

時間制約と一貫性制約を満たす
実時間データ獲得システム
RTDS

島川博光 井戸譲治 竹垣盛一

三菱電機(株) 産業システム研究所
〒661 尼崎市塚口本町 8-1-1

システム構築の低価格化が叫ばれる今日、Unix, Windows などの非実時間 OS 上の監視システムやデータベースシステムで、実時間システムからのデータを利用したいという要求が産業や交通管制の分野で高まっている。Real-Time Data Server(RTDS)は観測対象から実時間でデータを獲得し、非実時間 OS 上のアプリケーションに提供するデータ獲得システム(DAS)である。さらに、RTDSは獲得データから特定状態を検知する能動機能を備えている。本論文では、実時間データベースで論ぜられている時間制約と時制データベースで論ぜられている時間に関する一貫性制約の両方を満足する DAS の計算モデルを提案する。特に状態検知機能を実現するために我々は2種類の ECA カップリングモードを使った。我々のモデルに基づき RTDS を構築することの利点を考察する。

RTDS: A Data Acquisition System
Satisfying Timing and Consistency Constraints

Hiromitsu Shimakawa George Ido Morikazu Takegaki

Industrial Electronics & Systems Lab., Mitsubishi Electric Corp.
8-1-1, Tsukaguchi Hon-machi, Amagasaki, Hyogo, 661 Japan

Real-Time Data Server(RTDS) is a data acquisition system(DAS) which works as a gateway to supply data from real-time systems to non-real-time systems. In this paper, we propose a DAS computational model to satisfy both timing constraints and temporal data consistency constraints, in addition to logical consistency constraints which are important in general database systems. We show the advantage of the computational model from view points of satisfaction of the two constraints in DAS.

1 まえがき

プラント制御、交通管制システム、医療など監視・制御システムでは観測対象から実時間でデータを獲得するデータ獲得システム (DAS) は不可欠である。システム構築の低価格化が叫ばれる今日、Unix, Windows などの非実時間 OS 上の監視システムやデータベースシステムといった既存ソフトウェア資産と実時間システムをオンラインで結合し、実時間で獲得されたデータを既存の資産で利用したいという要求が高まっている。Real-Time Data Server (RTDS) は観測対象から実時間でデータを獲得し、これを時制データとして非実時間 OS 上のアプリケーションに提供する DAS である。また、観測対象の特定状態を検知できればアプリケーションは効率的に動作できる。RTDS は一般の DAS の機能に加え獲得データ項目値が指定された条件を満たせばアプリケーションに伝える能動機能を備えている。

本論文では DAS が、実時間データベースで論ぜられている時間制約と時制データベースで論ぜられている時間に関する一貫性制約を満たさなければならないことに着目し、これら両方を満足する計算モデルを提案する。特に状態検知機能を実現するために我々は 2 種類の ECA カップリングモードを使った。このモデルに基づき、我々は RTDS を開発している。我々のモデルに基づき RTDS を構築することの利点を考察する。

2 要求と制約

2.1 RTDS に要求される機能

Unix, Windows 上の監視システムやデータベースシステムなど、既存のアプリケーション・ソフトウェア資産を利用するために、RTDS は観測対象から実時間でデータを獲得し、これを時制データとして非実時間 OS 上のアプリケーションに提供する DAS である。DAS に要求される機能を列挙する。

獲得 周期/非周期で発生するデータを時制データとして獲得し続ける。

保存 時制データを長期間保存する。獲得が継続されるなら保持量には限界があるので、最新データの獲得時には最古のデータを上書きする。

最新時制データの周期転送 監視のため周期的に最新時制データを転送する。

過去時制データの検索 任意の過去の期間の時制データを検索し転送する。

また、観測対象の特定状態を検知できればアプリケーションは効率的に動作できる。RTDS は一般の DAS の機能に加え獲得データ項目値が指定された条件を満たせばこれをアプリケーションに伝える能動機能を備えている。

RTDS がアプリケーションに提供する時制データは、観測対象の状態変化の履歴を示すデータ項目値の並びで

ある。我々は以下に示すような形式を持つ時制データを時系列と呼ぶことにする。時系列は、その特性を表すヘッダと、場面の並びからできている。場面とは、データ項目値の組であるスナップショット [4] と、観測された時刻を表す時刻印の組である。

