PNC TN9410 97-049

公開資料

知的保全管理システムの概念構築

1997年4月

動力炉・核燃料開発事業団
大洗工学センター

公　開　資　料
PNC TN9410 97-049
１９９７年４月

知的保全管理システムの概念構築

須田一則*、米川強*
吉川信治*、小澤健二**

要　旨

原子力プラントの運転・保守における稼働率や安全性を向上を図るべく様々な研究が行われている。我々は既存のプラントにおいて運転員及び保守員が果たしていた役割を人工知能及び自律ロボット等で代替する自律型プラントの要素技術として、自律型運転制御システムと知的保全管理システムの開発を実施している。

特に、自律型プラントの保守機能を司る知的保全管理システムは、プラント内の保全管理に係わる状態の監視、機器の異常・故障の判断、保守・補修に係わる保全方策の策定を自律的に行う機能が必要であり、また、その結果を運転制御システム及び自律ロボットへ指示・伝達するという重要な役割を有している。本報告では、人工知能技術のマルチエージェントによる分散協調手法を採用し、センサ・アクチュエータレベルの協調による問題解決を実施する方法として、分散協調技術による故障検知機能、故障診断評価機能及び保全方策策定機能の３機能の検討、及びこれらの情報伝達機能を司る分散協調通信機能について検討を実施した。その結果、本システムを構成する主要な機能及び自律型運転制御システムと自律ロボットとの関係を考慮に入れた概念構築を行うこととともに、本システムと外部システムとの情報伝達の明確化を行った。今後は各機能を試作し、全体システムの検証を行う予定である。

---

*　大洗工学センター基盤技術開発部先進技術開発室
**　現在、敦賀事務所

COMMERCIAL PROPRIETARY
PNC TN9410 97-049
April, 1 9 9 7

Conceptualization of An Intellectual Maintenance Management System

Kazunori SUDA*, Tsuyoshi YONEKAWA*
Shinji YOSHIKAWA*, Kenji OZAWA**

Abstract

It is studied in many research institute to enhance availability and safety of nuclear power plants operation and maintenance. On this account, development of autonomous plants has been carried out to replace the role of operators with artificial intelligence and autonomous robots. We have been developing an intellectual maintenance management system since 1994.

As the first step, concept of an intellectual maintenance management system was constructed. The intellectual maintenance managerial system is in charge of maintenance function of an autonomous plant. The intellectual maintenance managerial system has three functions which is monitoring state and judging abnormal machine and deciding maintenance plan by autonomy. This system has an important role of indication and communication of the result to an autonomous operation system and autonomous robot. In this examination, we adopted the distributed and cooperative system technique by multi-agent of AI technology and examined a method to enforce problem solving by cooperation of sensor and actuator. In this report, we examined trouble detection and troubleshooting evaluation and maintenance plan decision function by the distributed and cooperative system technology, the distributed and cooperative system communication-function that these information releases functions was taken on.

In conceptualization of the intellectual maintenance managerial system, we clarified of major functions to constitute this system and relation between autonomous operation system and autonomous robots. We clarified the information exchange scheme between this system and an outside system furthermore. In future, we will prototype each function and inspect the total system.

---

* Frontier Technology Section , Advanced Technology Division , Oarai Engineering Center
** Current affiliation : Turuga Office

目　次

図　目次

添付資料

表　目次

付図　目次

1．緒言

原子力プラントの安全性、信頼性、稼働率の向上を目的として、運転制御の高度化、知能化に関する研究開発が国内外で精力的に進められている。わが国における代表的な試みとしては、運転員に対し、プラント状況に関する適切かつ過不足ない情報を提供し、安全性、信頼性の向上を図るという知的運転支援システムの開発がある。しかし、あくまでも運転員を支援するものであって、最終的な判断、操作は運転員に委ねられている。自律型プラントの開発は、プラント運転・保守において人間が果たしている役割を人工知能やロボットに極力代替させ、最終的な運転操作までを人間に依存することなく人工知能が行うことを目指している。その自律型プラント開発の要素技術として、人工知能を備えた自律型運転制御システムの開発[1],[2]及び知的保全管理システムの開発[2]を行っている。

知的保全管理システムの開発の第１段階として、知的保全管理システムの概念構築について検討を行った。そこでは自律型プラントの全体構成及び知的保全管理システムの役割、ＦＢＲプラントの代表的な系統を対象にした知的保全管理システムの基本構成、知的保全管理システム内の通信方式、機能及びシステム構成について検討を行ったので、その結果について報告する。

2．自律型プラントにおける知的保全管理システムの役割

知的保全管理システムの役割は、既存プラントにおける保守員の役割を自律型プラントにおいて人工知能等を用いて自律的に行わせることである。すなわち、知的保全管理システムは、運転員となる自律型運転制御システム及び保守員となる自律ロボットとを有機的に結合して、プラント内の保全管理に係わる状態の監視、機器の異常・故障の判断及び保守・補修に係わる保全方策の策定を自律的に行い、その結果を双方に指示・伝達するという重要な役割を有している。

自律型プラントは、大別して自律型運転制御システム、自律ロボット、知的保全管理システム及びマンマシンインタフェイスで構成し、各システムは、図１に示すようにシステムレベルネットワークがゲートウェイを介してプラントレベルネットワークに接続され、必要な指示・伝達情報の授受を行う。自律型プラントでは、センサー・アクチュエータ等の機器レベルから制御・診断モジュール並びに人間の頭脳を代替する意思決定システム及び知識ベース間の情報交換は全てネットワークを介して行っている。原子力プラントのような大規模システムにおいては、全ての情報交換をひとつの共有エリアにて行った場合、システムの部分故障もしくは情報伝達の遅延により、制御・診断情報等の伝達さらにはプラント運転が出来なくなる可能性が考えられる。よって、システムににおいての危険分散及び情報交換を最小限に抑えるため、分散型のシステム構成かつ階層構成とした。階層構成にしたときの情報伝達のボトルネックになる階層間の情報伝達はゲートウェイにて必要な情報だけデータのやり取りを行うものとする。

知的保全管理システムと自律ロボットとの関係は、(1)知的保全管理システムにより作成された作業工程、作業手順等を、保全作業要求の形で自律ロボットへ通知する、(2)自律ロボットは、通知された情報を行動基準として保全作業を実施する、(3)自律ロボットによる作業の結果は、知的保全管理システムへ通知され必要に応じて作業工程あるいは作業手順の再構築を行い、その結果を自律ロボットへ再通知する、というものである。

一方、自律型運転制御システムとの関係は、(1)知的保全管理システムによるセンサ・アクチュエータレベルでの故障原因同定結果の報告や保守・補修完了後の試運転の要求を行う、(2)自律型運転制御システムは、保全要求、試運転結果や機器レベルの発生事象に関する情報を知的保全管理システムへ通知する、というものである。

なお、自律型プラントにおけるマンマシンインタフェイスの役割は、プラント管理者が緊急時を含むプラント状態に対応して自律型運転制御システムや自律ロボットに適切な指示・命令を与える際に必要な自律型プラントの状態に関する情報を的確に提供するためのものである。

3．知的保全管理システムの基本概念

自律ロボット自身が空間移動機能だけでなく、保全計画策定のためのアルゴリズムをも保持し、ロボット間で通信しながら計画立案することを想定した場合、個々のロボットは、まずプラントの異常・故障箇所の診断・同定する機能を備えていなければならない。これらの診断・同定機能には、高度な診断知識と大規模なプラントデータが必要である。一方、個々のロボットには、自律走行するための環境理解のための画像認識や、目標物体までの到達計画（プランニング）など大量かつ複雑なデータ処理が必須となる。例えば、自律ロボットは自身のまわりの局所的な環境情報しか持たないため、複数のロボットが協調して保全箇所を修復する場合、他ロボットとの間で対象物の画像理解や特徴抽出結果に矛盾が無く、整合性のあるデータ処理を実現する必要がある。このように、自律ロボット群間の整合性のある行動のため、情報処理に加えて保全計画策定まで実行させることは過度な処理負荷をロボットに与える恐れがある。

以上の観点から、センサ・アクチュエータ・静的機器の故障診断及び保全方策策定のプラニングや保全順序のスケジューリング機能は知的保全管理システムが受け持つことが望ましく、その考え方のもとで概念構築を行った。

3．1知的保全管理システムの基本構成

知的保全管理システムは、図2に示すように故障検知機能、故障診断評価機能及び保全方策策定機能の3つで構成する。それらの機能のうち、保全方策の策定を実行する機能エージェントは、原子炉系、1次冷却系、2次冷却系、水・蒸気

