

製造設備向けリアルタイム SQL 生成ユニットの開発

渡部 裕二^{†1} 石川 博^{†2} 水野 忠則^{†3}

工場の製造現場では生産の最適化をはかるために品質検査値, 稼働率, 生産性のような様々な設備情報が必要である. 本論文では, 生産設備自身に直接データベースと通信するための機能を付与する SQL 生成ユニットについて述べる. 同ユニットは, 設備イベントの検出機能や, 設備情報と生成時刻から SQL 文を生成する機能を有し, リアルタイムな設備情報の獲得を簡易な構成で可能にしている.

Development of Real Time SQL Generation Module for Industrial Equipment

YUJI WATANABE,^{†1} HIROSHI ISHIKAWA^{†2}
and TADANORI MIZUNO^{†3}

Optimization on the manufacturing floor requires facilities information such as quality inspection value, operating rate and productivity. This paper describes the SQL generating unit that allows the facility directly to communicate with database. This unit generates SQL from time stamp and facility event detection. And it realizes acquiring manufacturing information from facility in real time with simple configuration.

1. はじめに

近年, 情報およびネットワーク技術の普及は製造現場にも及んでいる. オフィスワークで

は情報処理技術の活用によって飛躍的に効率を向上させたが, 実作業中心の製造現場ではこれまで, 作業指示表や検査書・実績報告書などの紙データを電子化する程度にとどまっていた. しかし最近では, 工場全体での製造工程の緻密なスケジュール管理によって短納期化や在庫の圧縮をはかり, あるいは膨大なデータから製造不具合を詳細に分析することで高品質化をすすめるなど, 製造の様々な局面で情報を収集し分析して最適に処理させようとする動きが活発である. しかし, このとき課題となるのが設備や装置からの情報収集である. 人間の多様な行動に比べて簡単そうに見える装置・設備の挙動も, データ化して取り出すには様々な仕掛けを用意する必要があり, そのためコストもかかればトラブルも多い. これまで設備や装置におけるコンピュータの役割は主に機械制御やプロセス制御を中心としてきており, 情報の収集と連携にはあまり良い手段が提供されておらず, 処理のボトルネックとなっていた. 本論文では, 特に設備情報の収集に焦点をあて, 生産設備自身が SQL 文を生成する機能を持ち, 直接データベースと通信させることによって, リアルタイムな設備情報の獲得をシンプルな構成で可能としたリアルタイム SQL 生成ユニットの開発について述べる. なお論文中, 設備と装置という用語が使い分けられるが“設備は装置のやや規模を大きくしたものの”程度の差異であり, 本論文の議論では同義とらえていただきたい.

2. 従来研究と課題

2.1 製造現場におけるデータ収集の目的と構成

製造現場にも様々な情報があり収集の方法も用途によって異なる. μsec で変化する加工プロセスの分析から, 人の操作履歴の集計など様々なものがあるが, 本論文では製造設備の状態遷移や挙動を取りこぼしなく収集し, データベースへ登録できることを目的とする. まずはじめに前提となる生産システムにおけるデータ収集の構成から説明しておく.

2.2 生産システムにおけるデータ収集の構成

図 1 にデータの収集に注目した生産システムのハードウェア構成を示す. 本論文ではこれを“ファクトリオートメーションにおける 3 層モデル”と呼び, 以下に情報システムにおける 3 層クライアントサーバモデルと対比しながら, その動作と課題について述べる.

通常, 製造現場にある様々な装置は, すべて人手作業による生産でないかぎり複雑な機械類を動作させるためコントローラと呼ばれる制御部を有する. この制御部には組み込み型のボードコンピュータや, PLC (Programmable Logic Controller) が使われ, 装置に取り付けられたセンサや入力機器の信号を受け取り, それに基づく判断によって次の動作を決め指令となる信号を出力する. この信号にそってアクチュエータなどを用い機械を動作させ,

^{†1} 三菱電機株式会社名古屋製作所
Nagoya Works, Mitsubishi Electric Corporation

^{†2} 静岡大学情報学部
Faculty of Informatics, Shizuoka University

^{†3} 静岡大学大学院創造科学技術研究部
Graduate School of Science and Technology, Shizuoka University

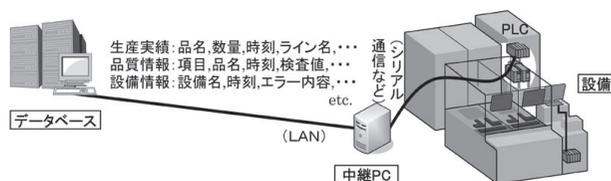


図 1 ファクトリオートメーションにおける 3 層モデルの構成
Fig.1 Configuration of 3 layers model in factory automation.

温度や圧力を制御して装置の動作を司る。

ここでいう装置とは、たとえば部品の組み立てを行うロボット、機械加工をする工作機械、部品の良否を画像を使って判定する検査装置、ベルトコンベアで製品を搬送するシステムなど様々なものがある。一方コントローラが有する信号情報には作業開始ボタンの ON/OFF 状態といった単純な状態量から、ロボットアームの位置情報、検査結果、設備アラームと時刻など多様な情報が含まれる。さらにこれらを組み合わせた情報が装置内で管理されることも多く、結果として生産実績や故障履歴、品質情報、設備履歴を持つことになる。

このようにコントローラは装置情報をすべて把握しているので、装置情報を収集する場合このコントローラから取得すればよい。コントローラには簡易なものも多く、必ずしもすべてが通信機能を有するわけではないが、上記のボードコンピュータや PLC はネットワーク機能を備えており、これらを用いた装置からはネットワーク経由で情報を取得することができる。しかしこれまではそのまま情報システムに接続されるのではなく、装置と情報システムの中間にはほとんどの場合、PC のような中継機能が介在している。

