

マルチエージェントによるネットワーク機器状態監視システム

A Monitoring System with Multi Agents for Network Connected Nodes

浦島 智, 安宅 彰隆, 畑田 稔

Akira URASHIMA, Shoryu ATAKA, and Minoru HATADA

富山県立大学¹Toyama Prefectural University²

{a-urasim, ataka, hatada}@pu-toyama.ac.jp

ネットワークに接続されたPCやプリンタ等の多数の機器を管理する上で、それらの障害等による可用性の低下を少なくするため、また機器の利用状況を調査するために、管理コストの面からネットワークを利用した監視システムが必要となっている。このような監視システムはその目的から、ネットワーク障害などに対して頑強である必要がある。そこで、エージェントを用いることにより、監視機能が複数に分散し、ある程度自立して動作する監視システムを試作した。これにより、監視システムの一部に障害が起きた場合にも他の部分に障害が波及せず、監視を続けられるシステムとなった。
キーワード：システム管理、エージェント、分散システム、SNMP

In order to manage a lot of network connected PCs and printers, a monitoring system is necessary for decreasing unavailability from software and hardware troubles and for surveying the utilizing conditions. This monitoring system should be robust in the case of network troubles and hardware failure, etc. We have designed and developed an agent-based monitoring system with distributed and self-supporting monitoring functions. In this system, whole monitoring system is not affected and keep on operating from partial system breakdown.

Keywords : system management, agent, distributed system, SNMP

1 はじめに

今日の情報技術の進歩によって、PC、WS、プリンタ等の多数の機器が直接ネットワークに接続されるようになった。大学の計算機センターにおいても、ネットワークに接続された多数の機器を運用しているが、ユーザに対する利便性のなどのため、かなりの数の機器が幾つかの離れた場所(部屋)に分散して配置されている。

これらの機器を運用する上で、SE等の保守要員の巡回によるメンテナンスや、ユーザからの報告による障害検知も重要であるが、それだけでは十分ではない。コスト削減の要求の中、

少ない保守要員では、他の手段を用いずには十分可用性を維持できるような頻度で巡回することは難しい。ユーザからの報告に関しては、パフォーマンス低下の様な軽微な障害の場合や、近隣の同一機器を使える様な代替手段がある場合に、報告が行わないことも多数ある。

また、障害時の対応とは別に、現在のボトルネックの検出や将来におけるシステム更新の設計のため、平常時におけるシステムリソースやユーザの振る舞いに関する情報収集も不可欠である。

このようなネットワーク接続された機器を管理・運用する上で、ネットワークを利用した管理ツールは非常に有用である¹⁾。適切な管理ツールを選択することによって、例えば、数十台あるPCの使用状況やプリンタの紙詰り等

¹〒939-0398 富山県射水郡小杉町黒河 5180

²〒939-0398 5180, Kurokawa, Kosugi-mati, Imizugun, Toyama, JAPAN

の障害の情報収集, それらを利用した週・月単位での利用状況や, ダウンタイムの統計など, 様々な機能が提供される。

現在, ネットワークシステム管理用ツールとしては, “Openview” や “MRTG” のような多数の商品やフリーウェアなどが出ており, 広く使われている。

しかし, これらのツールの多くは, 集中的な管理を単一のマシンで行うものであり, ツールを運用している管理用の機器に障害が発生した場合, すべての管理ができなくなってしまう。

また, 地理的に分散された機器配置に対しては, 管理する側の機器と管理される側の機器間のネットワークも長くなりがちである。機器を繋ぐネットワークが多段になると, その分だけネットワーク障害の生じる可能性は高くなる。集中的な管理方法では, 途中経路のネットワーク障害時にも情報収集は不可能となる。

この他に, 現在のネットワークシステム管理用ツールは, 管理者以外に対する情報提供が十分とは言えないという問題もある。

多くのツールは, 管理者にのみ情報を提供するように作られている。こういったツールでは, 管理される PC やプリンタを直接使用しているユーザに収集した情報が提供されない。また, ユーザに情報が提供される場合も, 提供された情報を理解するためにはある程度の専門的な知識を必要とする場合が多く, その情報をどう解釈すべきか, それによってどう行動すべきかなどといった情報が共に提示されることはほとんどない。

