

オブジェクト指向デザインパターンを用いた 分散制御システムの3次元協調シミュレーション

旭川工業高等専門学校
北海道大学大学院工学研究科
モトローラ株式会社

○戸村 豊明
金井 理, 岸浪 建史
伊深 和浩, 上広 清, 山元 進

要 旨

近年, FA や BA において分散制御システム (DCS) が導入され始めている. FA 用 DCS のように, ネットワーク特性だけでなく制御対象の視覚的挙動も検証したい場合, DCS シミュレーションモデルと制御対象の3次元形状モデル間のイベント連鎖を実現するソフトウェアが必要となる. そこで本研究では, シミュレーションモデルと制御対象の3次元形状モデル間のイベント連鎖メカニズムを提案し, それを実装した DCS の3次元協調シミュレータを開発する.

1. はじめに

近年, FA や BA (ビルオートメーション) などの分野において, 図1に示すように, 多数の制御ノードをオープン規格のネットワークで相互接続した分散制御システム (DCS) が導入され始めている.

事象駆動型の制御ソフトウェアが実装される各制御ノードとそのセンサ・アクチュエータの挙動は, 固有の有限状態機械とそれらの間のイベント連鎖として記述できる. これまで我々は, オブジェクト指向デザインパターン¹⁾に基づき DCS シミュレーションモデルを構築し, 各制御ノードの通信パラメータを設定する事により, 制御ノード間のトラフィックによる通信遅れやパケット損失を予測できる DCS シミュレータを開発してきた²⁾.

一方, FA 用 DCS のように, ネットワーク特性だけでなく, 制御対象の視覚的挙動も精密に事前検証しなければならない場合, 有限状態機械ベースの DCS シミュレーションモデルと制御対象の3次元形状モデル間のイベント連鎖を実現できるソフトウェアも必要とされる.

そこで本研究では, 我々が提案してきたデザインパターンに基づき記述された DCS シミュレーションモデルと, Java3D³⁾ で記述された制御対象の3次元形状モデルの間のイベント連鎖メカニズムを提案し, そのメカニズムを実装した DCS3次元協調シミュレータを開発する.

2. オブジェクト指向デザインパターンに基づく DCS のモデリングとシミュレーション

本研究では, 図2に示すような DCS モデリング・シミュレーション手続き⁴⁾を提案している. DCS の要求分析・設計 (図2の①~④) において, UML⁵⁾ (Unified Modeling Language) によって記述される DCS 設計モデルは, 以下の手順に従ってモデリングされる.

- ① DCS の機器構成仕様から, DCS における制御装置と制御対象は, 各々いくつかのサブシステムへ分解される. また, DCS の動作仕様から各サブシステムが持つべき機能として, いくつかのユースケースが定義される.
- ② DCS の機器構成仕様から, DCS に固有のセンサ・アクチュエータをそれぞれ具象クラスとして定義した後, それらの具象クラスのオブジェクトを組み合わせる事により, DCS の静的構造が定義される.
- ③ ①で定義された各ユースケースを実現するためのオブジェクト間の相互作用を定義する.

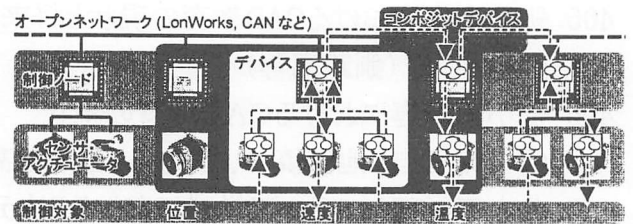


図1. DCS の構造と挙動

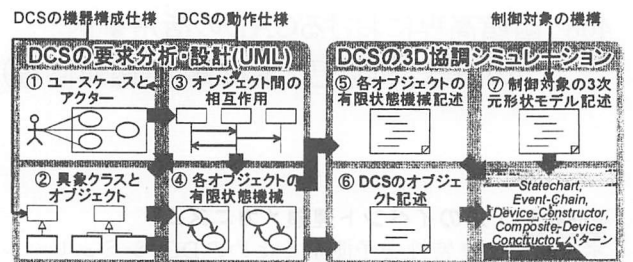


図2. 本研究で提案するデザインパターンに基づく
DCS モデリング・シミュレーション手続き

- ④ ③において定義された相互作用同士を組み合わせ, 各オブジェクトが送受信するメッセージを抽出する事により, 各オブジェクトの状態・イベント・アクションからなる有限状態機械を定義する.