2.2 2 種類の制約

時制データをもれなく獲得しなければならないので、DAS では実時間データベースと時制データベースで扱われる 2 種類の制約を考慮する必要がある。

2.2.1 時間制約

実時間データベースの研究においては、トランザクションを実行する価値と終了する時期との関係から 3 つのトランザクションの締切が定義されている [2][3]。

厳密な締切 間に合わなければ災害が起こるような締切。締切に間に合わない場合のトランザクション実行の価値は大きな負の値である。

確固とした締切 間に合わなければ、その時点で実行の価値が 0 となる締切。

緩やかな締切 締切以降、時間経過とともに実行の価値が減少しいつかは 0 となる。

これらに対し締切がないトランザクションでは、時間が経過してもトランザクション実行の価値は変化しない。

2.2.2 一貫性制約

時制データベースでは、現実世界において事実が真となった正当時間と事実がデータベース上に格納されたトランザクション時間という 2 つの時間が論ぜらる [4]。本来、正当時間とトランザクション時間は直交している。しかし DAS においては、現実世界の事実は発生後直ちに格納されるべきであるから、これらはその差が一定値よりも小さく、縮退している [3]。

格納するデータに対しては、外部からセンサを通して獲得された基礎データと他のデータから導出された導出データの区別 [3] と、連続データと離散データの区別 [10] がある。離散データの値は時間の経過しても無効になることはないが、連続データは時間経過とともに絶えず変化する外部実体を表現している。したがって、連続データを格納する時制データベースでは通常のデータベースにおける意味的な一貫性だけでなく、時間に関する一貫性も保持されなければならない。このため、連続データが正当であると考えられる期間を導入し、連続データの時間に関する一貫性が定義されている [6]。

絶対データ一貫性 連続データ項目の最新値の正当時間 (もしくはトランザクション時間) と現在時刻との差は、このデータ項目の有効期間よりも常に短くなければならない。

相対データ一貫性 あるデータ項目の新しい値を導出するのに必要なデータ項目の値は、相対の有効期間内に観察されたものでなければならない。導出されるデータ項目と導出に必要なデータ項目の集合は、相対的一貫集合を形成する。ここで、個々の相対的一貫集合に対して、1つの相対の有効期間がある。

CT を現在時間とし、連続データ d に対し d_{vt} を d の正当時間、 $avi(d)$ を d の絶対有効期限とする。さらに d が相対的一貫集合 R の要素であるとき、 R の相対的有效期間を $rvi(R)$ で表現する。 d が正しい状態である必要十分条件は、

1. d の値が意味的に一貫性をもつ。
2. d が時間的に一貫性をもつ。すなわち、
絶対データ一貫性: $(CT - d_{vt}) \leq avi(d)$
相対データ一貫性: $\forall d' \in R, |d_{vt} - d'_{vt}| \leq rvi(R)$
を満たす

ことである [2]。

2.3 DAS における問題点

DAS では獲得の遅延は獲得データの価値を 0 としてしまうので、獲得には確固とした締切がある。絶対データ一貫性を保持するために個々の連続データ項目の獲得周期は絶対有効期限を考慮して決定されなければならない。相対データ一貫性を保てるように相対的一貫集合の各要素の獲得周期は決定されるべきである。

一方、データ提供は緩やかな締切をもつと考え Unix や Windows などの非実時間 OS 上に構築された多くの既存アプリケーション・ソフトウェアで実時間で獲得されたデータを利用しようとする研究がある [1],[8]。一般に DAS では獲得データをアプリケーション・ソフトウェアで利用するために、これらのほとんどが持つソケット・インタフェースを使って履歴データを転送する。アプリケーション・ソフトウェアが動作する OS にも、また、データ転送に使用するネットワーク・プロトコル上にも実時間性がないので、DAS におけるデータ提供ではこれに対処する特別な措置が用意されていなければならない。従来の実時間データベースの多くの研究 [5][12][13] では、一様な特性を持つトランザクションが締切内に終了する確立を高めることに主眼がおかれており、DAS における実時間システムと非実時間システムの結合上の整合性が考慮されていない。Kim らは異なる実時間性を持つタスクごとにスケジューリング手法を用意する方法を提案 [10] しているが、その実現は簡単ではない。

また、状態検知を実現する手続きには複雑な条件分岐がある。したがって条件評価を含んだ手続きの実行最悪時間を事前に見積もることは困難である。ところが、実行最悪時間を事前に見積もることは実時間システムを