そこで, 以上に挙げたような問題点, 特に障害時の問題点を踏まえて, 本システムはネットワーク機器に対する監視システムを分散して実装する。自立して監視を行う機能を複数に分散することによって, 単一の監視用の機器の故障が, システム全体へ影響を及ぼすことを防ぐ。また, その監視機能を PC やプリンタ等の機器の近くに配置することによって, ネットワーク障害による情報収集の中断の可能性を少なくすることができる。さらに, 分散した機能に冗長性を持たせることによって, 故障時に自動的に他の監視用機器が情報収集を引き継ぐようにする。同時に, ネットワーク機器から得られた情報をユーザに対して, 分かりやすく提供する機能を実装する。

2 マルチエージェントによる監視システム

2.1 監視システムとしての要求

本システムはユーザへの情報提供と共に, 管理者の障害検知・利用統計・パフォーマンス評価のために用いるものである。

本システムの対象は, 大学の計算機センターにあるような, ある程度離れた複数の場所に配置された数百台程度の PC や WS, ネットワークプリンタ等の管理である。この場合, 授業等で多人数が一斉に使う状況を考えると, できるだけ障害のある状態で放置されないことが望まれるが, 代替の機械が付近にあるため即時の障害対応は必要ない。

本システムが, 障害検知等の監視作業を行うことによって, 管理者(機器の設定の変更や整備等を仕事とする SE 等)の人数は少数でよくなる。日に1~2回程度の巡回と, 障害報告の対応ができる程度の人的資源があればよい。障害対応のような作業は管理者が行うものとする。

本システムによる PC や WS 等のネットワーク機器の管理プロトコルとしては, SNMP²⁾(Simple Network Management Protocol) を主に用いることにする。SNMP とは, UDP(User Datagram Protocol) を利用するアプリケーションプロトコルで, 単純なポーリングなどの手法によって, 機器の管理や制御を行うものである。数多くのネットワーク機器が SNMP の機能を実装しており, SNMP で扱うことのできる機器情報も, 非常に多くの種類が定義されている。しかし, 各機器がそれに関わる定義された機器情報をすべて提供しているわけではなく, メーカーや製品によって提供される情報は異なっている。また, 実装されていたとしても, 間違った情報を返す機器が見受けられる。そのため, SNMP を用いる機器を管理する場合は, それぞれの製品毎のカスタマイズが可能にする。

SNMP 以外で機器情報を収集する手段として, ICMP の Echo Request を送る方法を採用している。これにより, SNMP に対応できない機器に関しても ON/OFF 状態を把握できる。また, WS 等に関しては rsh や ssh などの利用も情報収集方法の一つとして採用している。こ

れにより、実行中のプロセスリストといった内部情報を取得できる。

現在もネットワーク機器は長足の進歩を遂げつつある。このような状況にフレキシブルに対応するため、複数の種類のネットワーク機器に対応するための機能をプログラムの形で実装する。あらかじめ定められた種類の機器情報収集機能とそれに対する反応を、単純なルールによって設定する方法に対して、このプログラムによる方法は、状況に応じたダイナミックな情報収集パターンの変更が可能であること、機能の追加が容易であることなどの利点がある。

2.2 機能分散手法

以下に、本システムの特徴である機能分散の手法について述べることにする。

本システムは図1に示すように、システムの機能を、全体を把握する「管理コントローラ」の部分、個々の機器状態の監視を行う「管理エージェント」の部分、それらを繋ぐ「エージェントベース」の部分に分割している。

管理コントローラは、管理者によって使われ、監視システム全体の管理を行う。エージェントベースは、ユーザ側の機器にネットワーク上での近い場所に置かれ、ユーザへの情報提供の窓口の機能と、管理エージェントの動作環境を提供する。管理エージェントは、管理コントローラで産み出されエージェントベースに送られるオブジェクトで、管理される機器に対応し、情報収集と機器状況の判断、ユーザへの情報提供を行う。

ここで言うエージェントとは、管理されるPCやWS等の一台一台に対応し、それらの機器情報を収集して、管理者向け・ユーザ向けに情報を変換するためのプログラム及びデータを含んだオブジェクトのことであり、SNMPの用語におけるエージェントとは異なるものである。

本システムでは、システムのほとんどの部分をJava³⁾で記述することにした。Javaを用いることによって、プログラムの形で実装された各ネットワーク機器に対応する管理方法を、オブジェクトとして簡単に転送することができるからである。

