分散型セル制御用基盤システムの実現

電気通信大学 総合情報処理センター　高田昌之

要　旨

生産システムの制御を分散環境で実現するには、その要素たる複数の制御システムがデータを共有し、外部で発生した事象を統一的に扱える事が望ましい。更にシステムの拡張性や保守性、ソフトウェアの再利用性等を考えると、プログラムがモジュール構造を持つ事が望まれる。本報ではこの目的の為に設計した “Glue Logic” と、そのプロトタイプの実現の詳細についてアプリケーションの作成事例を通して説明する。

1　はじめに

ＦＭＳなどによる多品種小量生産システムの制御を、可能な限り人間の手を煩わす事無く効率良く行なう為には、各セルの制御装置を分散システムで構成し、実時間で動的な制御を行なう事が重要であると考えられる。

このような分散処理能力を実現する為には、各セル制御装置間の情報の共有・交換をはじめ、外部事象の発生の検出、協調して動作するプロセスの同期やそれらの生成や消滅などを支援する処理環境と、さらにはその環境を生産技術者が容易に利用できるようにする為のプログラミング言語処理系 / 実行時環境との整備が必要である。

前報(参考文献 1)では分散型制御基盤システムである Glue Logic の機能と、これを利用するシステムが持つべきプログラミング・パラダイムについて述べた。本報では Glue Logic の実現と、これを利用した簡単な生産ラインのシミュレーション・システムについて述べる。

2　Glue Logic の機能

以下に現在構築中の Glue Logic の機能を示す。

この Glue Logic は、生産制御プログラムを小さなモジュールの集合として構築し、これらの間でのメッセージ交換やデータ共有を行なう事で、システム全体を事象駆動型のエージェントの集合体として柔軟に結合する事ができる、糊のような機能を持つシステムとして設計された。詳細については参考文献 2 を参照されたい。

プロセス間通信機能 基盤システムが全てのプロセス間通信を中継することで、ＯＳや通信方式に依存しないエージェント間データ通信機能を提供する。また、セル内 / セル間での差異を吸収する。

名前による動的構造データ管理 共有データの集中管理を行ない、構造を持ったデータをシンボリックな名称のみで参照し、管理できる。

データ共有・同期参照・条件付き更新 複数のエージェントが利用するデータが共通の Glue Logic で管理されていれば、それらのデータは共有される。相互に関連するデータ群を整合性を保ったまま更新できたり、排他制御を行なう為に特定の名前のデータが特定の値である時にのみデータを更新できることが必要である。

データ変更通知 Glue Logic が管理するデータが変更された場合、その事実をあらかじめ要求を出していたエージェントに通知する。この結果、ポーリングの必要が無くなる。

依存データ更新 Glue Logic が管理するデータ間に依存関係が定義されていた場合、基となるデータに変更があった場合には自動的にそのデータに依存するデータを再計算する。データ変更通知と組み合わせると、データ間に特定の関係が成立した時に通知を行なうことができる。

3　Glue Logic の実現

現在はプロトタイプ版として、機能の変更が容易な perl5 による実現を行なっている。今後はこの版によって機能的な検証を進め、ある程度機能仕様が固まった所でＣ言語による実現に切替える予定である。

また、現在のサーバ・クライアント間のプロトコルも、デバッグなどを容易にするために ASCII 文字列を基にしたものを用いている。これも最終的には、情報の転送効率を高めるため、バイナリ・データによって表現する事にする予定である。

プロトタイプ版として実現された現在のシステムは、以下のような特徴を持っている。

1. プロトタイプ・システムは、データの再計算などに用いられる演算式の記述能力に一部の制限があるほかは、実用 / 評価システムと同等の機能を持つ。
2. すべての共有データは特定のプロセッサ上で稼働するひとつのサーバ・プロセスで管理する。
3. 全てのエージェントはこのサーバ・プロセスとのみ通信する。このため、できたシステムは、サーバ・プロ