系及び蒸気発生器等の機器のサブシステム単位に分割する。故障検知及び故障診断評価を行う機能エージェントは、センサ、アクチュエータ、あるいは蒸発器、過熱器等の静的機器ごとに割り付けられ、それぞれの機能エージェント間で協調して局所的な情報交換を行い、保全方策策定を行う機能エージェントへ点検保守に関する保全管理データを提供する。また、保全方策策定を行う機能エージェントは、故障検知及び故障診断評価の結果とその他プラント運用上の制約事項及び定型的な保全作業スケジュールデータを記述した保全計画に基づき、作業工程、作業手順を作成する。この際、１つのサブシステムに割付けられた保全方策策定機能エージェントだけでは、大局的な制約を踏まえた上での計画を立てることはできず、他のサブシステムに割り付けられた保全方策策定のための機能エージェントとの協調が必要となる。

知的保全管理システムは、図２に示すようにマルチエージェントシステムとし、故障検知及び故障診断評価を行う以下の３種類の機能エージェントと、各サブシステム単位で分割された保全方策策定エージェントで構成される。

①センサ評価ｴｰｼﾞｪﾝﾄ：センサの特性評価と点検の必要性の判定を行う。

②アクチュエータ評価ｴｰｼﾞｪﾝﾄ：アクチュエータの特性評価と点検の必要性の判定を行う。

③静的機器評価ｴｰｼﾞｪﾝﾄ：静的機器の特性評価と点検の必要性の判定を行う。

３．２知的保全管理システムの機能

(1)故障検知機能

センサエージェントは、温度センサー、圧力センサー等、その種類の数だけ評価エージェントを有し、プラントから伝達されるプラント状態（状態変数名と値の組）を受信してプラント故障の発生を自律的に検出する。すなわち、自身が受信する状態変数の集合の中から、自身に必要なデータのみを選択受信し、自エージェント内の検出アルゴリズムでデータ処理を行う。各評価エージェントは互いに非同期に動作し、各自の評価アルゴリズムに従ったデータ処理を終了した段階で、評価結果を診断監視データベースエージェントに送信する。故障発見から、故障の状況を故障診断評価エージェントに送信するまでの故障検出機能構成を図３に示す。

(2)故障診断評価機能

故障診断評価エージェントは、センサエージェント群から送信される故障箇所状況と設計・保守データ等を相互に比較して、診断を開始する。診断が終了した段階で、自律型運転制御システムのプラント管理システムに対して運転の停止・継続・出力低下などのプラント状態設定・運転を要求するとともに、保全方策策定エージェントに保全方策策定を依頼する。また、一定時間内に診断

に必要なデータが揃わない場合、それまでの途中の診断結果に基づき、プラントの状態変更の指示あるいは他データの要求等を行う。図４に故障診断評価機能構成を示す。

(3)保全方策策定機能

保全方策策定エージェントは、故障診断評価エージェントからの診断結果に基づく保全仕様を受信して保全方策の策定を実施する。保全方策策定は、候補ロボット群の探索、ロボット群の保全スケジューリング、対応ロボットに対する作業指示の３ステップで行う。図５に保全方策策定機能構成図を示す。

①ステップ１

故障診断評価エージエントからの診断結果にもとづいて候補ロボット群を探索するステップである。故障診断評価エージェントから送信される保全仕様に基づき、その仕様を実現するために必要な自律ロボット候補を探索する。そのため、保全仕様を自律ロボット群に伝達する。ロボットへの要求仕様内容は、保全箇所（場所、大きさ等）、保全内容（溶接、キズ探査等）等である。この仕様を受信したロボット群は、自身の機能に照らし合せて、手を挙げる。具体的には、自身の機能、保全開始可能時刻、保全に要する時間、他機能との制約関係、現在位置、現場到着までの所要時間等の情報を故障診断評価エージェントに返信する。ここで、他機能との制約関係とは、溶接ロボットを例に取れば、キズ面が研磨された後でしか溶接してはいけないというような制約条件である。

②ステップ２

受信した複数の候補ロボットからの情報に基づき、それらロボット群の操作順序を策定する（保全スケジューリング）。操作順序は、各返信情報に記述された他機能との制約条件や所要時間、開始時刻等を勘案して決定する。

③ステップ３

保全スケジューリングが決定された段階で、対応ロボットに作業実行を指示する。ロボットには作業進捗を逐次報告させ、所期の時間内で保全が終了するかどうかの推論を同時に行う。また、知的保全管理システムには、リアルタイムでプラントに関する最新情報が入っており、場合によっては先のスケジューリングで割り当てたロボットの作業を中断させても新規の作業を優先させる必要が発生することがある。そのため、保全方策策定エージェントは必要に応じて再スケジューリングを動的に実行することが要求される。

３．３知的保全管理システムの通信方式

知的保全管理システムは、故障検知、故障診断評価、保全方策策定の３機能を持ち、各機能をそれぞれに対応するエージェントで実現する。エージェント間の

情報伝達の方式としては、１台の計算機にて集中管理を行う情報の共有（共有メモリ）と分散型のメッセージ（放送）がある。ここで、共有通信方式とメッセージ通信方式の知的保全管理システムへの適用に関する検討を行う。

(1)共有通信方式

共有通信方式の代表的なものとして、図６に示す黒板モデルがある。黒板モデルとは、音声理解システムHEARSAY-Ⅱ[3]で用いられた知識の制御方式であり、主に知識源と黒板（共有メモリ）からなる。黒板に書き込まれる仮説は、処理レベルに関係なく一様な構造を持ち、統一的な方式に従って評価値が与えられており、原則としてどの知識源からでも共通に通信（アクセス）できる。各知識源は自分自身が動作可能となるための前提条件を持っている。黒板上にこの前提条件を満たすものが現れたとき動作可能となり、黒板上の仮説を検証してその評価値を修正したり、自己の生成した新しい仮説を黒板上に書き込むことができる。このように、各知識源は黒板との間で仮説を交換しながら間接的に通信するので、それらの間には原則として従属関係はなく、独立に動作が可能である。各知識源は並列に動作しながら、互いに仮説の検証と生成とを繰り返し処理を行う。

共有通信方式は、計算機の共有メモリ空間を利用した通信方式であり、小規模なシステムにおいては非常に有効かつ簡便にシステムを構築することができる。しかしながら、大規模システム及び分散システムにおいては共有メモリに全てのエージェントがアクセスするので、ネットワークトラフィックやある一つの処理ができないために全体の処理が停止していまう可能性がある。

(2)メッセージ通信方式

メッセージ通信方式は、各メッセージ発信者がそれぞれ自身が所持しているデータを自身の繋がっているネットワークに発信し、必要なデータはネットワークから抽出することにより、情報交換を行うものである。例えば、テレビ局から情報が放送され各家庭は必要な番組を選択して見ていると同じように、各センサー・アクチュエータエージェントが自分のデータを放送によりネットワークに発信し、必要なデータだけを抽出する。分散化されたエージェントにデータを放送することで共有通信方式と比較してデータが一カ所に集中することなく通信が可能であり、通信トラフィックを解消できる利点がある。また、放送通信には同期・非同期を問わず行えるので、一部のデータからの送信待ち等による時間遅れを防ぐことが可能であり、分散システムに最も適した通信方式である。

以上の検討の結果から、大規模かつ分散化したデータを使用する必要がある知的保全管理システムではメッセージ通信方式を採用する。

３．４知的保全管理システムのデータベースの基本構成

保全作業を自律的に実施するためには、各機能エージェントが共通で使用することができるデータベースの構成をシステムの体系に見合ったものとすることが必要である。各データベースは、監視・診断・評価・計画を分散的にかつ協調して実行する各機能エージェントによって規約として与えられるものと、各機能エージェントが分散的に解釈した情報と、各機能エージェントが協調して得た情報とに大別される。

作業工程、作業手順に関する情報は、プラント全般に亘る機能エージェントが個々に協調してはじめて得ることができるものであり、プラント全体の情報空間に位置づけられる。一方、点検保守を目的とした監視・診断は、機器あるいは設備単位で行えば十分であり、性能劣化評価、寿命評価は機器あるいは設備を対象としたものであるため、サブシステム内でクローズして処理できる。

このことから、保全管理に係わる情報のうち、プラント全体で共有すべき情報である点検保守作業の行動基準となる作業工程、作業手順は、システムレベルであり、その他の情報はサブシステムレベルにて管理する構成とする。図７に知的保全管理システムの処理フローを示す。

３．５知的保全管理システムのシステム機構

各エージェントが、他のエージェントについてお互いに知らなくても、必要な情報の伝達を正しく実行できるシステム機構及びメッセージ通信を用いる分散システムについて検討する。

各エージェントが集中管理機構などの制約を受けることなく常に最大の能力を発揮できるように、知的保全管理システムに放送通信方式を採用した分散オペレーティングシステムを採用した。