管理機能の一部を分離して移動させ、機器の

近くから情報の収集等を行う類似の研究として、ネットワークワームを利用する研究⁴⁾がある。この研究のネットワークワームは複数の機器を順次調査するのに対し、本システムのエージェントは機器と1対1で対応し、平行して動作する。このため、複数の機器を長期間、継続的に監視するには、本システムが有利であると言える。

以下に、管理コントローラ、エージェントベース、管理エージェントそれぞれについてより詳しく述べることにする。

2.2.1 管理コントローラ

管理コントローラは、管理者が監視システム全体を制御し、情報を収集するための窓口である。

その主な機能の1つは、管理エージェントに関するものである。これには、管理エージェントの生成、エージェントベースへの配布、エージェントベース内の管理エージェントの削除が挙げられる。また、もう一つの重要な機能として、ログに関するものがある。これはエージェントベースからのログの収集と管理コントローラ内での整理と保存である。

管理コントローラは、管理されるネットワーク機器と直接のやり取りは行わない。そのため、管理コントローラの存在する機械に障害が起きた場合や、管理コントローラとエージェントベース間の通信に障害が起きた場合にも、機器の監視は続けることができる。また、同時になければ、別々の場所にいる複数の管理者が同一の監視システムを調整することが可能となっている。

また、監視システムそのものの管理に関する機能を集約することで、他のエージェントベース等に必要とされる機能を少なくし、全体としてシンプルなシステムを可能にしている。

管理コントローラの画面を、図2に示す。管理コントローラはすべてJavaで記述されているため、Javaの実装された機械であれば機種を問わず動かすことができる。図に示すのはLinux上で動作しているものである。

図2の左側には、管理コントローラで生成された管理エージェントの名前が表示されており、同図右側には2つの内部ウィンドウの形で接続中のエージェントベースが表示されてい

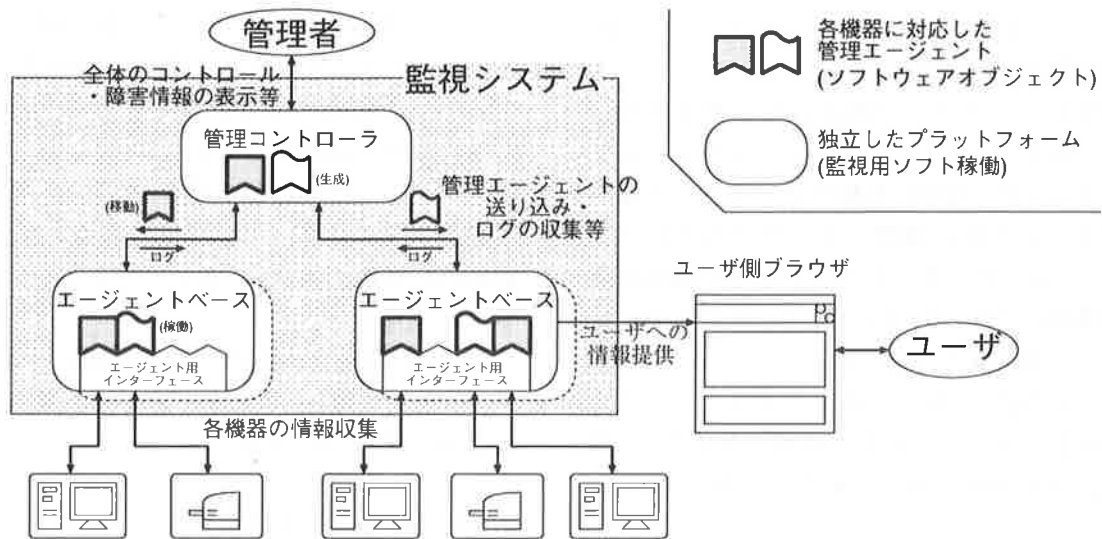


図 1: 監視システムの全体図

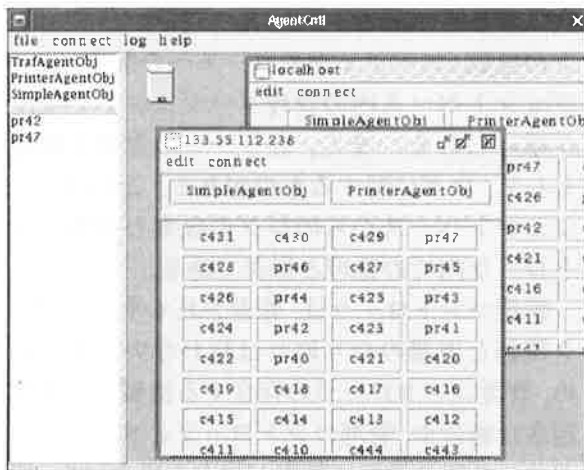


図 2: 管理コントローラの画面

る。エージェントベースのウィンドウの内部には、管理エージェントの名前が多数表示されている。管理エージェントの配布や削除は、画面内の Drag & Drop で簡単に行えるようになっている。