2.3 生産システムにおけるデータ収集の処理方式

従来はどのように装置内部の情報を収集していたのか、処理内容について装置コントローラとして PLC を使った図 2 を用いて述べる。

従来のシステムでは、装置側にはいつどのような情報を発信すればよいか設定する機能を持たなかったため、中継 PC は装置を常時チェックして、データを取得すべき状況となったときにデータを収集するという方式をとっていた。そのため通信方式は中継 PC からポーリングにより周期的に問い合わせる方式となる。接続の形態は様々で、1 対 1 のシリアル通信接続である場合も多い。またデータのフォーマットや対話形式（プロトコル）も独自のものが多く、相手 PLC のメーカーや機種に合わせ個別に開発の必要があった。取得されたデータはたとえばセンサの計測値そのままのバイナリデータであるため、小数点位置の調整や

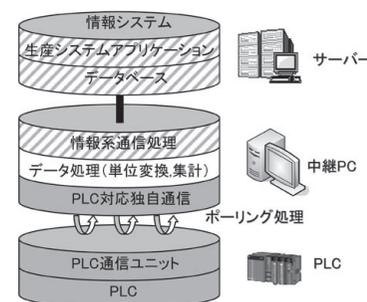


図 2 ファクトリオートメーションにおける 3 層モデルの処理方式
Fig.2 Communication method of factory automation 3 layers model.

単位変換などの演算、あるいは平均値/合計値/最大値などの集計も行われる。そしてその後、サーバ上にあるデータベースに書き込む操作が行われる。この構成は 3 層クライアントサーバにおけるユーザインタフェース層を装置インタフェースに置き換えた形態と見ることができ、装置側通信の差異を中間で吸収した構成となっている点でメリットはある。しかし中継 PC は装置インタフェース層から受け取ったデータをもとに多彩なアプリケーション処理をしているとはいえ、装置側での機能不足であるイベントの監視やプロトコル変換・フォーマット変換を補助している位置づけである。

2.4 従来構成での収集方法の課題

PC を中継機能として用いるこの構成ではいくつかの問題点がある。

(1) リアルタイム性

PC はどちらかというとリッチな画面を優先させ、リアルタイム性については操作者が体感的に不満に感じない程度で妥協している。したがって、PLC との通信に割り込む別の処理もありポーリングの周期も必ずしも一定しない。しかし PLC によって制御される事象データは装置内で長時間保持されるとは限らない。このため PC によるポーリングではその次の周期までに事象が変化してしまうと、データの取得ができず図 3 に示すように取りこぼし¹⁾が発生する。

(2) 通信トラフィックの増大

ポーリングでは有意なデータであるかどうかは問い合わせなければ分からず、取りこぼしを極力減らすためにはできる限り高頻度に問合せをかける必要がある。このため通信路にはつねにデータが流れている状態となり、他用途との共用が困難で、かつ接続相手数やデー

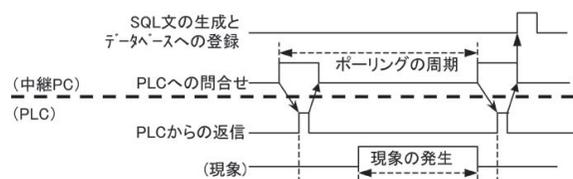


図 3 データの取りこぼし

Fig. 3 Missing data in communication.

タサイズによってレスポンスのばらつきが生じるため、確実なデータ取得に支障をきたすことがある。

(3) データ時刻の精度

取得したデータは、その内容だけでなく生成した時刻情報が重要な場合が多い。厳密にいうと PC 中継によるデータ収集で管理できる時刻は事象が発生した時刻ではなく、PC が通信によってデータを受け取った時刻であり、通常 200 msec 以内程度に収まるがシステムの構成によっては無視できなくなる。

(4) コスト

中継のための PC のコストが必要である。PC といえば低価格なものと思われがちだが、生産設備としての信頼性を確保するためには電源の強化や耐環境性を向上させるための筐体などが必要となり、全体としては高価なものになる。

(5) 信頼性

PC は基本的にホームユースであり、耐久性・耐環境性などはオーバスペックにならないよう適度なレベル設定がされている。一方、製造現場ではデータの収集といえども停止してしまうと甚大な被害となるため、導入機器の信頼性は厳密に要求される。また PC ではややりスクのあるような機能でも利便性や新規性を優先して採用されることが多く、いわゆる“枯れた”安定技術を指向する生産設備とは異なる設計思想による機器である。

(6) 製品ライフサイクル

同様に、PC は一般消費者の新しい需要を喚起するために積極的に新機能を導入し、そのために 3~4 年の短期間で機能向上やモデルの変更がなされる。製造現場では、安定的に製造できるようになったラインは変更することなく 5~10 年以上使い続けられる。したがって、使われる機器の提供も同様の期間保証されなければ生産財としての利用は困難である。たとえば数年後に同一仕様の保用品を調達しようとしても、すでにその製品の製造が打ち切られているといった事態が発生し、生産ラインの長期にわたる運用に支障をきたすことになって

しまう。

3. 関連研究と本論文の考え方

元来、製造設備からのデータ収集には専用の機器を付加し記録紙などにプロットするような仕組みが使われてきた。しかしこれではデータ集計・分析が難しいことから、PC (Personal Computer) の登場以降 PC にデータとして読み取らせ、これを PC 上のアプリケーションソフトウェアで集計・分析させるシステムが使われるようになった。PC の OS である Windows とコントローラとの通信インタフェースは OPC (OLE for Process Control)²⁾ として標準化されており、この仕様に基づいたシステム³⁾ が多く登場している。さらに工程進捗や品質管理などのデータのように広く工場全体の情報と組み合わせることで経営の指標や納期回答などに展開される場合には、大規模・高信頼・検索性のゆえに最終的にデータは RDB に格納する構成が使われる⁴⁾ ようになってきている。これがここまでで説明してきたシステムであり、前述したような課題を有している。