ここでは、知的分散ＯＳ（IDPS）について検討を行う。IDPSが動作するシステム環境を図８に示す。ＡからＨはエージェントを表し、複数台ある計算機（ＷＳ：Work Station）は、ＬＡＮ（ローカルエリアネットワーク）で接続され、各ＷＳにはIDPSが搭載されている。IDPSはＯＳ核と呼ばれる部分と種々のサービス機能を提供するエージェント群から構成されている。ＯＳ核はエージェントの生成、登録、削除やエージェント間通信のような基本機能を持っており、アプリケーションプログラムをエージェントとして作成・登録したり、アプリケーションエージェント間のメッセージ交換を司っている。不特定多数のエージェントが動的に生成され自律的に動作するためには、それらの相互間でメッセージ交換機能が必須となる。IDPSは信頼性の高い放送通信機能を備えており、かつシステムの多重化等の高度化も柔軟に対応できるシステムである。よって、知的保全管理システムのエージェント群が動作するための基盤機構として採用することとす

る。

4．結言

自律型運転制御システムと自律ロボットとを有機的に結ぶ知的保全管理システムの開発を実施している。本報告では、自律型運転制御システム及び自律ロボット間の情報伝達、及びプラント機器の故障検知機能、故障診断機能、プラント機器の故障保全、定期保全において必要となる保全方策策定機能の概念検討を行い、知的保全管理システムの機能の明確化を行った。また、分散協調マルチエージェントシステム、知的分散ＯＳを使用した通信システムを採用することにより、大規模分散環境における知的保全管理システムの開発の方向性を示すことができた。

今後は、本報告書で記した基本機能の詳細検討、分散化されたエージェント間を協調制御する分散協調機構の検討、センサ・アクチュエータ部の異常を判定するセンサバリデーション機構の検討に取り組む予定である。

5．謝辞

知的保全管理システムの概念検討に当たり、通信方式検討及び分散システム構成において協力して頂いた原子力システム株式会社瀬谷義一氏、山本裕史氏の両氏に感謝の意を表する。

6．参考文献

［1］小澤健二他、電気学会原子力研究会資料、”知的運転制御システムの開発”、NE-95-15(1995)

［2］小澤健二、”知的運転制御システムの開発”、原子力工業、Vol.42-No.5(1996)