ログは、図 3 の様な形で行単位で区切られ、エージェントベースから送られる。最初の長い数字が 1970 年 1 月 1 日 00:00:00 GMT からのミリ秒数で、ログの出力された時刻を表す。次の数字はエージェントベース内のログのシリアル番号であり、これにより同時刻のログを区別する。3 番目の文字列はログを出力した管



図 3: 収集されたログの例

理エージェントのタイプであり、4 番目の文字列は管理エージェントの個別の名前である。5 番目の 1 桁の数字はログレベルであり、そのログの内容がどの程度の緊急性を持つかを表す。最後の文字列が、ログの内容である。

図 3 のログからは、例えば c406 という名前の機器 (この場合 PC) が、2000 年 10 月 12 日 16 時 20 分 27 秒に立ち上げられ、その 10 分後に電源を切られたことや、ほぼ同時期に pr47 というプリンタが紙切れで使えなくなったことなどが分かる。

ただし、管理エージェントのプログラム作成や、ログの整理と可視化の部分については、まだ管理コントローラの外部での作業を必要とする。

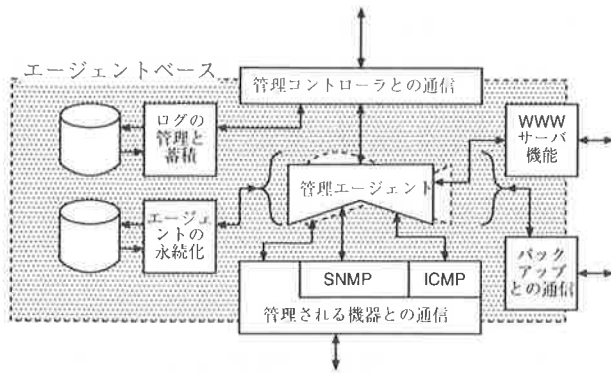


図4: エージェントベースの機能ブロック図

2.3 エージェントベース

エージェントベースは、管理エージェントの動作環境を提供するものである。ユーザ側の機器のネットワーク的な近傍に置かれ、ログの蓄積や外部との通信を行う。

これらの機能を図4にブロック図として示す。

エージェントベースは、管理コントローラによって操作されるが、管理コントローラとの通信に障害のある場合にも自立して動き、管理エージェントの指示にしたがって、機器からの情報収集を続けることができる。

管理エージェントからのログや管理エージェントそのものは、エージェントベースの稼働する機械のストレージにも納められ、ハンガアップや不意の電源断などの場合であっても、次の起動時に自動的に元の監視状況を再現できる。

また、バックアップとなるエージェントベースをあらかじめ別に設定しておけば、定期的にバックアップとのエージェントベース間の通信を行い、稼働中のエージェントベースが停止した場合に、休眠していたバックアップのエージェントベースが監視を引き継ぐことができる。

管理エージェントには、機器の状況をユーザに分かり易く説明するための機能を持たせるが、この説明の情報を受け取り、HTTPサーバとしてユーザに提供するのも、エージェントベースの機能である。

エージェントベースの大半はJavaで書かれているが、Linux上で特権を要する通信ポートを使用するために、一部をC言語で動く別プログラムとしている。

エージェントベースは比較的シンプルなプログラムとなっている。これは、管理者からやや手の届きにくい所に置かれるものであるため、より信頼性のあるプログラムであることが望まれるからである。

2.4 管理エージェント

管理エージェントは、各機器に対する監視を司るプログラムである。機器からの情報収集とその整理、管理者やユーザへ提供するための情報の変換をその主な機能とする。

管理エージェントは、管理コントローラで生成・設定され、エージェントベースに送られ、そこで稼働する。

エージェントベース中の管理エージェントは、SNMPやICMP等を用いた機器情報収集を行い、それらの情報から機器状態の推定を行う。推定された機器状態やその変化の中で、必要な情報をログとして、エージェントベースに渡す。また、エージェントベースから要求があった場合、機器状態をユーザ向きに分かりやすく説明する情報を、アイコンやラベルテキストの形でエージェントベースに渡すのも、管理エージェントの仕事である。