これに対し PLC と通信するのではなく装置と直接通信する機器を付加し、これにポーリングによる取りこぼしを防ぐ仕組みを搭載する試み⁵⁾ もあるが、装置情報との同期精度は粗く、PC でタイムスタンプを付与する仕様とあいまって時刻精度は秒を超える。またログファイルへ蓄積されたデータは検索速度や操作性の面で劣るため、専用ソフトを開発して表示や分析を行うか、再度データベースへ登録するような操作が必要である。

一方、機器に付加した専用ユニットのメモリにいったん実時間でデータを蓄えた後ファイルとして一括転送する方式⁶⁾ も提案されている。しかしユニットにイベント判断機能はなく、単に周期的にデータを取るため、連続する変化量を記録するには適しているが、非定期的なイベントによって生じる状態遷移や変化を記録する場合には無駄なデータが多く、データ量が過大になってしまう。

ポーリングはすでに待ち行列の巡回問題として様々な解析がなされている⁷⁾。そしてデータ消失を防ぐためには時間制限や優先度を設ける手法⁸⁾ が使われるが、そのためにシステムが複雑となりプログラム開発のコストも増大する。本論文では PLC の 1 ユニットにイベント判断機能を持たせ、ポーリング処理そのものをなくし本来無効な通信の発生を抑止しようとする。

またデータ獲得の際には PLC の制御に追従するため時間制約があり、処理が予測可能でなければならない⁹⁾。そこで本論文では、データベース処理とデータ獲得をリングバッファを介して切り離し、リアルタイム性の検証はユニット内でのデータ獲得とリングバッファへ

の書き込みまでを考える。

4. リアルタイム SQL 生成ユニットの実現

4.1 ハードウェアプラットフォーム

ハードウェアプラットフォームとしては生産設備で最も使われることが多い PLC を使用し、SQL 生成機能を PLC の追加ユニット上に実装した¹⁰⁾。これには以下のような理由がある。

- (1) 装置の制御用コンピュータシステムとして製造現場での普及度が高く、実用への展開が容易である。
- (2) センサや入出力機器との信号の授受に使われるフィールドネットワーク機能を持ち、これら装置情報を自由に読み書きできる。
- (3) 装置は PLC の制御周期ごとに変化する信号に基づいて動作しており、データをこの周期に合わせて取得することでデータの取りこぼしがなく十分なリアルタイム性が得られる。

ここで PLC のシステム構成と動作を簡単に述べておく。図 4 に PLC の構成を示す。PLC は一般的にシステムバスを持つベースユニットに、電源や CPU ユニットを容量や能力にそって選択し、また入出力や通信機能を持つユニットを必要に応じて選択・装着することで構成される。入出力ユニットはさらにセンサやアクチュエータに接続され、通信ユニットはネットワークに接続され装置として機能する。

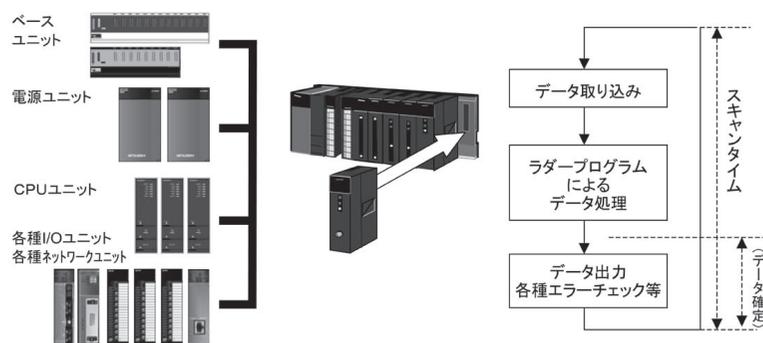


図 4 PLC の構成と基本動作

Fig. 4 Configuration and operation of PLC.

PLC の基本動作として、まず外界の入力信号やデータがシステムバスを經由して CPU ユニットに読み込まれる。続いて、この入力信号やデータに対し“ラダー”と呼ばれるプログラムに基づいた処理や判断が実行され、最後にその結果がバスから各出力ユニットを經由して出力される。この 1 巡の手順をスキャンと呼び、これが繰り返されることによって装置を制御する。このスキャンの時間は主に処理 CPU ユニットの性能と処理すべきプログラムのサイズに依存するが、一般的に 1~10 数 msec 程度となることが多い。PLC でもスキャンタイム以下の短時間で変化する事象を制御することはできないが、主な使用目的は高速応答を必要とする速度制御や振動制御のようなダイナミクスの制御ではなく、装置の動作順序などの論理的な制御である。

このように PLC によって制御される設備からの情報取得では、制御周期すなわち PLC のスキャンタイムに合わせたリアルタイム性を持たせる必要がある。

4.2 課題と対応

本論文で使用したハードウェアは PLC の付加ユニットの 1 つである。リアルタイム OS が搭載されており、その上で動作するアプリケーションを自由に開発可能である。また、システムバスを經由して PLC 本体とのデータのやりとりができ、またスキャンのタイミングを通知することができるため、スキャンのタイミングに一致させて入力データを取得することにより、少なくとも PLC によって制御される事象の取りこぼしはない。

設計上の技術課題を再度整理して対応策を述べる。

- (1) PLC に制御される現象を取りこぼさない。
取りこぼさないためには PLC のスキャンタイムごとにデータを取得する必要がある。また PLC のプログラム処理中は出力データが書き換えられるため、PLC から同期信号を発生させデータが変化しない時間帯（図 4）にのみデータを獲得する。この時間制約を守るため、取得するデータ量には制限を設けてある。
- (2) データを取得する条件の判断機能を持たせる。
SQL 生成ユニットに、スキャンタイムで獲得した条件データの演算（論理演算，四則演算）結果により、収集データを獲得するか否かの判定をするような機能を持たせた。ただし時間制約のためこの演算回数にも制限を設けてある。そして、PLC 本体の制御プログラムを修正することなく、この条件を容易に設定変更可能でもある。
- (3) 時間制約を守りながら同時に多数のデータを獲得しデータベースに登録する。
優先度の異なる 2 つのタスクとリングバッファを設け、優先度の高い周期タスクでは時間制約のあるデータ獲得とリングバッファへの書き込みを行い、優先度の低いタスクではリング

