

プラント監視制御ミドルウェアの開発

—マンマシン管理機能の検証手法—

6N-6

中川 晃一 石岡 卓也 平井 健治 竹垣 盛一
三菱電機(株)産業システム研究所

1.はじめに

近年、計算機の高性能化、低価格化が進むにつれて、制御システムにおいてもS/Wの果たす役割が大きくなってきている。マンマシンS/WはこうしたS/Wの1つで、高機能のGUIを持った大規模なプログラムである。このマンマシンS/Wも以上のような背景からプラント制御・監視などの高信頼性を要求される分野にも使われるようになってきているが、マンマシンS/Wは、オペレータ操作を扱うことや高機能を実現するための機能分散化により信頼性を保証することが一般のS/Wに比べ難しい。我々はこうしたマンマシンS/Wにおける信頼性保証のための機構について研究を行っている。本報告では、マンマシンS/W検証ツールとして開発したシステムモニタについて報告する。この中では、まずマンマシンS/Wの信頼性保証の難しさについて述べ、次にシステムモニタを我々が開発中のマンマシンミドルウェア[1]（以下ミドルウェアと記述する）について適用した例を用いて、システムモニタを使ったS/Wシステムの構成と、システムモニタの機能について述べる。最後に検証例としてミドルウェアの連続試験について述べる。

2.マンマシンS/Wの信頼性

GUIを用いたマンマシンS/Wを検証するためには、キー入力やポインタ操作などのイベントシーケンスに対する的確なイベント処理とレスポンス性を確認しなければならない。しかしイベントシーケンスを入力する場合にはGUI画面の表示とイベントシーケンスの同期、GUIの処理とイベント発生タイミングなどを正しく与える必要があり、人間による確認操作が非常に困難になる場合がある。

またS/Wの規模が大きくなると、マルチプロセスで構成される場合が多くなる。こうしたS/Wの発生する障害を検証する場合、障害の発生したプロセスとそのプロセスへの入出力を知るために、いくつか

Development of Middleware for Plant Monitoring and Control Systems:The verifying method of man-machine systems by KOICHI NAKAGAWA,TAKUYA ISHIOKA,KENJI HIRAI, MORIKAZU TAKEGAKI,(Industrial Control System Laboratory,Mitsubishi Electric Corporation).

のスタブやドライバを使って、イベントデータを作成して試験を繰り返し行いながら検証作業を行わなければならない。こうした作業にかかるコストは検査対象にあるS/Wが大規模、高機能であればあるほど高くなるといった問題もある。

3 構成

図3-1にシステムモニタを使ったミドルウェアの構成を示す。ミドルウェアはXウィンドウシステムを使用しているため、オペレータの操作はすべてXサーバと呼ばれるプロセスを通して伝えられる。またプロセス間通信にはSystem Vのメッセージキューを使っている。

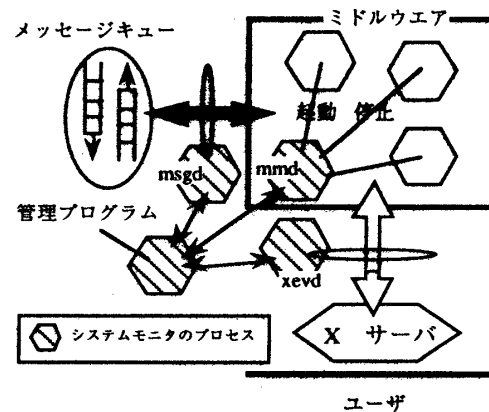


図3-1 システムモニタの構成

同図のようにシステムモニタは3つのデーモンと、1つの管理プログラムの4つのプロセスで構成される。デーモンにはメッセージキューとのI/Fを監視するメッセージ収集デーモン(msgd)や、XサーバとのI/Fを監視するXイベント収集デーモン(xevd)があり、それぞれプロセス間通信の監視と、オペレータによる操作の記録を行う。もう1つのデーモンはマンマシン管理デーモン(mmd)と呼ばれミドルウェアを構成するプロセスの起動状態を監視する。これらのデーモンは管理プログラムから送信される簡単なコマンドで制御できる。また管理プログラムは3つのデーモンと同じネットワーク上にあり、デーモンから送信されたデータを外部ファイルへ保存したり、デーモ