［3］石田享、”分散人工知能の技術と応用”、人工知能学会、Vol.5,No.4,pp.441-448

［4］清水謙多郎、”分散・並列ＯＳの機能と実現技術”、トリケップス、No.16

［5］関俊文他、”オブジェクト指向分散システムにおける放送待機冗長処理方式”、T.IEE Japan,Vol.114-D,No.3

［6］仁賀博一、”総合保全システムの開発と適用効果”、日本鉄鋼協会、pp17-52

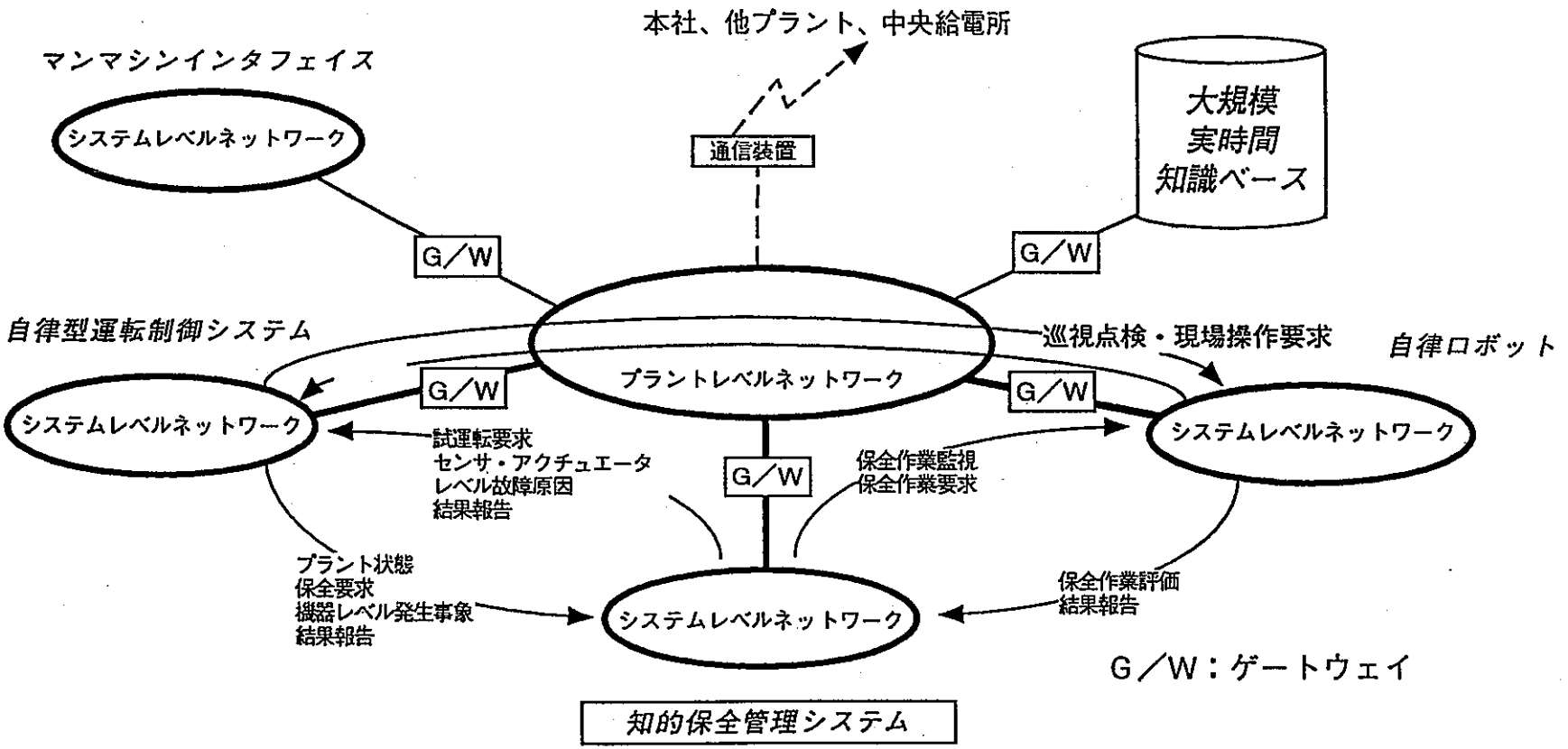


図１　自律型プラント全体構成図

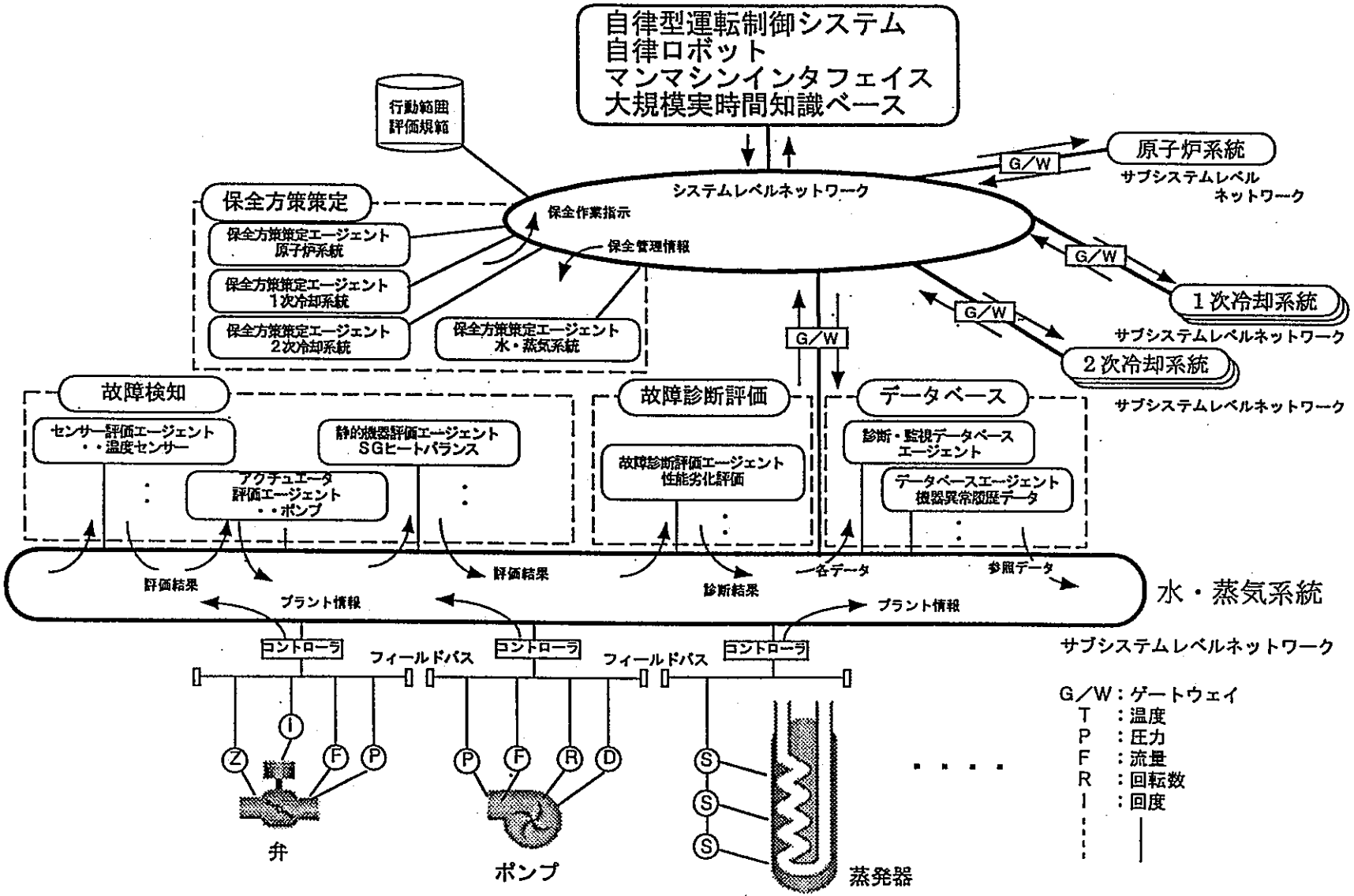


図2　知的保全管理システムの構成

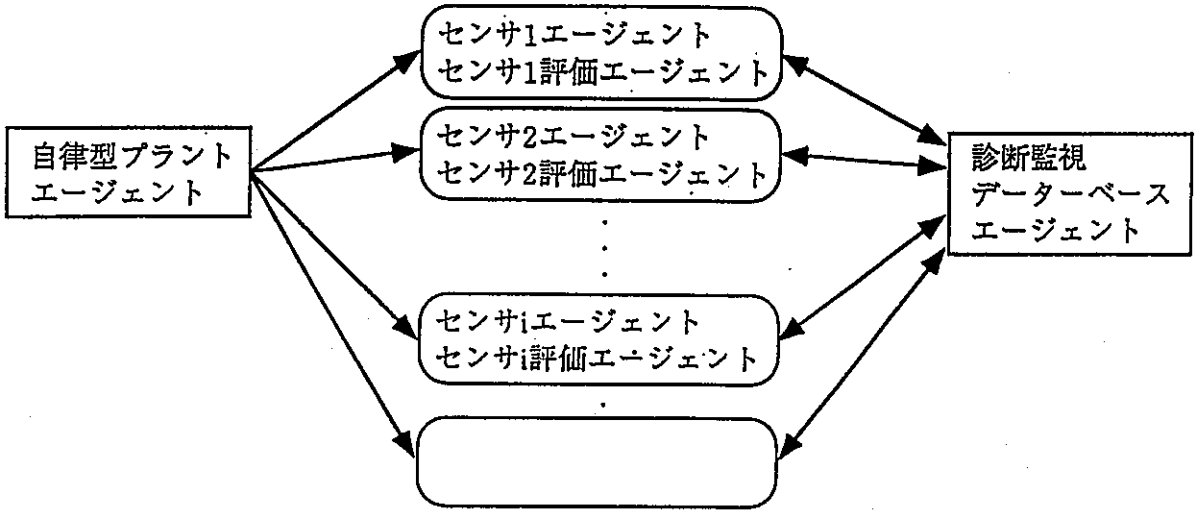


図３　故障検知機能構成図

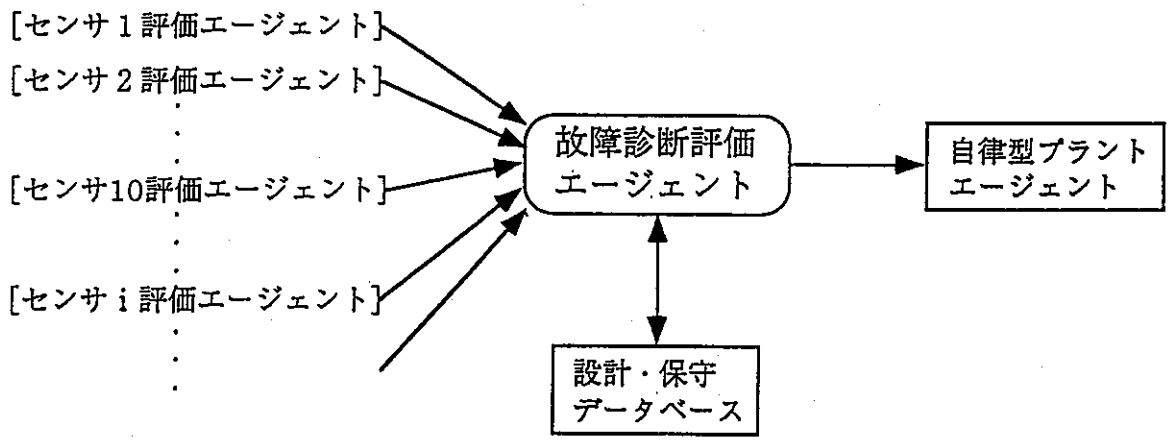