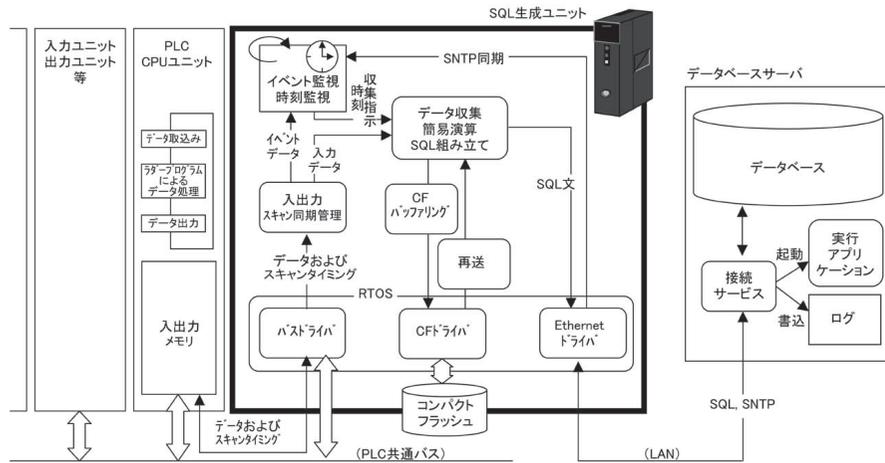


図5 ソフトウェア構成
Fig. 5 Software configuration.

バッファからの読み出し、SQL 分の生成、送信を時間制約なく実行する。このような手法は既知でありいくつかの報告がある¹¹⁾。なおこの場合、条件によってはリングバッファがあふれるおそれは残っている。

(4) 多数のユニットのタイムスタンプ精度と同期性を確保する。

1つのサーバに接続されたユニット間では、SNTP を使用して全ユニットをサーバの時刻に合わせる方式とした。しかし、この方式での時刻精度はサーバに依存し 30 msec ~ 40 msec という報告もあるため、PLC の制御スキャンタイムレベル (1 ~ 10 msec) の時刻精度を目的としたデータ獲得には不十分であり、本論文でも課題を残している。

図5 にソフトウェア構成を示す。

まずスキャンタイミングを監視して、PLC 入出力メモリが固定されている間に PLC 共通バスを経由してデータを受け取る。このうちのイベントに関連するデータを演算・判断して SQL 文の生成が必要かを判断する。定刻・定周期イベントについては内部時計による。これら処理は設定により変動するがデッドラインを守るよう処理量の上限值が決まっている。必要とされたデータはいったんリングバッファに記録され、別の優先度の低いタスクでデータを取り出し SQL 文の生成が行われる。何らかの原因で SQL 文が受け取られなかった場合には CF (Compact Flash) メモリに保存し再送を試みる。そして一定の間隔で



図6 フィールドへの装置内データの割付け
Fig. 6 IO assignment to data field.

SNTP 時刻同期がサーバとの間で実施される。

4.3 SQL 文の生成

開発したユニットの特長は装置から直接 SQL でサーバにあるデータベースに情報を書き込むことにある。このためには装置自身が、いつ、どのデータを、どこに、どのようなフォーマットで格納すればよいか知っていなければならない。ユニット内部には書き込み先のデータベースとテーブルが記憶され、さらにレコードの各フィールドとそこに書き込まれるべき装置内のデータのアドレス指定をテーブルとして定義しておく。ユーザインタフェースは図6 が用意されており、データベースから抽出したデータ構造に対し、PLC のユーザプログラム作成時に生成されたタグテーブルを参照し、プルダウンメニューから選択し割り付けるだけで簡単に設定が可能である。タグテーブルは実アドレスに対する名前の定義を持っており、装置内情報のアドレスは実値を記述する必要はない。この設定に基づいてまず SQL 文のテンプレートが用意され、次に外部から取得したデータがこの構文に埋め込まれてメッセージとして組み立てられる。

SQL テンプレートの例を図7 に示す。ここで “tagA. 生産指示数 A”などは、装置で測定された実際の値が代入されるタグ名を意味しており、最終的には PLC の実アドレスを参照してデータが取得されるので図8 の SQL 文が生成されて発信されることになる。

この設定は1つのユニットについて複数持つことができ、1つのユニットから異なるデータベースへの書き込みや異なる構成のレコードの書き込みも自由に行うことができる。

このように SQL 生成ユニットを用いることで非常に単純な設定だけでデータベースへの書き込みが完了する。

4.4 装置イベント

どのようなデータを取得し、どのようなフォーマットで、どのデータベースに記憶する

```
INSERT INTO NEWTBL_A ( DATETIME_A, SEISANSIJI_A, SEISANZISSEKI_A,
FURYOUHIN_A, FURYOURITU_A, TOUNYUJIKAN_A, TEISIJIKAN_A,
KADOURITU_A ) VALUES
('日時 [YYYY/MM/DD hh:mm:ss]'),'(tagA. 生産指示数 A)',
'(tagA. 生産実績 A)', '(tagA. 不良品数 A)', '(tagA. 不良率 A)',
'(tagA. 電源投入時間 A)', '(tagA. 稼働停止時間 A)', '(tagA. 稼働率 A)');
```

図 7 SQL テンプレート
Fig. 7 SQL template.

```
INSERT INTO NEWTBL_A ( DATETIME_A, SEISANSIJI_A, SEISANZISSEKI_A,
FURYOUHIN_A, FURYOURITU_A, TOUNYUJIKAN_A, TEISIJIKAN_A,
KADOURITU_A ) VALUES
('19800/08/18 14:23:48 ', '20', '19', '1', '5', '600', '132', '78');
```

図 8 生成される SQL 文
Fig. 8 Generated SQL message.

