計算機自動運転エキスパートシステムの提案

7G−4

上野　仁　森本成重　村岡正則　吉岡正老郎　木下俊之
(株)日立製作所　システム開発研究所

1．はじめに

大型計算機システムは金融オンラインを始めとした各種オンラインシステムに多く用いられており、その障害は社会活動、企業活動に大きな影響を及ぼす。

大型計算機のシステム障害のうち、１０％前後は計算機オペレータの適切な処置により防止可能か、または短時間での回復が可能と考えられる。しかし、障害状況に応じた適切な処置を行える熟練オペレータの確保は難しく、これを補う自動運転支援システムの開発が望まれている。[2)3)]

本発表ではシステム障害時の自動対処能力を持つ自動運転システムの必要性とその構成方式を示す。

2．システムダウン障害の要因分析

約２５０サイトの大型～小型計算機によるオンラインシステムについて、１９８５年から１９８８年の障害要因を調査し分析した結果、次のような割合でシステム障害が発生していることが分かった。ここでシステム障害とは、システム停止となる障害を言う。

(1)ハードウェア障害　２５％
(2)ソフトウェア障害　３５％
(3)運用上の問題による障害　３０％
(4)不明　１０％

ハードウェア障害は、主として中央処理装置の障害である。ソフトウェア障害の中には基本ソフトウェア障害の他に、ユーザプログラム不良による障害も含む。運用上の問題による障害は、ディスク使用容量の見積り誤りなどのように難しい問題から、休日の保守担当者の構成変更忘れやオペレータのキー入力ミスなどの単純な問題まで、多種の要因を含む。また、「不明」はこの調査では詳細不明の障害を示す。

ハードウェア障害とソフトウェア障害に対しては、部品の高信頼化やソフトウェア品質向上による障害発生率低減の努力がなされており、運用上の問題に関しては自動運用ツールを用いたオペレータ介入作業の削減などの対応がなされている。

しかし、運用上の問題に起因する障害の３分の１（障害全体の１０％）は、オペレータ介入が必要となる状況でのオペレーション誤りにより障害に陥いるケースや、障害発生時の対応に手間どるケースであり、熟練度の高いオペレータが適切に介入すれば障害が回避できた可能性がある。

その他、一時的なソフトウェア障害、ハードウェア障害でも、システム再起動による業務処理の再開が可能であるが、障害時の対処に不慣れなため再起動に時間を要し、長時間のシステム停止となった例もある。

このようなオペレーション上の問題を改善するために、各サイトではオペレーションマニュアル作成やオペレータ教育の強化等の対策が実施されている。しかしこれらの対策の効果は人に依存することが多く、恒久的な対策とはなりにくいという課題がある。[1)]

そこで、対策の確実化を支援するため、システム障害時の対処を自動化する障害対処自動化システムを提案する。

3．障害対処自動化システムの構成

障害対処自動化システムに必要な要件を次に示す。

(1)対象システムからの独立性･･･対象システムのオペレーティングシステム（ＯＳ）やハードウェアの障害発生時も、障害対処自動化システムは動作すること。
(2)障害監視と対処のリアルタイム性･･･人間オペレータ以上のリアルタイム性で対象システムの障害を検出し対処すること。
(3)障害対処手順の柔軟な記述性･･･対象システム運転ノウハウの記述とその妥当性確認が容易であること。

第１の条件より、障害対処自動化システムは対象システムから論理的、物理的にできるだけ隔離して接続された装置上で動作する必要がある。対象システムとの共有部分が多いほど、対象システムの障害の影響を受ける危険性が高いためである。

また、第２の条件により障害対処自動化システムは、対象システムで発生した障害等の特殊なイベントに対して数秒から数十秒以内の応答が要求される。これを満たすには対象システムの状況をオンラインで監視し操作する必要があり[4)]、対象システムで発生するイベントを人手により入力するオフライン方式ではこの要求に対応できない。

第３の条件は、サイト固有の障害検出手順と障害回復手順を記述するために必要となる。例えば、障害回復手順には、単純に再ＩＰＬを行なう場合と、ファイル回復等の回復処理が必要な場合などがある。このような条件はサイトの方針や運用方法の変更により異なるので、これに対応する手順記述の変更と確認が容易である必要がある。

仮想計算機(VM)を含む複雑な構成を持つ計算機システム

Proposal of Automatic Operation Expertsystem for Mainframe Computers
Hitoshi UENO, Narishige MORIMOTO, Masanori MURAOKA, Masaichiro YOSHIOKA, Toshiyuki KINOSHITA
HITACHI,Ltd.

では、複数のオペレータコンソールを持つ。オペレータコンソール操作のうち、各種サブシステム操作と正常時のＯＳ操作は、ＯＳ上のサブシステムの一つとして動作する従来の自動運転システムで自動化されている。

しかし、ＯＳ障害やハードウェア障害発生時には、ＯＳコンソールやハードウェアコンソールを操作する必要がある。そこで、障害発生時のオペレータ介入自動化のため、対象システムの外部に障害対処自動化システムを設ける。対象システムと障害対処自動化システムとの間は、各構成要素（ＯＳ、ＶＭＣＰ、ハードウェア）を制御するコンソールインタフェース、または同等の制御が可能なインタフェースを用いて接続する。（図１）