管理者へ提供されるべき情報と、ユーザに対して提供される情報には、内容や表現に大きな違いがあるものと考えられる。そのような違いを複数のレベルとして捉え、それぞれを独立させてシステム管理に用いる研究⁵⁾もあるが、本システムでは、提供する情報の取捨選択は、管理エージェントのプログラム作成者の判断に委せている。

プリンタの機器情報を取得するエージェントの場合、

1. ICMP echo Request を送る。もし返事がなければ、電源が入っていないかネットワークに異常がある。もし返事があれば、2へ行く。
2. SNMP で正常か異常等の状態を調べる。返事がなければ、起動中(長時間なら異常)で、1へ行く。返事があり、異常がある場合は、3へ。返事があり、異常がない場合は、4へ。
3. SNMP で紙切れやトナー切れ等の異常状

態を調べ、それを現在の状態とする。また、状況をログとして出力する。変化があれば、2へ。

4. 正常であると判断する。(前回は異常なら正常に復帰したことをログとして出力する) SNMPでネットワークインターフェースの累計入力データ量を調べ、ある程度以上の増加なら使用中であり、そうでなければ空いている。

といった情報収集・ログの生成を行う。また、要求があった場合、収集した情報から正常なら「♪」のアイコンを提示し、異常なら「!」のアイコンと、状況の説明や対処策を提示する。

多種類の機器に対応する場合には、その分の管理エージェントの種類が必要である。また、同一機種が多数ある場合には、同一種類の管理エージェントが機器数分だけ動くことになる。

管理エージェントもすべて Java で書かれており、エージェントベースが動いていれば、機種を問わず動作させることが可能である。

3 実際の機器管理への適用

本システムは試作段階であり、今のところ日常の運用の一部として使われる段階には至っていない。しかし、システムとして実際の機器管理に利用できる段階にはあり、富山県立大学計算機センターの管理下にある部屋(学生に解放されたパソコン等の端末室)のPC、ネットワークプリンタに対して、本システムの試験的な適用を行った。

この時の本システムの機器構成は、以下の通りである。

- ・ 管理コントローラ用 PC 1台
(CPU: Pentium-III 450MHz, OS: WindowsNT)
- ・ エージェントベース用 PC 1台
(CPU: Pentium-II 233MHz, OS: Linux)
- ・ バックアップエージェントベース用 PC 1台
(CPU: Pentium-III 600MHz, OS: Linux)

本システムで、約50台の機器を30秒に1度の頻度で監視した場合、もっとも負荷の高いエージェントベース用PCでも、ロードアベレージは最大0.1程度であった。このことから、本シ

ステムで2~300台程度の機器が十分に監視できるものと考えられる。

また、管理対象機器とした機器は、以下の通りである。

- ・ 自由利用/実習用 PC 45台
(CPU: Pentium-II 266MHz, OS: Windows95)
- ・ プリンタ 8台
(ネットワークレーザプリンタ)

これら全ての機器は100Base-TのスイッチングHUBを介して接続されている。

これら監視の対象とした機器に関して、PCではハングアップ以外の障害が1年間に11件、プリンタでは紙・トナー切れ以外のトラブルが1年間に2件報告されている。

この試験運用時に、WWWブラウザを通してユーザ側に示された情報の例を、図5に示す。画面の上のフレームには、実際の部屋と同じ配置で、各ネットワーク機器に対応する小さい矩形のアイコンを配置している。目の形のアイコンは使用中のPCを、「Zzz」と書かれたアイコンは電源の切られているPCを、「♪」のアイコンは正常に動作中のネットワークプリンタをそれぞれ表している。下のフレームには、アイコンをクリックした時に機器の状態の説明や異常の場合にはその対処法などが表示される。

このシステム使用中に、管理コントローラを止める、もしくは管理コントローラの稼働しているPCのネットワークケーブルを外す等の、管理コントローラに関する障害を模擬した試験を行ってみたが、いずれの場合にもエージェントベースはそのまま稼働を続けた。この後も、管理コントローラを再起動し、エージェントベースに対して再接続することにより、停止前と変わらない作業を行うことができた。