か、という内容は前項のデータベースの定義/入力データのアドレスの割付設定で定義できる。しかし“いつ”データを取得するか、それをどのように定義するか、ということが設備の情報取得では重要であり本ユニットの大きな特徴でもある。

データを取得するタイミングには大きく分けて 2 つの考え方がある。1 つは一定の周期・時刻で現象の変動にかかわらずデータを継続して取得する方法。もう 1 つは、現象を監視し必要な条件を満たした場合にのみデータを取得する方法である。前者はまったく未知の事象を分析する場合に使われるが、データ量が非常に多くなり、しかも有意なデータであるかどうか不明であることが多い。一方、後者はデータを必要とする条件が明確になっている必要があるが、生産状況の監視のように「製品の検査が完了したらデータを取得する」場合や「1 つの工程が完了したら実績報告を上げる」などはこの条件にあてはまり、有意なときのみ通信が発生する。

装置の状態を直接監視することで確実にデータを取得できるため、SQL 生成ユニットでは後者のように外部事象をイベントにする場合にこそ従来方式に比べメリットが大きい。

本ユニットではイベントのための通信データが不要となるだけでなく、イベントの判断と収集データの取得が同時に行われ、イベントが条件を満たした場合のみ通信が行われるた

表 1 データ収集のための装置イベントの種類
Table 1 Equipment event for data acquisition.

	方式	内容
1	定刻	指定時刻にイベントを発生
2	定周期	指定周期でイベントを発生
3	値監視	メモリ内容と指定値を比較して条件に合えば イベント発生
4	ハンドシェーク	ビット入力の ON/OFF を監視しイベントを発生 イベント終了後ビット出力を ON/OFF する
5	ユニット立ち上がり	電源投入直後にイベントを発生

め、取りこぼしのないデータ取得を実施しながら、通信路が混雑するということがない。さらに従来のポーリング問合せ方式の場合、通信速度と取得データの量、および現象発生の頻度や信号の維持時間につねに留意しながら設計を進めなければならないことから、システムの実現は非常に面倒であり不確実である。

なお、本ユニットのように監視周期（スキャンタイム）が短い場合であっても取りこぼしが皆無とはいえない。このようなデータを確実に取得したい場合に、外部事象の遷移を保留させることができるのであれば、信号処理によるハンドシェークをとることが可能である。具体的には、ユニットでのデータ取得完了を待って信号を出し、装置側にこの信号が出力されるまで他の状態に進まない仕組みを持たせる方式である。

定周期イベントも含め、本開発で定義した装置イベントの種類を表 1 に示す。

4.5 バッファリング

信頼性を確保する機能として通信データのバッファリングがある。製造現場にとって、ネットワークやデータベースサーバにトラブルがあったからといって生産に関する重要な情報が消失してもらっては困る。従来の PC を使用したシステムでは、PC 自身の信頼性の低さからデータの消失が発生するだけでなく、ネットワークやデータベースサーバのトラブルに備えた機能が不十分であり、確実なシステムの構築や機能確認には非常に手間がかかっていた。本 SQL 生成ユニットにはデータが確実にデータベースに書き込まれたことが確認できない場合に、自動的に内蔵のコンパクトフラッシュメモリにバックアップする仕組みを持たせてある。しかも、その後の定期的なリトライの結果、データの書き込みが復旧したと判明すれば、それまでバックアップしておいたデータを再発信する。データには生成時の正確なタイムスタンプが付与されているので、リアルタイムな取得ではないもののマクロな分析などにはまったく支障がない。

4.6 タイムスタンプと時刻合わせ

データにとってその発生した時刻は重要な要素であり、前述のバッファリング機能を活かすためにもデータ自身に発生した時刻をペアで記録させることが必要である¹²⁾。しかし従来方式では時刻は PC で管理されており、このためデータに付与されるタイムスタンプは、正確にはデータが生成された時刻ではなく PC が通信によってデータを受け取った時刻となる。通常は 1 ポーリング周期以内の遅れであり微小なものであるが、ポーリング周期は接続した設備台数や取得データ量にも左右されるためバラツキもあり無視できない場合もある。本ユニットは時刻管理機能を有しており、設備内部でイベント検出と同時にデータが生成された時刻を直接タイムスタンプとしてデータに付加して記録することが可能である。したがって仮にデータ通信で遅れが生じ、いったんバッファに保管され後に発信されたとしても、データ生成時刻は正確に把握できる。ただし、複数台の装置が連動している場合や工場全体の設備を管理する場合には注意が必要である。収集したデータのタイムスタンプの基準時刻が合っていないと正確な判断はできないため、システム全体の時刻合わせをしておく必要があるが、本ユニットには SNTP 時刻同期機能¹³⁾ も持たせてあるので、複数のユニットをサーバの持つ時計に正確に同期させることが可能である。

5. 評価と実証

本章ではデータ取得の速度、およびシステム構築のコストと工数について検証結果を報告する。前者はどのような用途にまで使用できるかの判断材料であり、後者は実システム開発時にこの方式を採用した場合のメリットの目安となるものである。

5.1 処理速度の評価

図 9 に示すような構成で処理速度を測定した。実際の装置やセンサ/入出力機器は接続せず、PLC があたかも入力データを受け取ったかのように自分自身で特定のデータを生成し発信する。結果を図 10 に示す。測定時間はイベントを検知してから SQL 文の生成と発信、その後サーバより完了メッセージの返信が PLC で確認されるまでとし、時間は PLC の内部クロックにより計測した。この結果によれば、UPDATE/INSERT 処理では 1 フィールドあたり 1.3~2 msec 程度の時間で処理可能であるが、フィールド数が少ない場合オーバーヘッドが目立ち、単位あたりの処理時間が見かけ上長くなってしまふことが分かる。その結果、情報がデータベースに登録されるまでの処理には最低でも 200 msec 程度必要である。ただし、データの記録そのものは以前の通信処理が完了していなければ一時的にバッファに蓄えながらデータを収集し続けるので、バッファ容量を超えるまでは PLC のスキャン周期での