セスやそれが稼働しているプロセッサの障害に対する冗長性を持たない。

4. 機能的な評価を行なうことを主目的としており、処理速度を速くする事は特に問題とはしていない。
5. 移植性や可読性を高める為、サーバ，クライアント API ともインタプリタ言語である perl5 で記述する。この言語は非常に移植性が高く、各種の OS 上で稼働させることができると共に、外部のプログラムとの接続性も高く、X window system を直接操作できるパッケージ等も用意されている。
6. 非常に高機能のプログラミング言語を用いた事で、サーバ，クライアント API モジュールとも非常にコンパクトになり、実験の為に機能を追加・変更することも容易である。サーバは 700 行強，クライアント側 API モジュールは 500 行弱の大きさである。
7. 各クライアントはサーバと二つの TCP/IP socket を介して接続される。ひとつはトランザクションの処理に、もうひとつはデータ変更通知に使用する。

4 アプリケーション作成例

このプロトタイプ・システム上に、ごく単純な生産ラインシミュレーション・システムを作成した。以下にその概要を示す。

4.1 シミュレーションの内容

一つの生産ラインに数種類の製品種を定義し、それぞれについて一つまたは二つの部分生産ラインを用意する。全種共通の入力キュー（待ち行列）から入ってきた素材に対して対応する部分ラインで作業を行ない、最終的に共通出力キューに吐き出すものである。各部分ラインの中での分流・合流は行なわなかった。

4.2 アプリケーションの実現方法

このアプリケーション全体は、クライアント API モジュール同様に perl5 で記述した。X window system とのインタフェースには Tk モジュールを用い、シミュレーションの過程をすべて graphic に表示するようにした。

アプリケーション・プログラムは、作業を行なう『プロセス』と、プロセス間を繋いだり複数個が直列に継って待ち行列を構成したりする『バッファ』との二種類のエージェントからなり、これらは独立したプログラムの体裁を持っている。どちらも内部でクライアント API モジュールを呼び出しており、アプリケーションのソースコードにはプロセッサ間、あるいはプロセス間の通信を行なっている事を窺わせる記述はほとんど無い。

これらの相互関係はすべて、 Glue Logic 内部の関連する名前に与えられた値、および Glue Logic を利用するエージェントに与える引数として与える。シミュレーションの対象となる生産ラインの仕様を変える時にも、プロセスやバッファの挙動を記述したエージェントのプログラムには一切変更を加える必要がなかった。

4.3 エージェントに与える情報

プロセスやバッファの挙動を記述したエージェントのプログラムには、アンカーと名付けた Glue Logic 内の名前を対応付ける事とした。この名前は対応するエージェントに対する細かな仕様や、内部状態を保持する構造を持ったデータで、この値を適宜変更する事によって各エージェントの振舞いを制御した。また、ここに指定した名前を使って、エージェントにメッセージを送ったり、そのメッセージに対する応答を得たりする事もできた。

シミュレーションの対象の状態変化は、すべて Glue Logic の名前の値の変化によって起動される。この変化はすべて Glue Logic からメッセージによって通知されるため、一つのエージェントが複数のプロセスやバッファのシミュレーションを行なう事ができる。実際に一つのエージェントがカバーするオブジェクトの数は、エージェントのプログラムとは全く独立に制御することができた。

5 おわりに

本講演ではプロトタイプ版の分散型セル制御用基盤システムの概要を説明した。また、これを用いたシステム構築の一例を示した。

なお、現在 Glue Logic に関連する各種のドキュメントを World Wide Web によって公開している。 URL は `http://www.tl.cc.uec.ac.jp/Glue/` である。また、準備が整い次第、プロトタイプ版のソース・コードの公開も始めたいと考えている。

References

1. 高田昌之: "自律分散型セル制御用プログラミング環境", 1995 年度精密工学会 秋季大会講演論文集 D-34, pp.425-426 (9/30/1995)
2. Takata, M.: "An Integreated Environment for Factory Automation", *Integrated Computer-Aided Engineering*, John Wiley & Sons, Inc., Vol.2, No.4, pp.249-263 (1995)
3. 高田昌之: "オブジェクト指向による FA ソフトウェアの開発", 精密工学会誌, Vol.58, No.10, pp.1649-1651 (1992)
4. 高田昌之, 長谷川雅樹, 松家英雄: "近未来の Shop Floor Control System", 1991 年度精密工学会 春季大会講演論文集 D-37, pp.617-618 (3/27/1991)