また、バックアップとなるエージェントベースを別に設定している場合には、稼働中のエージェントベースが停止すると、エージェントベース間の通信が途切れることによって、約1分後に休眠していたバックアップのエージェントベースが、機器に対する監視をはじめるとを確認した。

これらの試験結果は、監視システムを構成する一部の機能に障害が生じた場合でも、監視システムの機能全体には影響が及ばず、障害のな



図 5: ユーザに提供される情報

い部分での監視が継続されるということを示している。

さらに、通常時には、管理のためにエージェントベースと管理される機器との間でやり取りされる通信量 (50 台で往復合わせて 1kB/s 程度) に比べて、エージェントベースから管理コントローラへログとして送られる通信量は、2500 分の 1 程度にまで少なくなることも確認された。

4 今後の課題

本システムを試験的に運用してみた結果、以下のような問題点が明らかになった。

まず、管理エージェントの作成には Java の知識が必要となっていることである。本システムには、Java で直接記述しないで管理エージェントの作成ができるという選択肢も用意した方がよいと思われる。このために、例えばロードアベレージがある値以上になったら報告するといった、単純な管理エージェントに関して、ルールによる記述を Java プログラムに変換するソフトウェアの開発が考えられる。

次に、現在はログに関する整理機能がシステ

ム内に存在しないということが挙げられる。本システムとは別のツールを用いることにより、ログを処理し統計情報等を抽出することは可能であるが、リアルタイムのグラフを書くなどの管理機能の充実のために、これをシステムに組み込むことが必要であると考ええる。

さらに、実際の管理作業に組み込むことにより、障害時の対応を含めた運用実績を蓄積していくことも必要であると考えている。

5 おわりに

今回、システム管理の一部として必要となる機器の監視について、機能を分散して配置することによって、機器やネットワークの障害に対して、システム全体の障害に結び付きにくい監視システムを提案し、試作した。

また、このシステムでは、機器監視の制御にエージェントの仕組みを用いることによって、機能の分散を全体としては単純な構造のまま実装し、複雑な監視手順にも対応できる機構となっている。

そして、この監視システムを実際の機器監視に試験的に適用し、実用性を持つことを確認した。さらに、運用中に、機器やネットワークの障害を模擬した幾つかの試験を行い、システム全体の障害が起らないことを確認した。

本システムは試作段階であり、グラフを描く機能のようなユーティリティとしての機能を拡充する必要性はあるが、基本としての監視機能については、十分実用的なシステムであると考えている。

参考文献

- (1) 西野 正行, 尾崎 貴司, 佐藤 敏朗, 渡辺 浩之: “分散システムの集中管理と自律分散制御実現へのアプローチ”, 分散システム運用技術シンポジウム '98, pp. 117-120 (1998-2).
- (2) Philip Miller 著, 刈田幸雄訳: “マスタリング TCP/IP 応用編”, オーム社 (1998).
- (3) Sun Microsystems, Inc., “The Source for Java Technology”, <http://java.sun.com/>

- (4) 小野木 渡, 清水 亮博, 大野 浩之, “ネットワークワームを利用したネットワーク管理手法”, コンピュータソフトウェア, 日本ソフトウェア科学会, 岩波書店, Vol.16, No.3, pp. 29-46 (1999).
- (5) 近藤 祐志, 勅使河原 可海: “ユーザレベル・アプリケーションレベルに着目したネットワーク・システム管理の実現”, マルチメディア, 分散, 協調とモバイル (DICOMO '99) シンポジウム, pp. 483-488 (1999-6).

著者略歴



浦島 智 1970 年生, 1993 年京都大学工学部卒業, 1999 年同大学院工学研究科博士課程修了, 同年 7 月富山県立大学工学部助手, 計算機センター兼務, 博士(工学)

安宅 彰隆 1952 年生, 1979 年東京大学理学部卒業, 1984 年同大学院理学研究科博士課程修了, 同年 4 月富山県立技術短期大学講師, 富山県情報教育センター副主任兼務, 1989 年 10 月同大学助教授, 1990 年 4 月富山県立大学工学部助教授, 理学博士

畑田 稔 1942 年生, 1967 年姫路工業大学電気工学科卒, 1972 年京都大学工学研究科博士課程単位取得退学, 同年 4 月 (株) 日立製作所入社, 1998 年 11 月富山県立大学工学部教授, 2000 年 4 月同大学計算機センター所長, 工学博士