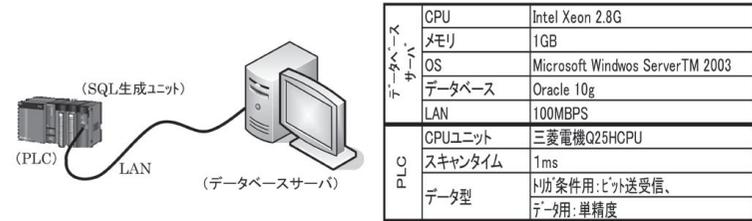


図 9 評価システム構成
Fig. 9 Evaluation system.

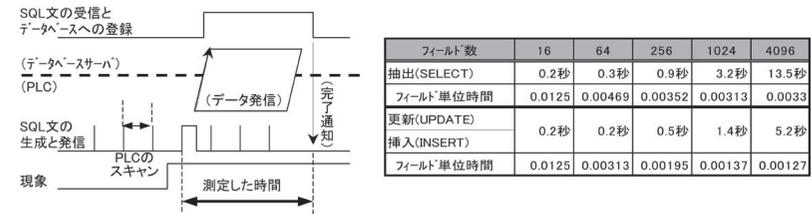


図 10 処理速度の評価
Fig. 10 Evaluation of processing efficiency.

データ収集が可能である。

5.2 システム化コストと工数の評価

実用に向けた評価基準としては、このユニットを使用した場合のシステム全体のコストおよび開発工数の低減も大きな要素である。

従来の中継 PC を使った方式でシステムを構築した例を基準として比較した結果を図 11 に示す。システム構成としては複数の設備を制御ネットワークにより設備群としてまとめ、その中心となる PLC からそれぞれデータを収集し最終的に 1 台のデータベースサーバにデータを集約する例で考える。処理能力差を考慮して中継 PC を SQL 生成ユニット 3 台で置き換えている。なお、ここで設備間の“制御ネットワーク”とは遅れ時間を保証したサイクリック伝送ネットワークで、PLC 間でメモリ(入出力データ)を共有する機能がある。本コスト比較には関与しないが実用途では一般的に使用されるため記述した。

SQL 生成ユニットでシステム化したソフトウェア費用については別システムでの作業時間を参考に、見積り誤差を考慮して 130%と余裕をみた数値で算出した。SQL 生成ユニット

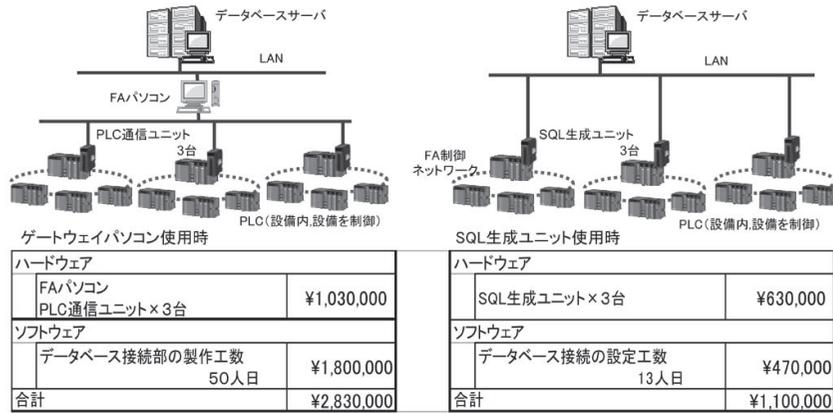


図 11 システムコストと工数の比較
Fig. 11 Cost evaluation.

トはパラメトリックな設定のみで作業が終わるため、対象システムが変わってもそれ程作業内容の変動はなく十分な見積り精度と考えられる。なお、機器コストの中で“FA パソコン”は耐環境性のための防塵筐体など高信頼化した PC を使用。一方で SQL 生成ユニットは元来、PLC の一部として現場での使用に耐えうる信頼性・耐環境性を有するものであり標準仕様品である。

結果、全体としてほぼ 1/3 と大幅な費用削減となることが検証できた。その大部分をソフトウェアの開発工数の差が占めており、これはプログラムレスであることが主要因だが、プログラム開発につきもののバグの発生や担当者が代わった場合の引継ぎの困難性などを大幅に軽減できるだけでなく、収集データをいろいろと変えて解析を実施したいような場合、たとえば現場改善・突発的な不具合の解析などに、コストをかけることなくシステムを柔軟に変更できることも意味しており、メリットが大きい。

5.3 適用実証

SQL 生成ユニットを実際の工場に適用した成果について述べる。この適用対象は三菱電機（株）名古屋製作所内のサーボモータ組立工場であり、多品種小ロットの生産を実施する量産工場である。部品加工の終わったステータとロータ、エンコーダなどを 1 つのモータに組み立て、各種検査を経る最終工程まで、約 200 の設備に対して、83 台（2008 年 8 月現在）の本ユニットを導入して工程情報や検査情報をきめ細かく収集、あるいは生産指示

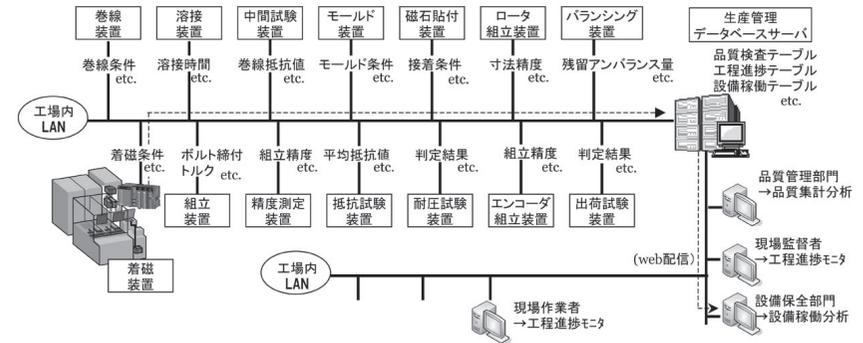


図 12 実施例のシステム構成
Fig. 12 Application system.