構築する上で必要不可欠である [6]。この矛盾が状態検知を DAS に導入する上で最も大きな問題となる。

3 計算モデル

DAS はデータを指定された期間内で獲得し、かつ、獲得された基礎データと導出データの一貫性を維持しなければならないので、データ獲得システムの研究は実時間データベースの研究に多くの共通の問題点を持つ。しかし、時間制約と一貫性制約の両方を満たす一般化されたシステムの実現は簡単ではない [5] ので、それぞれの問題で要求される機能をよく吟味して、個々のシステムでの制約を満たす実用的な方法を見つけなければならない。DAS における時間制約と一貫性制約を満たすために我々は RTDS のための計算モデルを提案する。

3.1 構成要素

計算モデルは領域と機構に基づく。領域はデータを保持している。機構は1つの手続きを1サイクルとして繰り返す。各サイクルでは、ある領域上のデータを読みだし、これを処理して、その結果に応じ別の領域上を書くか他の機構を起動する。読み出しと書き込みはすべて一度に実行され、これらが交互に実行されることはない。機構内の手続きに条件分岐は少ないので、機構が手続きを実行するために必要な最悪実行時間が見積もりやすい。

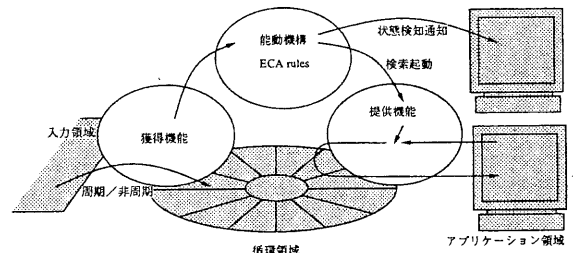


図1. 計算モデル

RTDS の計算モデルは 6 種類の要素からなる。

入力領域 そこからデータを獲得すべき領域

循環領域 獲得された時制データを十分長い期間保持する領域

アプリケーション領域 RTDS が時制データを送る領域。
アプリケーションは提供されたデータをここから取り出す。

獲得機構 は入力領域からデータを獲得し、時刻印を付加して、循環領域に格納する。

提供機構 は循環領域から時制データを取り出し時系列としてアプリケーション領域におく。

能動機構はイベント、条件、ふるまいのECAルールで実現される。あらゆる獲得がイベントであり、獲得されたデータに対する条件が満たされたとき、指定されたふるまいが実行される。

各機構内の手続きの実行は、1サイクルにおいて、データの読みだし、書き込み以外に他の機構から影響を受けない。循環領域には、複数の機構がアクセスするので、排他制御が必要である。循環領域には時制データが獲得時刻順に並んでいるので、循環領域全体を排他制御せずに循環領域上で書き潰しが起こっている部分を示した小さな管理情報を排他制御することで一貫性をもつデータを読み書きできる。

各機構は、独立したサイクルを繰り返し実行する。RTDSでは、実現を容易にするため、各機構の1サイクルの実行が指定された期間内に終了することをDeadline Monotonic Scheduling Algorithm[7]をつかって保証する。

3.2 循環領域

循環領域は獲得周期ごとに1つずつ用意される。循環領域上では最新データの獲得時には最古のデータが上書きされる。循環領域上では場面が時刻印の順に並んでいるので、任意の期間の場面は2分探索により高速検索が可能である。

すべての循環領域をメモリで構築することはあまりに高価である。RTDSでは、最近のデータへのアクセスは早くなければならないが過去のデータへのアクセスは遅くても良いと考え、最近のデータはメモリ上のリングバッファに、過去のデータはディスク上の連続ファイルに保持している。連続ファイルの内容は循環的に更新される。

3.3 獲得機構

入力領域には、センサが観測したデータ項目値を書き込む。獲得機構は入力領域上のデータ項目値、即ち基礎データ、もしくはそれらから得られた導出データに獲得時刻を示す時刻印を付与して場面とする。獲得機構は循環領域の最も古い場面上に新しい場面上を上書きする。

獲得機構は、周期的な起動のためのタイマからのイベント受信機能と非周期的な起動のための外部からのイベント受信機能をもつ。周期的に起動される獲得は連続データ項目値を得るのに使われる対して、非周期的に起動される獲得は観測対象からのデータ発生通知に応じてデータ項目値を得るのに使われる。

基礎データに対する絶対データ一貫性を保持するために獲得は確固とした締切をもつトランザクションとして必ず成功しなければならない。逆に獲得は必ず成功するので、再実行や取消の手段を用意する必要はない。