障害発生時の対処方法は対象システムの性質によって異なり、ハードウェアコンソールを用いて再起動を試みる単純なレベルから、障害発生時の状況に応じて起動するＯＳを切替えて再起動するという複雑なレベルまである。したがって自動対処手順は、サイト毎、対象システム毎に書き換えられる必要がある。

自動対処手順の記述方法には、エキスパートシステム構築ツールを利用して図２に示すようなルール記述言語を用いる方法と、従来の言語処理系を用いて図３に示すような手続き型言語を用いる方法などが考えられる。そこで、次の観点から特徴を検討した。

(1)障害対処ノウハウの記述性

手続き型言語では、手順記述の制御構造や、プリミティブなレベルのデータ構造を意識する必要がある。ルール記述言語では、これらを強く意識する必要が無い。

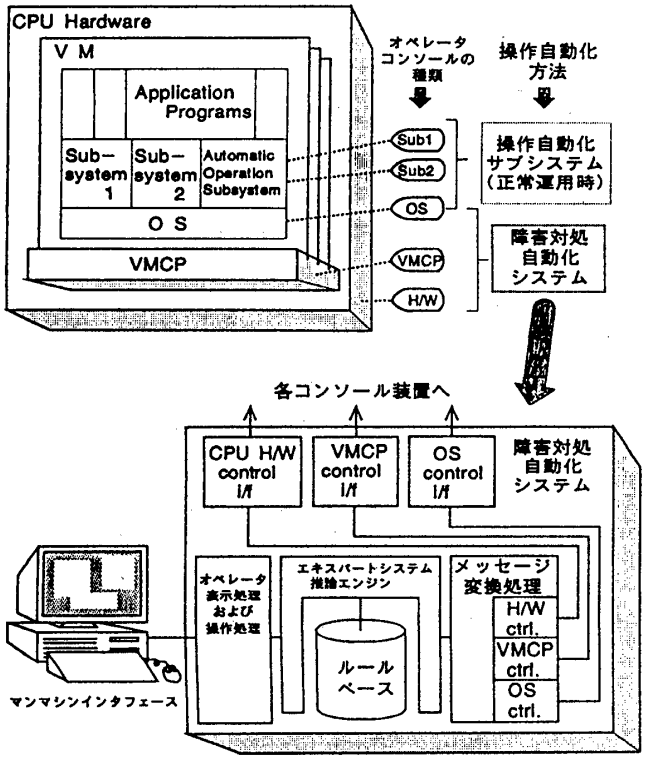

図１　障害対処自動化システムの自動化範囲と内部構成

(2)処理速度

ルール記述言語では、エキスパートシステムの推論機構がルールの探索処理を行なうため、手続き型言語と比較してオーバヘッドが大きい。しかし、ワークステーションクラスの計算機を障害対処自動化システムの処理装置として用いる場合、ルールの書き方にも大きく依存するが、数百個程度のルールを備えたルールベースを用いても、数秒以内の応答が可能であると見込まれる。

このように、ノウハウ記述性の観点からはルール記述言語を用いることが望ましく、処理速度もそれほど問題にはならない。

以上の観点から、エキスパートシステムを利用した障害対処自動化システムは、障害発生時の自動対処に有効な方式であると考える。

４．おわりに

システム障害の原因を調査し、障害時のオペレータコンソール操作自動化が、ダウン時間短縮の一つの鍵となることが分かった。また、操作自動化の手順記述には、エキスパートシステム上のルール記述が有利である。

５．参考文献

1) J. H. Griesmer et al. : 'YES/MVS: A Continuous Real Time Expert System', AAAI-84, pp. 130-136 (1984)

2) 守友, 大西他: ﾄﾗﾌﾞﾙ対処へのｴｷｽﾊﾟｰﾄｼｽﾃﾑの適用, 情報処理学会第37回全国大会, pp. 289-290 (1988)

3) 海老野, 信沢他: ｺﾝﾋﾟｭｰﾀ運用・利用ｴｷｽﾊﾟｰﾄｼｽﾃﾑ, NEC技報, Vol. 41, No. 12, pp. 85-91 (1988)

4) 廣澤, 栗原他: ﾊﾟｿｺﾝ制御による電子計算機ｼｽﾃﾑの自動運転ｼｽﾃﾑ(IV), 情報処理学会第41回全国大会(4), pp. 69-70 (1990)

記　述	ノウハウ
(Recover_from_down_1 if (System_down) then (send Recov_sys Re_ipl()))	①システム障害が発生したら、再ＩＰＬを実行する。
(Recover_from_down_0 if (System_down) (?J_rem @Job_name = MAST01A or @Job_name = MAST02A) (?J_rem @Job_name -> ?J_rn) then (send Recov_sys Operator_call(?J_rem,?J_rn)))	②システム障害時に、MAST01A または MAST02A というジョブが実行中なら、（ファイル回復のため）オペレータを呼ぶ。

図２　if-thenルールによる記述例

記　述	ノウハウ

void Sys_down()
{
char j_id[6];
if(srch_jq("MAST01A",j_id)!=NULL)
op_call(j_id,"MAST01A");
else
{
if(srch_jq("MAST02A",j_id)!=NULL)
op_call(j_id,"MAST02A");
else
re_ipl();
}
} /* end of Sys_down */
	①システム障害が発生したら、実行中のジョブの中に、MAST01A または MAST02A というジョブがあるかを調べ、あるなら、オペレータを呼び出す。両方とも無いなら、再ＩＰＬを実行する。

図３　手続き型言語による記述例