などを配信して生産効率を上げている。ユニットからのアクセス頻度は 1 日 7.75 時間で約 100 万回（挿入/更新/抽出の合計）。データベースとしてはテーブル数約 70、フィールド数約 5500、レコード数約 1680 万（容量を超えたものは日付の古いものからバックアップへ移す）となっている。

5.3.1 適用実証におけるデータの流れ

データの流れを中心としたシステムの大まかな構成を図 12 に示す。

図中で“巻線装置”、“溶接装置”などはモータ製造過程で使用される多数の装置名を示しており、各装置は PLC によって制御され、PLC は“巻線条件”“溶接時間”のように装置によって多様な情報を記憶し制御に使用する。収集される情報は分析の目的に沿ってこれらの中から必要なものが選択されるが、図では主に品質管理目的で収集されるデータを例としてあげている。品質以外にも、たとえばある装置で処理を始めたモータフレームのバーコードを PLC が読み取ることで、今どの装置すなわち工程にモータがあるか作業進捗情報が報告され、データベースに蓄積されてゆく。蓄積された情報は、品質管理部門であれば品質の集計分析、現場監督者であれば工程の進捗モニタ、設備保全部門であれば設備の故障情報などの稼働分析、というようにそれぞれのアプリケーションによって検索・集計され、また視覚的に見やすい情報に加工されて LAN 経由で工場の各所に web 配信される。この報告やモニタのリアルタイム性は数秒から数十秒程度のオーダーである。全データはデータベースとして確実に保存されるため、製品が市場に出荷された後であっても、各々のモータのフレームに付与されたバーコードから製品が識別さえできれば、工場では検索によりこの固体管理情

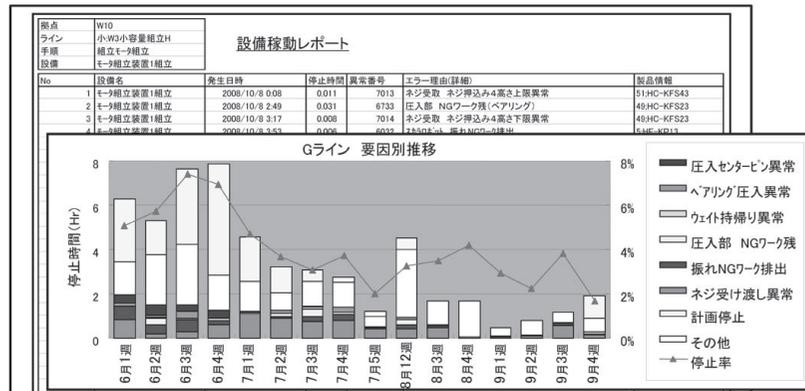


図 13 設備稼働レポートと設備稼働率の向上
Fig. 13 Facility report and improvement of availability.

報に紐付けされた加工条件など生産時の様々な情報を簡単に調べることができトラブルに対応できる。

5.3.2 データ処理による生産性の改善

SQL 生成ユニットを用いて収集されたデータをもとに、この工場ではいくつもの生産性改善の効果をあげている。

図 13 はある設備の名称/発生日時/停止時間/異常番号/エラー理由/製品情報が詳細に記録された稼働(停止)レポートの例である。これらすべての情報は設備が生産を行う場合に PLC が保持しており、ユニットは異常発生信号を監視してイベントが発生した場合にデータベースに登録する。何らかの要因で設備が停止すると生産が滞り工場としては損失を生じる。特に新しい設備を導入した新設ラインでは、当初はフル稼働がなかなか難しく、設備の問題点を改善しながら稼働率を上げてゆくことになるが、この場合大きな停止要因すなわち多回数発生し長時間停止した問題点から対処してゆくことが、単純ではあるが早期の生産立ち上げにつながる。とはいえ 200 を超える設備に対しこれらをきめ細かく記録し集計するには相当の人手コストがかかるため、これまでは作業者の大まかな判断に基づいて処理されていた。事例では全件全設備について稼働レポートを日々作成しながら重要な要因より対処したものである。またこのレポートには停止時の関連する設備内部状態も記録されるため、状況分析と対処が早くなる、対策の効果についての事後確認も同時に可能である、などのメリットもある。その結果“要因別推移”に示すように設備稼働(停止)の改善を進めること

ができた。これは週単位での要因別の停止時間推移を示しているが、従来の同様設備の立ち上げ経験に基づいて立案された改善計画に比べおよそ 50%の短期間で稼働率の向上を達成する結果を得ている。

ほかにも、リアルタイムにサーバに蓄積される各製品検査値の傾向管理を行い、生産設備の異常が予測されるような場合には速やかに事前保守を実施して、異常停止してから原因となるデータを収集していた従来に比べライン停止時間の大幅な短縮を実現。また、工程を最適化するため、各工程や作業員ごとの細分化された作業時間について装置情報をもとにして収集し、ラインバランスの調整を行うことで仕掛かりを削減、作業性上の課題の抽出、さらには作業員特性をもとにした作業分担の適性化など、多くの改善に結び付けることができた。

6. おわりに

本論文では、設備コントローラに SQL 生成機能を持たせ情報システムと装置の高効率な接続を可能とするユニット開発について述べた。またユニットを実際の工場に適用することで得られた成果について報告を行った。