データの保存も獲得機構で行なう。リングバッファが一杯となった時、次の獲得による上書き起こる前に場

面がディスクへ保存されないとデータが喪失される。保存もまた、確固とした締切をもつトランザクションとして実行される。

3.4 提供機構

提供機構は循環領域上の場面を検索し必要な値を導出する。提供機構は時系列を構成しそれをアプリケーション領域に送る。提供には、最新の時系列の周期的提供と、要求に応じた指定の期間とデータ項目値の時系列の提供がある。

最新の時系列の周期的提供に対しては、アプリケーションとそれへのデータ転送路であるネットワークに実時間性が保証されていないので、DASが最新データを周期内に転送しても、アプリケーションがそれを遅れて受けとったり、取りこぼすことがある。また、ネットワーク上でデータが喪失するおそれもある。しかし、我々は周期転送実行の価値はアプリケーションにより決定されることに注意すべきである。例えば、監視画面にデータが遅れて表示されることの価値は0でないこともある。また、喪失にたいしてもアプリケーションが前後のデータより補間できる場合もある。DASにはアプリケーションでの価値を知る術がない。そこでRTDSでは、最新時系列を周期的に転送することを確固とした締切をもつトランザクションとして実行することとした。時系列のとりこぼし・喪失はアプリケーションが、受信した時系列の時刻印を検査して発見するものとし、もし発見されれば、アプリケーションで前後のデータより補間するか、RTDSに過去の時系列として検索依頼を出す。

要求に応じた時系列提供では、実時間性よりもむしろ平均の応答時間が短いことが要求される。このトランザクションは締切をもたないものとして実行される。時系列のとりこぼし・喪失はアプリケーションが発見し、必要なら再送要求を出すものとする。

3.5 能動機構

状態検知のために、能動データベースにおけるECAモデル[11]を利用する。即ち、あらゆる獲得をイベントとして、獲得された基礎データもしくは導出データに対する条件が検査される。もし条件が成立したら、それに関連づけられたふるまいが実行される。

状態変化はいつでも起こる可能性があるので、能動機構が獲得のたびに条件の検査を行なう。しかし、条件分岐をもつ手続きの最悪実行時間の見積もりは実際にはほとんど起こらない条件の評価のために悲観的となることが多い。このことはDASとして最も大切なデータ獲得性能を低下させることになる。我々は評価にかかる時間を正確に見積もることができる条件とそれ以外の条件を区分し、前者を獲得機構で後者を提供機構で別々のトランザクションとして実行することでこの問題を解決することとした。獲得時に評価される条件を獲得時条件と呼ぶことにする。

能動機構のECAモデルにおけるふるまいは、アプリケーションへの通知と提供機構の起動である。アプリケーションに即座に通知されるべき状態は獲得時条件だけで判定できると仮定し、獲得時条件が成立すれば獲得機構は獲得されたデータ項目からなる時系列を転送して状態検知をアプリケーションへ通知する。アプリケーションへの通知は遅延モードのECカップリングと即時モードのCAカップリングで実現される。獲得時条件で検知できない状態の検出には複雑な条件判定が必要であると仮定し、獲得時条件成立時にさらなる処理を提供機構に依頼する。提供機構の起動は遅延モードのECカップリングと分離モードのCAカップリングで実現される。現行のRTDSでは獲得時条件は新たに獲得されたデータ項目値に対する線形の(不)等式で表せるものとしている。

4 構成

4.1 RTDSの構成

RTDSは図2に示すように共有メモリ上にリングバッファを置いた多プロセスで実現される。獲得プロセスはリングバッファに場面を書き込み、この場面は提供プロセスにより読み出される。通信プロセスはアプリケーションからの要求を受けとりそれへの回答をアプリケーションに返す。さまざまなCPUをもつアプリケーションからRTDSが利用できるように、通信プロセスとアプリケーション内のクライアント・スタブは転送データをネットワーク・オーダとホスト・オーダ間で変換する。総領プロセスはアプリケーションからの要求を解釈して適切な提供プロセス内の子スレッドにその実行を依頼する。