図４　故障診断評価機能構成図

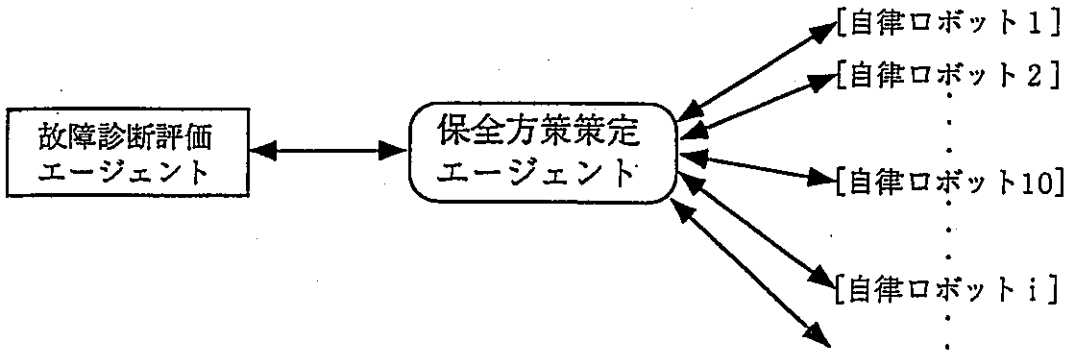


図５　保全方策策定機能構成図

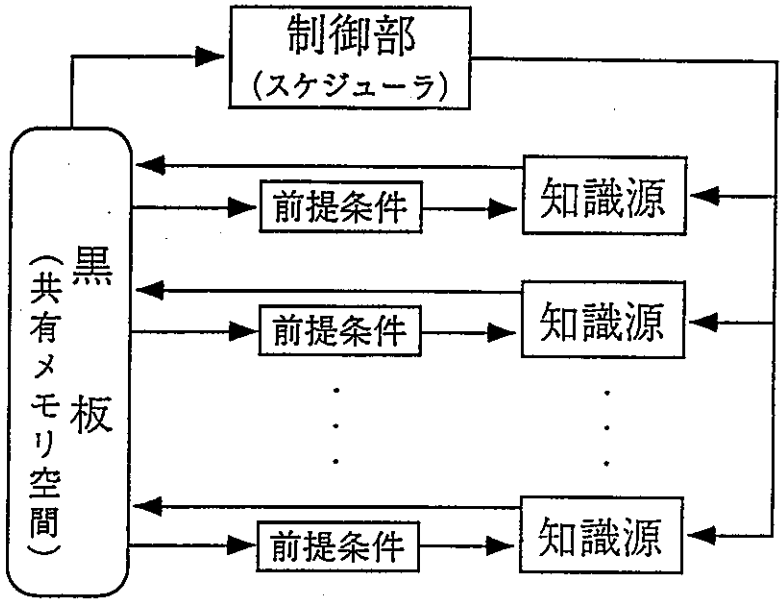


図6　黒板モデル

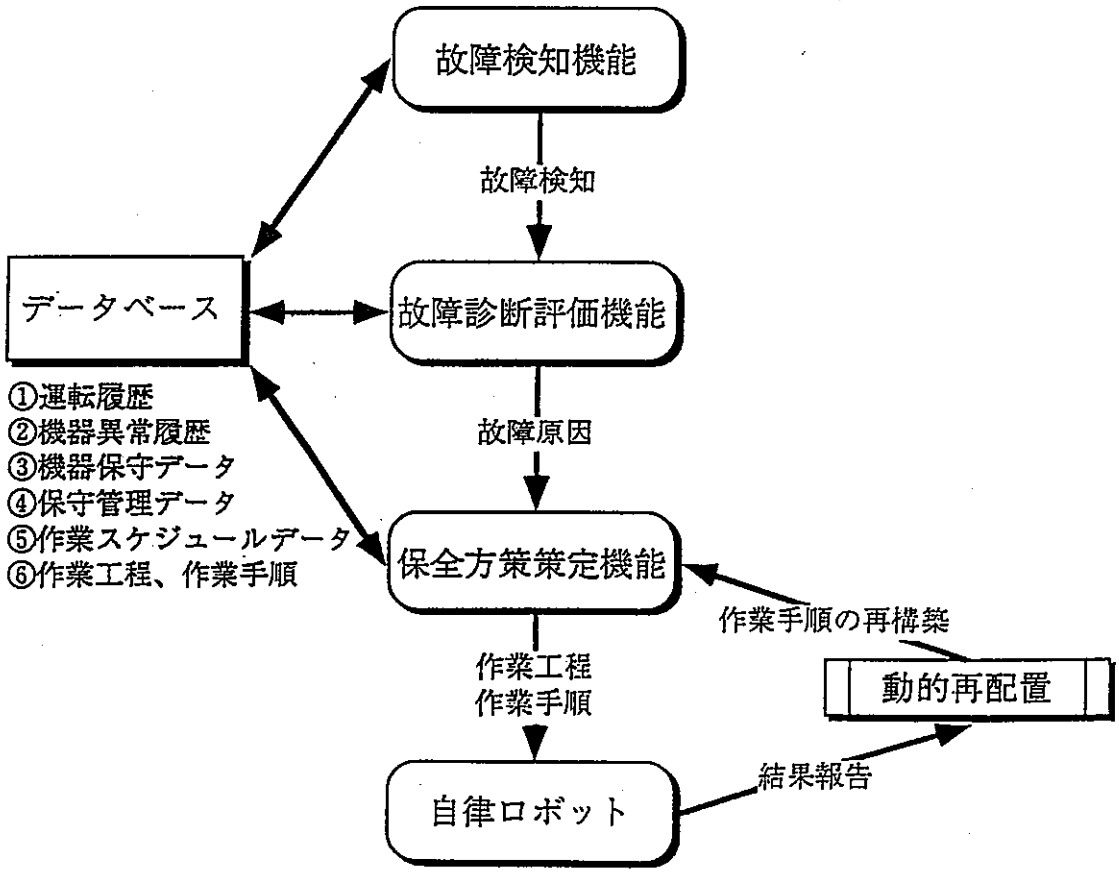


図7　知的保全管理システムの処理フロー

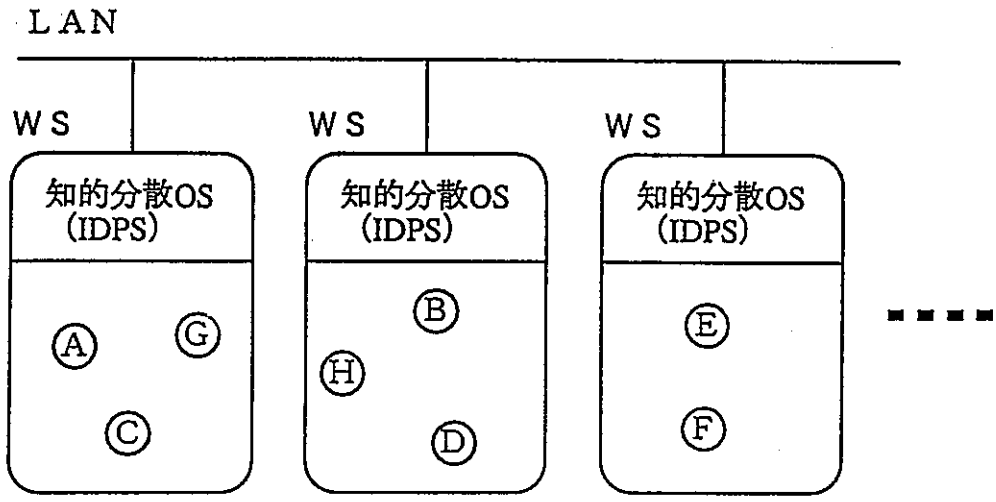


図８　知的分散ＯＳが動作するシステム環境

添付資料

1．分散ＯＳの研究開発状況

知的保全管理システムの概念構築に当たり、システムのロバスト性を高めるべく分散協調機能を有するシステムとして検討を行った。その分散システムの検討において調査した先進的な分散ＯＳの研究開発状況及び特徴を付表１に示す。