生産システムにおいて設備が自律的に動作するためには、データのダウンロードと収集だけでなく現場の稼働を操るためのインタラクティブで動的な情報交換が必要となると考えられる。たとえばアクティブデータベース^{14),15)}を生産システムの制御に使おうとする研究¹⁶⁾などが試みられているが、実際の装置を自律的に連携させるためには機器自身がデータ生成ルールを持ち、現場事象をイベント/コンディションとしアクションとしてデータベース操作を行う必要がある。今後、本報告をこの一要素として発展させることで、生産システムにとってさらに効果的な機器となるのではないかと考えている。

謝辞 本論文の実装に協力いただいた三菱電機(株)名古屋製作所の諸氏に謹んで感謝の意を表する。

参考文献

- 1) Watson, D.G., Harrison, T.V., Raghuraman, K.P. and Coffman, M.G.: Missed Data Rates of Data Transfer Methods from Programmable Automation Controller to SQL Database, *Journal of Information Technology in Agriculture*, Vol.1, pp.11-21 (2006).
- 2) OPC Foundation: *OPC DA 2.05a Specification* (1994).
- 3) Daneels, A. and Salter, W.: What is SCADA?, *International Conference on Ac-*

celerator and Large Experimental Physics Control Systems, pp.339-343 (1999).

- 4) Kato, R., Kashiwagi, S., Yamamoto, T., Suemine, S. and Isoyama, G.: Data Logging System for LINAC Operation Using ORACLE Database, *Proc. 1st Annual Meeting of Particle Accelerator Society of Japan and the 29th Linear Accelerator Meeting in Japan*, pp.498-500 (2004).
- 5) 津田 学, 藤井 徹: 自動組立工程における稼動情報収集・解析, *OMRON TECHNICS*, Vol.47, No.1, pp.8-14 (2006).
- 6) 西山佳秀, 藤井昌明: PLC サンプルング・ユニットの開発, *OMRON TECHNICS*, Vol.44, No.2, pp.92-95 (2004).
- 7) 高木英明: ポーリングモデル: 巡回サービス多重待ち行列, *オペレーションズ・リサーチ: 経営の科学*, Vol.41, No.2, pp.108-118 (1996).
- 8) 河合栄治, 門林雄基, 山口 英: 多重化 I/O の実行間隔制御による効率化手法, 第 6 回プログラミングおよびシステムに関するワークショップ (SPA'03), 日本ソフトウェア科学会 (2003).
- 9) Ramamritham, K.: Real-time databases, *Distributed and Parallel Databases*, Vol.1, No.2, pp.199-226 (1993).
- 10) 三菱電機(株): C 言語コントローラユニットユーザズマニュアル(詳細編) Q06CCPU-V/Q06CCPU-V-B/SW3PVC-CCPU-J (2005).
- 11) 五島洋行: 多チャネル・リアルタイム時系列データサーバ, 電子情報通信学会論文誌 D, Vol.J88-D1, No.2, pp.316-325 (2005).
- 12) Moyne, J., Parrott, J., Kalappa, N. and Li, Y.-S.: Practical Aspects Impacting Time Synchronization Data Quality in Semiconductor Manufacturing, *Proc. AEC/APC Symposium* (2006).
- 13) Mills, D.L.: *Computer Network Time Synchronization: The Network Time Protocol*, CRC Press (2006).
- 14) 土田正士, 小寺 孝: SQL2003 ハンドブック, ソフト・リサーチ・センター (2004).
- 15) 石川 博: アクティブデータベース, *情報処理*, Vol.35, No.2, pp.120-129 (1994).
- 16) 松本卓也, 加藤義晃, 長舟伸也, 若松栄史, 白瀬敬一, 荒井栄司: アクティブデータベースを用いた自律分散形生産システムの開発, 日本機械学会, 日本機械学会論文集 C 編, Vol.65, No.630, pp.837-843 (1999).

(平成 20 年 12 月 20 日受付)

(平成 21 年 4 月 8 日採録)

(担当編集委員 相良 毅)



渡部 裕二 (正会員)

1980 年京都大学大学院工学研究科精密工学専攻修了。同年三菱電機入社。名古屋製作所開発部において製造業向け FA 機器およびシステムの開発に従事。日本工業標準調査会委員, NPO ものづくり APS 推進機構理事, FA オープン推進協議会委員, OPC 協議会幹事会員。



石川 博 (正会員)

静岡大学情報学部情報科学科教授。東京大学理学部情報科学科卒業。東京都立大学を経て 2006 年より現職。東京大学博士 (理学)。著書に『次世代データベースとデータマイニング—DB & DM の基礎と Web・XML・P2P への適用』(CQ 出版社), 『JavaScript によるアルゴリズムデザイン—オブジェクト指向から DB・Web・マイニングまで』(培風館), 『データベース』(森北出版)等。国際論文誌 ACM: TODS, IEEE TKDE, 国際学会 VLDB, IEEE: ICDE 等学術論文多数。1994 情報処理学会坂井記念特別賞, 1997 科学技術庁長官賞 (研究功績者) 受賞。情報処理学会データベースシステム研究会主査, 情報処理学会 (データベース) 共同編集委員長, International Journal Very Large Data Bases Editorial Board, 日本データベース学会理事歴任。電子情報通信学会, ACM, IEEE 各会員。



水野 忠則 (正会員)

1945 年生。1969 年名古屋工業大学経営工学科卒業。同年三菱電機(株)入社。1993 年静岡大学工学部情報知識工学科教授。1996 年情報学部情報科学科教授。2006 年より創造科学技術大学院教授。工学博士。情報ネットワーク, モバイルコンピューティング, 放送コンピューティングに関する研究に従事。著訳書としては『コンピュータネットワーク』(日経 BP), 『モダンオペレーティングシステム』(ピアソン・エデュケーション)等がある。電子情報通信学会, IEEE, ACM, Informatics Society 各会員。