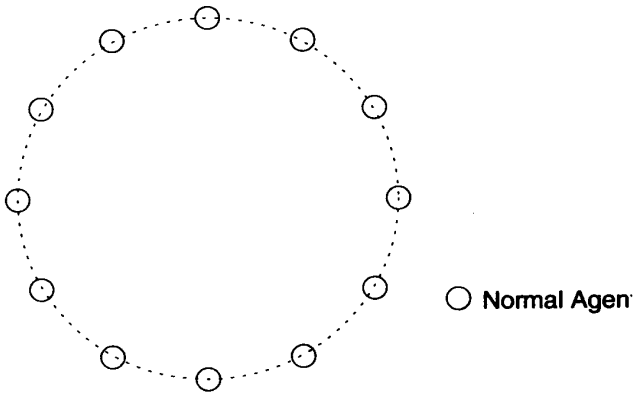

図11．正常な状態

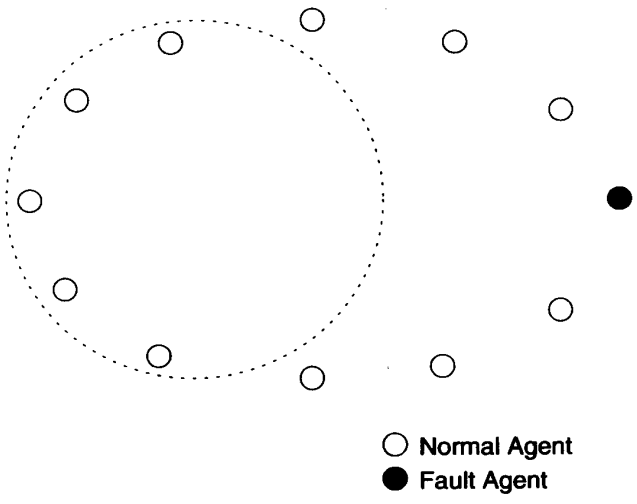

図12．故障した状態例

4.2 実 験

第4.1章におけるシステムにおいて，全エージェント数を10個，エージェントの持つ直径情報を10cmとして，実験を行なった。ここで，全エージェント数が10であることから，4個までの多重故障を許す。

4.2.1 実行確認

エージェントの状態を以下の表1のように設定して，システムを実行した例を示す。

表1．エージェントの状態

エージェントNo.	直径情報	距離測定装置	状 態
0	正 常	正 常	正 常
1	正 常	故 障	故 障
2	正 常	正 常	正 常
3	故 障	故 障	故 障
4	正 常	故 障	故 障
5	正 常	正 常	正 常
6	正 常	正 常	正 常
7	正 常	正 常	正 常
8	正 常	正 常	正 常
9	故 障	正 常	正 常

1．任意のエージェントを選ぶ
- No. 9のエージェントに決定

2．No. 9のエージェントが自身を中心とした *Subsystem* $H(v\,;\,\mu,\,\nu)$を構築し，検査結果をNo. 9のエージェントに集める

3．検査結果を解析した結果，No. 9のエージェントは故障であったので，別のエージェントNo. 0を選ぶ

4．新たにNo. 0のエージェントが自身を中心とした *Subsystem* $H(v\,;\,\mu,\,\nu)$ を構築し，検査結果をNo. 0のエージェントに集める

5．検査結果を解析した結果，No. 0のエージェントは正常であったので，No. 0のエージェントが他の全てのエージェントを検査する
- この時，修復可能な故障エージェントがあった場合，同時に修復処理を行なう

システムの実行結果として，表2のようになる。

表2．システム実行後のエージェントの状態

エージェントNo.	状 態
0	正 常
1	故 障
2	正 常
3	故 障
4	故 障
5	正 常
6	正 常
7	正 常
8	正 常
9	正 常

この結果から，正確な故障エージェントの同定と，修復が行なえることを確認した。また，他の全ての状態において，システムの実行を確認した。

5．おわりに

本研究では，分散協調処理であるマルチエージェントの自己診断可能システムについて，中央集権的な存在なしに，分散的に故障エージェントの同定，修復または排除をする為の効率的な検査方法とその構築法，検査結果からの状態の判断などを模索し，実際にシミュレーションを作成して，動作を確認した。これにより，これまでの自己診断可能システムより検査数が少なく，分散的に診断できるシステムを確認した。

しかし問題点として，このシステムでは通信に関する故障を想定されていないので，通信に関する故障が起こった場合，システムが正常に動作しないことが挙げられる。また，診断を開始する為のプロセスや各エージェントに共有のデータプールを用いているので，その部分が故障した場合，故障診断が不可能になる。よって，システムのすべての構成要素の故障を検出できるシステムが望ましい。ゆえに，これからの課題として，これらの問題点を解決したより故障に強いシステムの作成を考えていきたい。

参 考 文 献

1）T. Kohda : A simple discriminator for identifying faults in highly structured diagnosable systems, Journal of Circuits, Systems, and Computers, Vol. 4, No.3, pp. 255-277 (1994)

2）P. Preparata, G. Metze, and R. T. Chien : On the connection assignment problem of diagnosable systems, IEEE Trans. Electromag. Comput. EC -16, 6, pp. 848-854 (1967)

3）香田徹；t 重故障逐次診断可能システム，信学論 (D)，J61-D,9，pp 688-694 (1978)

4）香田徹；t 重故障同時診断可能システム，信学論 (D)，J61-D,9，pp 680-687 (1978)

5）澤田政宏，田邊慎一，吉田紀彦：自律分散的に秩序形成を行なうロボット群の並列計算機上でのシミュレーション，情報処理学会第46回全国大会，3A-1 (1993)