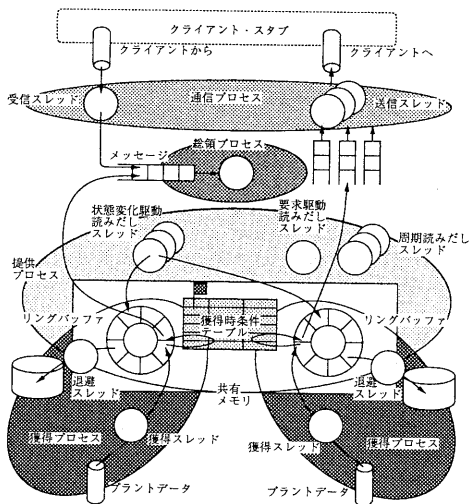


図2. RTDSの構成

4.2 獲得時条件

DASとしての基本機能であるデータ獲得性能の低下を防ぐためRTDSの状態検知機能では、ECAモデルにおける条件を、評価時間を見積もれる獲得時条件とその他の条件の2種類に分割して考えている。獲得時条件は新たに獲得されたデータ項目値に対する線形の(不)等式で表現される。

獲得時条件テーブル内の各エントリは条件1つに対応する。各エントリは条件の識別子、獲得時条件を表す式、条件成立時のふるまいを示したメッセージ・フィールドからなる。1つの獲得スレッドが割り当てられた獲得時条件を評価する時間を見積もれなければならない。獲得時条件テーブル上ではデータ項目値*i*の識別値は名前前でなく、獲得された場面の先頭アドレス*b*からの偏差*d_i*で表されており、獲得時条件は

$$a_{min} \leq \sum_{i=1}^n a_i(* (b + d_i)) \leq a_{max}$$

で表現される。1つの獲得スレッドが割り当てられた獲得時条件を評価するのに必要な計算量はデータ項目数を*n*、獲得時条件数を*m*として*O(m · n)*である。

電気系 2	名前	式						メッセージ		
		項数	第1項	第2項	...	下限	上限			
2	1	fix	var	fix	var	>	-5.7	*	id	argument
	2	3.0	x	-2.4	y					
3	1	fix	var	fix	var	<=	10.2	*		
	2	1.0	x							
3	1	fix	var	fix	var	=	0.0	*		
	2	1.0	p	1.0	q					

図3. 獲得時条件テーブル

4.3 提供におけるデータ喪失

RTDSとアプリケーションを繋ぐネットワークが混雑すればデータは喪失するが、RTDSではTCP/IPプロトコルを使った場合転送にかかる時間を見積もれないことから、UDP/IPプロトコルを使ってデータの転送を行なっている。また、RTDSでは、アプリケーションの時系列取り出しが遅れ新しい時系列が古い時系列を上書きしてしまうことによりデータの喪失が起こる。

RTDSでは、データ喪失時には再送要求の発行や喪失データの補間などの処置をアプリケーションがとる。したがって、アプリケーションにとって重要なのはデータ喪失を発見できることである。周期転送では転送される場面の時刻印で喪失を知ることができる。状態検知と過去の時系列検索の場合は、非周期的に転送される時系列のとりこぼし・喪失を発見できるようにRTDSは時系列に通し番号をつけて転送する。

5 考察

本論文で提案した計算モデルに基づきRTDSを構築することの利点を考察する。

5.1 時間制約と一貫性制約の一元管理

DAS では時間制約と一貫性制約が同時に満たされていなければならない。しかし、静的な実行周期の保証ができれば、2種類の制約は時間制約のみを満足することで一元的に充足できる [10]。RTDS では、静的な保証を与えるため各機構内のスレッドを Deadline Monotonic Scheduling Algorithm [7] に基づきスケジュールした。

獲得における時刻印の値は格納した時点、すなわち、トランザクション時間とする。獲得において指定される締切 i_a は、正当時間 t_0 とトランザクション時間 t_t との間で許される誤差である。よって $|t_0 - t_t| \leq i_a$ である。一般に i_a は、アプリケーションにとって実用上無視できるほど十分小さい値になるように DAS は設計される。

また、RTDS がアプリケーションに時系列を提供する際にいかなる場が基準時刻を b とする時系列内に含まれるかは場面の時刻印に対するレートと許容誤差で指定される。レート r は場面間の時刻印の隔たりを表す。許容誤差 e は1つの場面内にあるデータ項目値のトランザクション時間の最大の隔たりをしめす。 n を整数として、提供される時系列内の n 番目の場面 $\sigma(n)$ の時刻印を $\tau(n)$ とする。データ項目値 v のトランザクション時間 t_t が $b + n \cdot r \leq t_t < b + n \cdot r + e$ を満たすとき、 v は $\sigma(n)$ に属し、 $\tau(n) = b + n \cdot r$ と見なして良いものとする。