ンへ送信するデータの読み込みを行うといった比較的時間のかかる処理を行う、そのためシステムモニタの処理によってミドルウェアの実行が妨げられることは極めて少ない。

4.機能

以下にシステムモニタの機能について述べる。

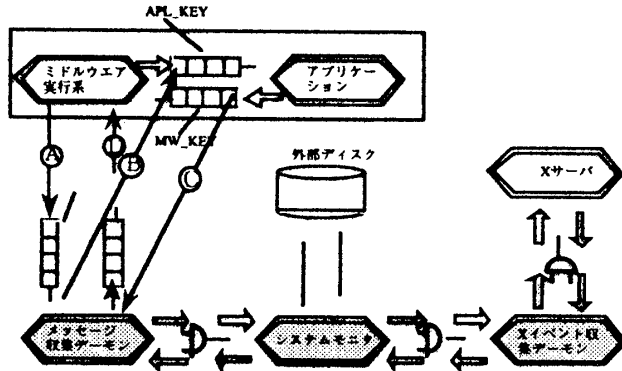


図4-1 システムモニタの状遷移図

(1) イベント受信

システムモニタはXイベント収集デーモンを通して、Xサーバが発生する操作イベントを受信する。システムモニタは内部のリングバッファ（ログデータバッファ）を使って連続してイベントの受信を行うことができる。

(2) メッセージ受信

システムモニタは図4-1のようにメッセージ収集デーモンを通じて「ミドルウェア」-「アプリケーション」間で発生したメッセージを受信する。ミドルウェアからアプリケーションへ送信されるメッセージはアプリケーションのキュー（APL_KEY）に送られず、一度システムモニタの用意したキューに送信（A）され、その後アプリケーションのキューに転送（B）される。一方アプリケーションからミドルウェアへ送信されるメッセージはミドルウェアのキュー（MW_KEY）へ送信（C）されるが、システムモニタがそれを横取りし、ミドルウェアのキューへ転送（D）する。

(3) 再生

システムモニタはログファイルの内容を元にミドルウェアの動作を再生することができる。再生を行うにはミドルウェアを初期状態に戻さなければならない。

(4) ログデータの保存

システムモニタはログデータバッファの内容を外部ディスクにファイルとして保存することができる。

保存したファイルはシステムモニタによって随時参照することができる。

(5) プロセスの異常監視

プロセスの起動状態を定期的に監視したり、それに応じてプロセスの起動、終了を行う。またプロセスが発行するエラーメッセージを収集する。

5.連続運転試験

プラント監視・制御作業では長時間の稼働を保証しなければならない。そのためにS/Wの扱う資源の管理を十分に行うことが必要である。こうした資源の中で最も重要なのはメモリであり、S/W起動後に必要とされるメモリの大きさは確実に見積ることができる。そのためには僅かなメモリの割付不良も無視することはできない。これは長時間の使用によって大きなメモリを消費してしまうこともあるからである。こうした割付不良を発生する箇所を検出するには、同じ操作をプログラムに繰返し与えながら、メモリの割付状態を監視するといった方法が使われる。こうした障害は場合によっては何十時間単位で現象が現れる場合もあるため、人間が操作していくことは大変困難になる。そこでシステムモニタを使うことにより、自動的に同じパターンを何度もプログラムに与えることができ、また障害が発生した地点までの操作記録をとることもできる。

6.おわりに

本報告ではシステムモニタの機能と使用例について述べた。システムモニタは莫大な試験パターンを人間が操作することなく、試験を実施することができる。また連続試験においては長期的にS/Wの挙動も調査することもできる。そのため今まで見つからなかった障害も検出することもできる。

今後は、記録する情報にプロセス情報（起動、停止、サイズ）などを加えて、障害時の状況再現を詳細化し、障害の原因究明に有効なツールとなるよう改良を加えていく予定である。また、同時に記録する情報から効率的に原因を究明が行えるような手法を検討していく。

参考文献

- [1]平井他,「プラント監視制御ミドルウェアの開発-マンマシン管理機能の概要と構成-」,情処全大、平成7年秋