付表１　並列・分散ＯＳの特徴（1/2）

OS	V-Kernel	Sprite	Mach	ToM	DPS
開発者、拠点	スタンフォード大学 メインフレームの計算機パワーとタイムシェアリングによる経済性、パソコン群からなるシステムの拡張性、信頼性の良さを兼ね備えたシステムを構築する。	UCバークレイ ネットワークと大容量メモリ、マルチプロセッサの性能を最大限に生かすシステムを構築する。タイムシェアリングシステムにおける資源共有と通信の長所を保ちながらネットワーク透過な分散システムを構築する。	CMU 初期Ｕｎｉｘの単純な設計、変更のしやすさ、新機能の追加が短期間で可能という特徴を受け継ぎマルチプロセッサ化、ネットワーク化、メモリ大容量化に対応できるＯＳを構築する。	京都高度技術研究所（ASTEM RI） ネットワーク透過な環境を実現し、分散した資源の共有と有効利用を実現する高信頼で小さなカーネルを持つネットワークシステムを作る。	東芝（S&S Lab.） 分散システムが持つ拡張性、信頼性、適応性、保守性を最大限に生かせるシステムを構築する。
Ｕｎｉｘとの互換性	非互換	互換 カーネルコールは、4.3BSD UNIXと似ている。	互換 4.3BSD UNIXとバイナリコンパチ	互換 UNIXコンパチブルなインタフェースをライブラリの形で提供する。	非互換
マルチプロセッサ・アーキテクチャ	ネットワーク上の分散ＯＳ NUMA（Non Uniform Memory Access）	ネットワーク上の分散ＯＳ NORMA（No Remote Memory Access）	色々なマルチプロセッサアーキテクチュに対応可能 UMA（Uniform Memory Access）	ネットワークで結合された異機種マシンを対象、マルチプロセッサも対象としている NORMA、UMA	ネットワーク上の分散ＯＳ NUMA
設計目標	・高速なプロセス間通信 ・小さなカーネルでネットワーク透過なプロセスとメモリ通信を実現【複雑な処理はカーネル外のサーバプロセスで実現。カーネル機能の多くもカーネルサーバとして実現。】	・UNIXと互換 ・ネットワーク透過なファイルシステムを実現 ・単一ＷＳ上でプロセス間共有メモリを実現 ・プロセスマイグレーション ・大容量Cache	・密／疎結合の汎用マルチプロセッサのサポート ・ネットワーク透過なリモートファイルアクセスのサポート ・スレッドの導入 ・ケーパビリティを基本とするプロセス間通信機能 ・大量データ転送を可能にする仮想記憶管理機能	・分散環境に適したプログラミングモデルの提供 ・異機種及在で統一的分散環境 ・高いセキュリティと使いやすいユーザ環境 ・ユーザごとに最適なファイル操作を環境が定義	・ＯＳ機能をＯＳ核とオブジェクトの共通知識に分離する事により、集中管理機構を持たない管理分散システム ・システムワイドな管理オブジェクトの共通知識の結合で実現 ・システムレベルでも多重度、位置独立なオブジェクト ・ネットワーク透過なファイルシステムを実現
軽いプロセス	・プログラム実行主体はプロセス ・プロセス間でアドレス空間のメモリスペースコストを削減【プロセスをプロセス記述子とアドレス空間記述子に分離することで節約】 ・高速なプロセス生成【生成は新しいプロセス記述子の割り当て問題に帰着される。microVAXIIで約5msec】	・プログラム実行主体はプロセス【但し、プロセス間でデータ共有ができる。全部共有か共有しないかが選択できる。】 ・カーネルはマルチスレッド【カーネルは小さな機能部分の結合であり、複数のプロセスが同時にカーネルサービスを受けられる。マルチプロセッサで特に有効。】	・プログラム実行主体はスレッド ・タスクは複数スレッドと持てる。Proces =Task + N * Thread; ・同一タスク中のスレッドが別々のCPU上で並列に実行可能。 ・１タスク内で複数スレッドの並列実行可能。 ・共有メモリを通信に使える。	・プログラム実行主体はスレッド ・モジュールはプログラムと抽象化したもの。 ・モジュールはコードとデータを持つ。 ・スレッドはプロセッサを抽象化したもの。 ・スレッドはスタックとレジスター値を持つ。 ・スレッドは複数モジュール間を渡り歩く（RPCしても同一スレッド）	・プログラム実行主体はスレッド ・オブジェクトはプログラムの独立性を高めたもので、複数のメソッドからなる。 ・メソッドはコードとデータを持つ計算機の機能単位。 ・メソッドは複数スレッドを持ち、並列実行可能。
プロセス間通信	・高速、透過なトランスポートレベルサービスをカーネルをカーネルが支援　VMTP ・高速化 (1)32Byte固定長メッセージ【システムコールに汎用レジスタを使用】 (2)クライアントのSend、Receiveを単一のプリミティブで実現【再スケジュール、バッファリングのオーバーヘッドを削減】 (3)各プロセスがカーネル中にVMTPヘッダのテンプレートを持つ。【ヘッド書込、メッセージスペース確保の省略】 ・マルチキャスト (1)グループID（多重グループ） (2)Logical addressing (3)Qualifier（指定プロセスの存在するサイトに送信） ・データグラム通信 Forwarding、ストリーム通信、Security、Priority支援	・カーネルtoカーネルRPC（Stub＋RPC transport） ・Implicit Acknowegement & fragmentation【RPC のResponseでACKを兼用、大量通信時は最後にまとめてACKを返送】	・Port、Messageの概念に基ずく。【記憶独立、安全性、データタイプ・タギング機能】 ・Port：プロテクト付きカーネルオブジェクト（メッセージのための有限長キュー） ・Message：固定長ヘッダ＋可変長の型付きデータオブジェクトの集合（ポートアクセス権、ポインタも送信可） ・非同期／同期通信の両方 ・Copy-On-Writeによる効率よい通信。（大量メッセージの転送を可能にする。） ・ＯＳカーネルはサイト間通信をサポートしない。メッセージサーバが取り次いでネットワーク透過にしている。 ・ネットワークPort：計算機単位のPort ・ネットワークサーバがローカルPortとネットワークPortの対応を管理する。 ・ネットワーク上のメッセージは暗号化	・独自の（単純な）RPCをサポート ・RPCで呼び出すモジュールが別の計算機上にある場合には、スレッドが自動的に他の計算機に転送される。	・ネットワーク透過 ・非同期放送通信（ACK無し） ・直接通信（宛先オブジェクト名、データの直接指定） ・放送範囲限定モード (1)自計算機内を対象 (2)全計算機を対象 (3)自計算機以外の全計算機を対象 ・オブジェクト名階層関係の導入（放送通信の効率化） ・別名定義可能 ・Message：固定長ヘッダ＋可変長の型付きデータの集合 ・Priority支援
メモリ管理	・仮想メモリ（ネットワーク透過が目標）【アドレス空間はRegionの集合からなり、各Regionにファイル（UIOオブジェクト）が対応付けられている。ページフォルトが起こると対応ファイルの内容がページキャッシュに入る】 ・Demand Paging とNon-Demand-Pagingを支援。 ・ページキャッシュの整合性は、Backing Server上のLock Managerが保証する。【簡単なOwnershipプロトコル、Problem Oriented Shaed Memory】 ・Read/Writeのようなパフォーマンスクリティカルな処理はカーネルが実行。ディレクトリ管理、ディスク割当、アクセス制御等複雑な処理はプロセスレベルのサーバが実行。 ・File-like Read/Writeアクセスもサポート ・シングルレベル・ストレージ（Memory Mapped File）	・仮想メモリ ・スワッピング、ページングに通常のファイルを使用。【File Systemの機能がそのまま使えて、WS間でスワッピング、ページング領域を共有できる】 ・仮想メモリとファイルキャッシュで物理メモリを共有。 ・Sticky Segment【データセグメントは利用終了後も暫くキャッシュに残す。但し内容が変えられた場合は捨てる。】 ・クライアントとサーバの両側に大量のキャッシュを設ける。 ・ファイルサーバがバージョン番号を持つことによりキャッシュの整合性を保証する。【キャッシュとサーバのバージョンが異なる時には再ロードする。複数のプロセスが同時に書込を行う時にはキャッシュは使わない。】 ・シングルレベルストレージ	・ネットワーク透過仮想メモリ (1)移植性（アーキテクチャ独立） (2)拡張性のあるアドレス空間操作（データ構造を単純で小型に） (3)密結合マルチプロセッサのサポート（Read/Write共有メモリ、粒度の細かいロッキング） (4)パフォーマンス（Copy-On-Write　Copyをなるべく遅らせる） ・アドレス空間＝ページ列 ＊論理ページサイズ＝2^n（任意サイズ設定可能） ＊インヘリタンス（Shared、Copy、None） ・メモリオブジェクト： ＊タスクの仮想記憶に割付る事のできる資源（一般にはファイル） ＊タスクアドレス空間内に写像 ＊参照キャッシュ（参照の頻繁なものはキャッシュに保存） ・影オブジェクト：Copy-On-Writeで変更のあったページを管理している。 ・外部ページャ：ページフォルト及びページアウトを扱うタスク（Memory Mapped File） ・共有メモリサーバ： ＊ネットワーク共有メモリをサポート ＊複数クライアント間で整合性を取る ＊全Read-only or 1つだけRead/Write mode	・仮想メモリ	・仮想メモリ ・オブジェクトごとに関連情報を１つの仮想連続アドレス空間に割付け、オブジェクトのメモリ空間上での分離・独立を実現。 ・アプリケーションオブジェクトのアドレス独立（計算機内／外へ移動 or 複写してもアドレスの再リロケーションは必要としない） ・アドレス空間＝ページ列 ＊論理ページサイズ＝2^n（現在n=16、１ページ=4kb）

付表1　並列・分散ＯＳの特徴（2/2）

OS	V-Kernel	Sprite	Mack	ToM	IDPS
分散ファイル	・Block level access［ディスクへのアクセスがバイト単位でなく、データブロック単位で行える。］	・ネットワークで1つのディレクトリ ・Block level access ・Cache in main memory ・ファイル名のネットワーク透過性を完全に実現［各EWSのファイルサーバはファイルの絶対パス名とPrefix Tableは放送問い合わせで動的に作成・変更］	・ＯＳカーネルレベルではサポートしていない。 ・AFS（IBM）を取り入れる。	・ユーザごとにマウントテーブルを持つ。 ・ユーザ毎にディレクトリ構造を設定できる。（異なるネットワークからでも自身のマウントテーブルをセットすれば、前と同じ環境でファイルをアクセスできる。	・Disk Agent、Access Agent、File Agentの３つのオブジェクトによる実現 ・ファイルとAgentの位置透過性 ・ファイルとAgentの多重度からの透過性 ・ファイル単位の分散 ・ファイル単位での多重化 ・Agentの負荷分散 ・Agentの並列動作
プロセス・マイグレーション	・カーネルが支援 ・プレコピー法［プロセス走行中にコピー、コピー後プロセスを停止して、変更ページを再コピー、プロセス停止は、プライオリティを特殊なレベルに下げる。］ ・負荷分散はプロセスレベルのサーバが行う。	・仮想メモリ機構がサポート［旧EWSがプロセスをページアウト、新マシンがページフォルトで全データをローディング］ ・ユーザレベルでネットワーク透過［各プロセスはhome nodeを持つ。Node dependentな処理はhome nodeで行う。	・ＯＳカーネルレベルではサポートしていない。 ・Agentでは、オンデマンドでページ単位にメモリ領域をコピーする。	・ＯＳカーネルレベルではサポートしていない。	・カーネルとオブジェクト管理によって実現。 ・システム停止時間の最小化（転送バッファ、受信メッセージのキューイング）
トランザクション	・マルチキャストを利用したコミットメント処理 ・多重資源管理		・ＯＳカーネルレベルではサポートしていない。 ・オブジェクト指向インタフェースCamelotによって、アトミック操作を行う。	・ＯＳカーネルレベルではサポートしていない。	・２相ロック方式 ・分散デッドロック防止、検出併用プロトコル ・メッセージ到着順序保証 ・多重資源管理
プロテクション		・特になし	・ページ単位プロテクション（Read、Write、Execution） ・Portにアクセス権（Receive、Send、Owner） ・サイト間通信メッセージを暗号化	・モジュールのアクセス制御はハウス&ユーザとケーパビィリティによる ・ネットワーク内の計算機へのアクセス制御はワールドによる。 ・各ハウスはユーザリストを持つ。 ・ハウスのユーザはハウス内でモジュールやスレッドの生成／削除ができる。 ・ケーパビィリティはモジュールの各エントリのcall権である。 ・エントリのケーパビィリティうい持つスレッドはそのエントリをcallできる。 ・エントリのケーパビィリティを持つハウス内のスレッドはそのエントリをcallできる。 ・ネットワーク内のノードの部分集合をワールドとし、各ユーザはアクセス権を持つワールド内のノードにアクセスできる。	・リンクプロテクション（ＯＳ核、システムオブジェクト、ユーザオブジェクト） ・セグメンテイションによる他オブジェクト領域破壊防止
リライアビリティ	・多重資源管理				・ファイルストップオブジェクト（放送並列多重処理） ・ファイルストップ放送通信（通信累計量によるチェック）
リアルタイム処理	・データグラム通信 ・Non demand paging		・別にリアルタイムMach、分散リアルタイムカーネル（ARTS）を開発		・優先度付きマルチタスクスケジューリング ・起動時刻、無効時刻の指定 ・割り込み起動メソッド
ネーミング	・平坦	・木構造（ファイル）	・木構造（ファイル）		・木構造（ファイル）
オブジェクトモデル			・採用		・採用