RTDS におけるデータモデリングにおいて時制データの—貫性制約は次のように満たされる。

5.2 絶対データ—貫性

最新データの周期的転送もしくは状態検知の即座の通知において絶対データ—貫性が満たされなければならない。循環領域の最新場面の正当時間とトランザクション時間の差は一定値 i_a よりも短い。最新データの周期的転送では、提供機構が循環領域の最新場面を読み出してアプリケーション領域に送る。これにかかる時間は一定値 i_c 以下である。状態検知の即座の通知においては、能動機構が状態を検知して通知データをアプリケーション領域に送る。これにかかる時間は一定値 i_n 以下である。絶対データ—貫性を得るため転送されるデータの絶対有効期限を i_{avi} として $i_a + i_c \leq i_{avi}$, $i_a + i_n \leq i_{avi}$ をみたくように i_a, i_c, i_n が設定される。

5.3 相対データ—貫性

基礎データ項目値よりデータを導出する場合に相対データ—貫性を考慮する必要がある。

循環領域上の複数の場面上のデータ項目値から時系列の n 番目の場面内のデータ項目値を導出する場合がある。許容誤差を e として $b + n \cdot r \leq t_t < b + n \cdot r + e$ を満たす時刻印、すなわちトランザクション時間 t_t を持つ循環領域上の場面に導出に使用することができる。獲得

締切の最大値を I_a とすると、相対的—貫集合 R 内のデータ項目値 d, d' について、それぞれの正当時間 d_{vt}, d'_{vt} の差が最大となるのは一方が発生と同時に格納され、他方が締切直前に格納された場合であるから、

$$\forall d' \in R, |d_{vt} - d'_{vt}| \leq I_a + e$$

が成り立つ。この条件をもとに獲得締切が設定される。

6 あとがき

RTDS は実時間で獲得したデータを非実時間システムに提供し、状態を検知する DAS である。本論文では、時間制約と—貫性制約を満足する計算モデルを提案した。特に状態検知機能のために獲得時条件とそれ以外の条件を区別し2種類の ECA モデルに基づく機構を我々は提案した。我々はこのモデルの利点を考察した。

謝辞

本研究に対し御指導頂く京都大学・上林教授に感謝いたします。

参考文献

- [1] 島川, 水沼, 竹垣, プラント履歴データの実時間獲得・提供システム, 信学論, Vol. J78-D-I, No.8, August 1995
- [2] K.Ramanatham, Real-Time Database, *International Journal of Distributed and Parallel Database*, Vol.1, No.2, March 1993
- [3] Gultekin Özsoyoglu and Richard T.Snodgrass, Temporal and Real-Time Database: A Survey, *IEEE Trans. on Knowledge and Data Engineering*, Vol.7, No.4, 1995
- [4] A.U.Tansel, T.Cliford, S.Gadia, S.Jajodia, A.Segev, and R.Snodgrass, Temporal Databases, Theory, Design, and Implementation, Benjamin/Cummings Publishing, 1993
- [5] Özgür Ulusoy, Current Research on Real-Time Databases, *SIGMOD RECORD*, Vol.21, No.4, December, 1992
- [6] Marc H.Graham, Issues in Real-Time Data Management, *J. of Real-Time Systems*, Vol.4, pp.185-202, 1992
- [7] C.L.Liu and J.W.Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *J. ACM*, Vol.20, No.1, pp.46-61, 1973
- [8] J.Muratore, T.Heindel, T.Murphy, A.Rasmussen, and Z.McFarland, Acquisition at Mission Control, *Comm. ACM*, Vol.33, No.12, 1990.
- [9] R.Rajkumar, Synchronization in Real-Time Systems, A priority Inheritance Approach, Kluwer Academic Publishers, 1991
- [10] Y.Kim, and S.H.Son, Supporting predictability in Real-time Database Systems, *Proc. of 2nd Real-Time Application and Technology Symposium*, 1996.
- [11] D.R.McCarthy and U.Dayal, The Architecture of An Active Database management Systems, *Proc. of ACM SIGMOD Intl. Conf. Management of Data*, 1989.
- [12] D.Hongm T.Johnson, S.Chakravarthy, Real-Time Transaction Scheduling: A Cost Conscious Approach, *Proc. of the 1993 ACM SIGMOD Intl. Conf. Management of Data*, pp.197-206, 1993
- [13] J.R.Haritsa, M.Livny, and M.J.Carey, Earliest Deadline Scheduling for Real-Time Database Systems, *Proc. 12th RTSS 1991*.