2．知的分散ＯＳ（IDPS）の機能と特徴

(1)エージェント・メソッド

知的分散ＯＳ（IDPS）では、アプリケーションプログラムだけでなくＯＳが提供するサービス機能もエージェントとして実現されるが、そのエージェントはプログラムの実行環境であり、物理的には1つの仮想アドレス空間である（仮想アドレス空間を占める実体の意味）。エージェントは複数のメソッドと呼ばれる手続きプログラム群とそれらの間で共有されるデータ群から構成される。

メソッドは、外部からアクセス可能なエージェント内の手続きのことで、その動作はメッセージ通信によって起動される。付図１はIDPSでのエージェントとメソッドの関係を示したもので、入力と記述されているのは他のメソッドからメッセージを介して、起動されることを表している。

(2)非同期放送通信

エージェント間通信は非同期型の通信と同期型の通信に大別されるが、IDPSでは非同期型通信を採用している。まず、非同期型通信であるか、これは要求エージェントがメッセージを送信したらただちに次の処理に移ることができる方式で、送信先のエージェントから応答が返信されるまで処理を中断して待つ必要がない。一方、同期型通信ではリモートプロシージャコール（PRC：Remote Procedure Call）のように、メッセージ送信元は応答が返信されるまで自処理を中断する、いわゆる要求・応答型の同期型通言方式である。

次にエージェント間通信あるいはエージェント聞通信を１対１通信と１対多通信で分類する。１対１通信は指定したエージェントやエージェントのみへメッセージを送信するのに対して、１対多通信としては特定グループ内に属するメンバーエージェントやエージェントに送信するマルチキャスト通信と不特定多数のエージェントやエージェントに送信する放送通信方式である。マルチキャストの範囲をシステム内すべての構成メンバーに拡大しだのが放送通信と言えるが、IDPSではこのマルチキャスト範囲を調整することができるようになっている。

(3)通信インターフェース

通信インターフェースとは、アプリケーションエージェントからみて、通信相手が自分と同じ計算機内（ローカル）にいるのかまたは他計算機内（リモート）にいるかによって、通信の仕方が左右されないように設計された通信インターフェースのことである。統一された通信インターフェースを実現するには、通信相手の位置によらないメッセージ送受信用の言葉を分散OSが提供する必要がある。この言葉のことをシステムコールといい、IDPSでは「Send」と「Wait」というシステムコールで統一している。「Send」の引き数は送信相手先エージェント名と送信するメッセージおよび計算機に関する情報から構成されている。計算機情報とは、

メッセージの送信範囲を自分の計算機内にとどめるのか他の計算機すべてに放送するのかといった指定ができるようになっている。例えば、送信相手が自分と同じ計算機にいれば、他の計算機にまでメッセージを送信する必要はないわけであるから、この場合は自計算機内にメッセージ到達範囲を限定しておくのか効率的である。一方、「Wait」は受信したいメッセージの名前を引き数としている。次に、メッセージ伝達方式であるか、２通りの方法をIDPSでは提供している。１つは、メソッド駆動と呼ばれるタイプで、付図２のようにエージェントAのメソッドmaがエージェントBのメソッドmbを起動してメゾソドmbにメッセージを引き渡す方法である。他の１つは、データ呼出と呼ばれるタイプで、相手エージェントBの共通データである変数Xに値をセットする方法である。相手エージェントでは、その変数値を利用したい自身のメソッドがシステムコール「Wait」でXを受け取ることができる。Waitを実行した段階でまだデータが到着していなければ、IDPSは自身の処理を中断して他の実行待ちメソッドに制御権利を渡す。データ到着と同時にIDPSはメソッドmbの「Wait」処理を再開させる。

IDPSでは、さらに放送しないマルチキャスト通信機能において、同一エージェント名を持つエージェント群にメッセージ伝達を一度に行うための機構が準備されている。エージェントを作成し、IDPSに登録する際に、付図３のようにエージェント名ＡからＨの間で名前の階層構造を同時に定義しておく。そうすると、実行時にエージェントA宛にメッセージが到着すると、そのメッセージはＡだけでなく、下位のエージェント群ＢからＨまで同時に伝達される。また、エージェントB宛にメッセージが到着したときは、Ｂ、Ｄ、Ｅに同時にメッセージが伝達される。この機構により適当なエージェントグループ範囲にメッセージ伝達を一度に行う事ができるようになる。

(4)エージェントの生成、移動、複写

ここでは、エージェントの動的な生成機能とエージェントの計算機（ＷＳ）間の移動や多重化のための複写（コピー）機能の特徴について述べる。まず、動的なエージェント生成機能としてインスタンスエージェントの生成機能について説明する。この機能はクラスエージェントと呼ばれている雛型のプログラムを１つ作成しておき、そのコピーを生成し、コピー（インスタンスエージェント）を特徴付ける初期値をコピーに与えることにより、必要な時に必要なエージェントを必要な数だけ生成することが可能になる。IDPSでは、ユーザからは見えないインスタンスエージェント生成・移動・複写を司る管理エージェントを提供しており、アプリケーションエージェントからシステムコール「Ｎｅｗ」を使って簡単にインスタンスエージェントを生成することができる。

次に、エージェントの移動と複写の機能としては、どのＷＳからどのＷＳに移動したかを指定するだけでエージェントの移動が行えるシステムコール

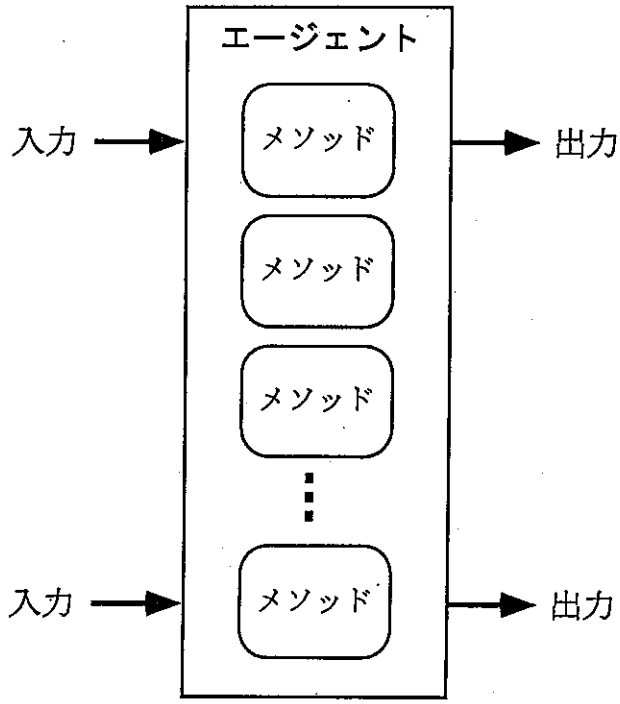
「Obj_move」と、どのＷＳからどのＷＳにコピーしたいかを指定するだけでエージェントのコピーが行えるシステムコール「Obj_copy」が提供されている。

(5)高信頼機構

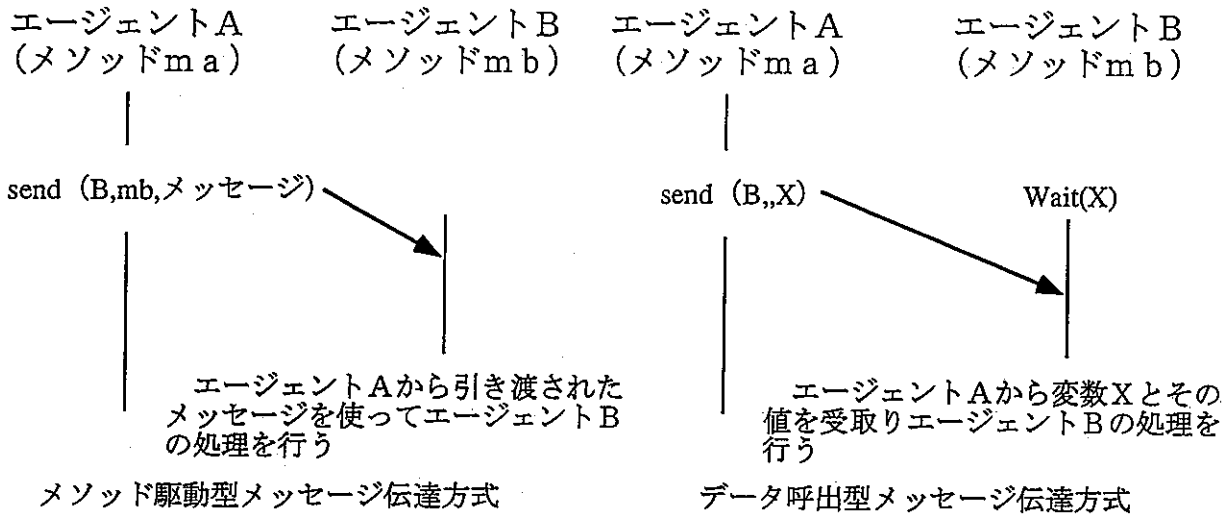
IDPSが提供する高信頼化機構について述べる。IDPSでは、放送による並列多重化処理機能を提供している。放送による並列多重化処理とは付図４に示すようにそれぞれ３重、２重、３重と多重化されたエージェントＡ、Ｂ、Ｃが同時に並列に動作している状況下で、エージェントＢへは３重化されたＡからメッセージを受信し、それらの中から正しいメッセージのみを選択し、さらにエージェントＣにメッセージを放送していく処理方式であるこの並列多重処理方式による高信頼システム構築は、瞬時のシステム停止も許されない場合に有効である。

付図４の方式では、個々のエージェントの処理結果が同じになることを保証することが必要になる。すなわち、多重エージェント間で同一の意志決定が必要になる。このためには、ＬＡＮ上に放送されたメッセージを全てのＷＳが同一順序で受信する通信機構が新たに必要になる。この通信機構は、「ＬＡＮ上に流された放送通信累積量（AMN：Accumulated Message Number）」をIDPSが監視し、この累積量に不一致が生じたならば、これを検出することで実現されている。付図５は、IDPSの同一意志決定機構Ｘ、Ｙ、Ｚの中で、ＸとＺがＹと異なる通信累積量を検出し、ＸとＺがＹに対して”Ｙが故障”と通知している様子を示している。Ｙは例えば、２台以上のＷＳから故障通知を受け取ると自身が故障と判断してシステムから離脱し、他ＷＳに害を及ぼさないようになっている。

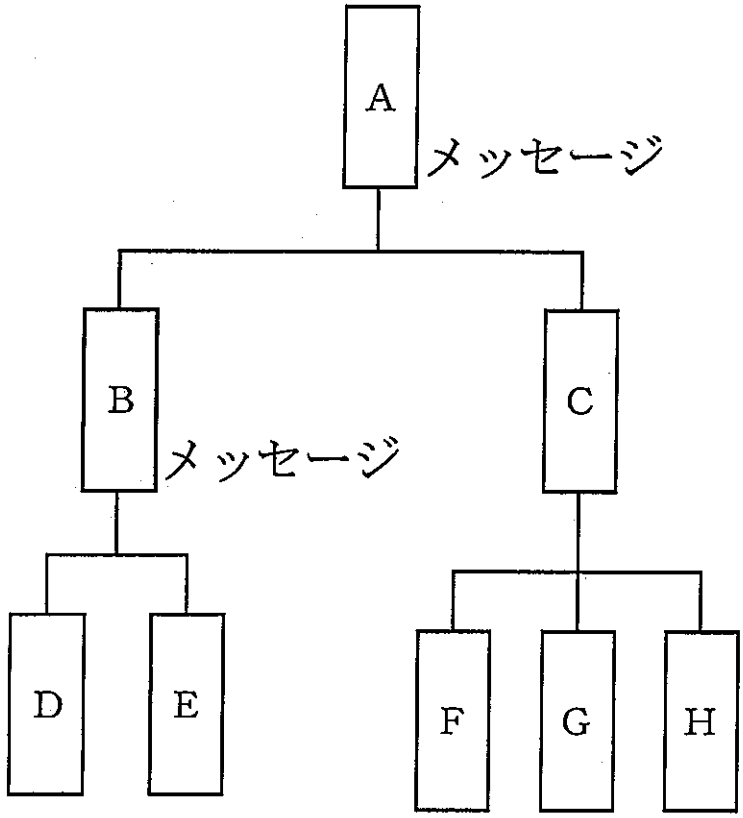
次に、多重化された並列に動作しているエージェント群からのメッセージの同定を行うために、ユニークなメッセージIDを発行し、それをメッセージに付加して放送してる。これで多重化メッセージの同定が可能になるわけであるが、同定された多重メッセージからどれが有効で正しいメッセージかを判定する必要がある。このため、IDPSでは、メッセージ有効数（ＣＮ値：Confirmation Number）と呼ぶ量を導入している。これは、エージェントの作成、登録時にそのエージェント自身のＣＮ値を定義することになっている。そして、実行時にそのエージェント宛に多重化されたメッセージが次々と到着したとき、同一メッセージID中で内容が等しいメッセージをＣＮ個受信したならば、その段階でそのメッセージを正しいメッセージと判定する。ここで、ＣＮ＝１の場合がいわゆる先着優先制御と呼ばれている方式である。以上の種々の機構により、経済的で高信頼なシステムを構築することが可能になる。



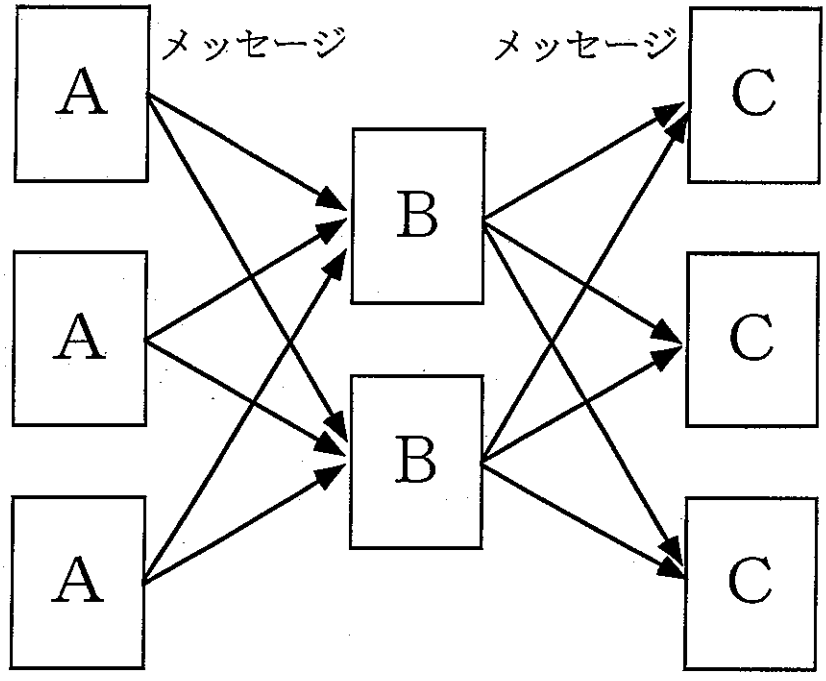
付図１　エージェントとメソッドの関係



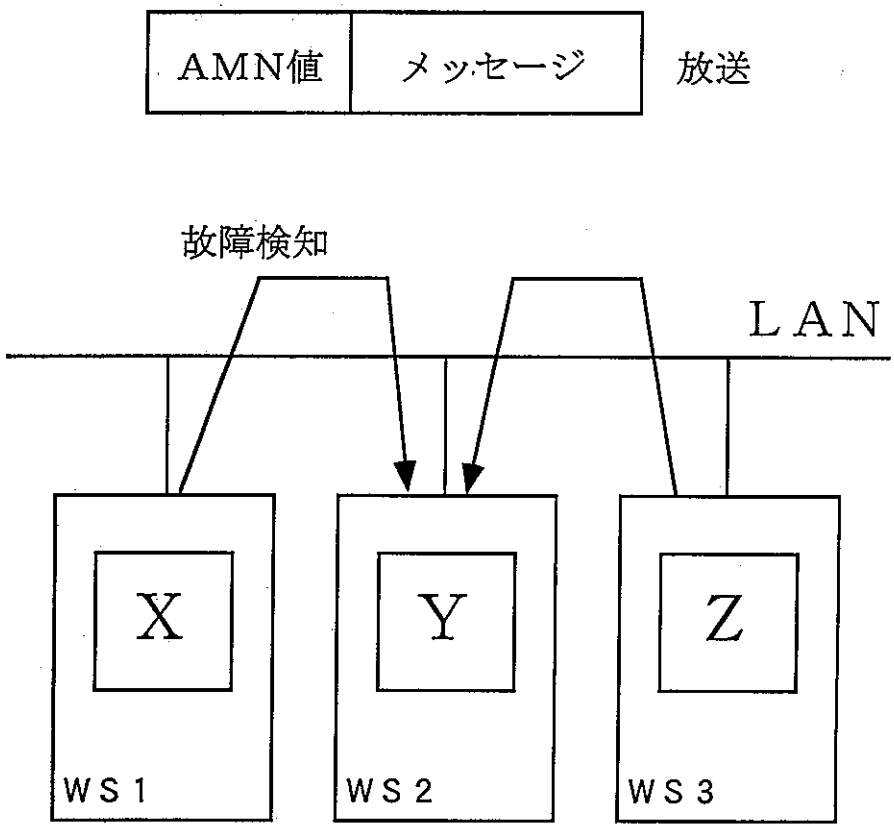
付図２　メッセージ伝達方式



付図３　放送通信範囲指定方式



付図４　放送による並列多重処理方式



付図５　故障検知方式