図2における⑤~⑦のテキスト記述は, 本研究で独自に提案した構文に基づいて記述されており, これらをテキストファイルとして3次元協調シミュレータへ与える事により, FA 用 DCS の3次元協調シミュレーションを実行する事ができる. DCS シミュレーションモデルの静的構造と動的挙動を規定するために, これまで我々が提案してきた以下に示す4種類のデザインパターン²⁾が, 3次元協調シミュレータの内部へ実装されている.

- ・ **Device-Constructor パターン** : センサ・アクチュエータの種類と個数を指定する事により, デバイスモデルの静的構造を規定する.
- ・ **Composite-Device-Constructor パターン** : デバイスの種類と個数を指定する事により, コンポジットデバイスモデルの静的構造を規定する.
- ・ **Statechart パターン** : 制御ノード・センサ・アクチュエータモデルの有限状態機械を規定する.
- ・ **Event-Chain パターン** : 有限状態機械間のイベント連鎖のメカニズムを規定する.

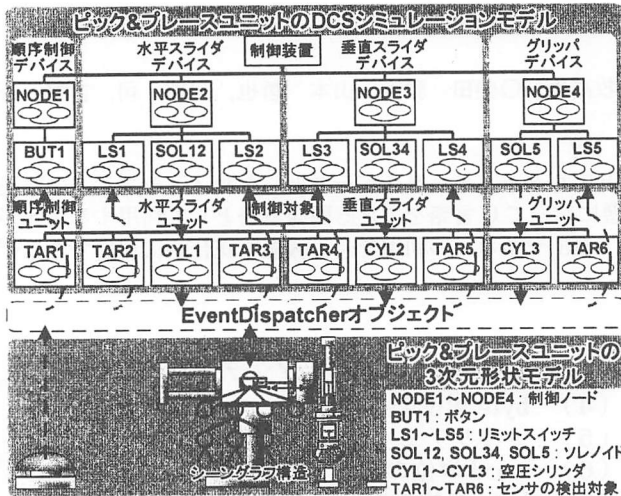


図 3. DCS シミュレーションモデルと制御対象の 3 次元形状モデル間のイベント連鎖メカニズム

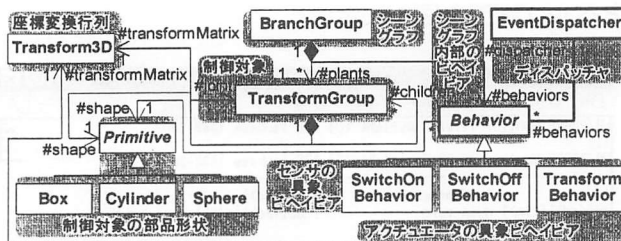


図 4. 制御対象の 3 次元形状モデルの UML クラス図

3. DCS シミュレーションモデルと制御対象の 3 次元形状モデルのイベント連鎖メカニズム

ネットワーク特性を予測するための DCS シミュレーションモデルと制御対象の 3 次元モデル間のイベント連鎖メカニズムを図 3 に示す。図 3 は、ピック&プレースユニット用 DCS を例とした場合であり、図 2 の⑤～⑦の記述に応じて、多種多様な FA 用 DCS の 3 次元協調シミュレーションを実行できる。図 3 における *EventDispatcher* オブジェクトは、DCS シミュレーションモデルと 3 次元形状モデルの間のイベント連鎖の仲介を行なう。3 次元形状モデルを必要としないシミュレーションの場合は、代わりに *TAR1*~*TAR6* がセンサへのイベントを生成する。また、図 4 は制御対象の 3 次元形状モデルのクラス図である。*Behavior* クラスの各具象クラスはシーングラフにおけるセンサ・アクチュエータの具体的ビヘイビアであり、*EventDispatcher* クラスを介して、図 3 上部のシミュレーションモデルのセンサ・アクチュエータの具象クラスとイベント連鎖する事により協調動作を実現している。

図 5 はピック&プレースユニット用 DCS における DCS シミュレーションモデルと 3 次元形状モデル間のイベント連鎖の例を示している。*CLOCK* からの *tick* イベントに応じて、*CYL1* が *dispatcher* を介して *CYL1B* へ位置 *position* を繰り返し送信する事でシーングラフのシリンダが前進し、その後 *LS2ONB* が *LS2* へ *on* イベントを送信する事で、シミュレーションモデルの水平シリンダが停止する。

図 3 のメカニズムを実現する 3 次元協調シミュレータの実行例を図 6 に示す。このシミュレータは Java で開発されており、制御対象の 3 次元形状モデルは図 4 の構造を持つ Java3D のシーングラフ³⁾ として実装されている。

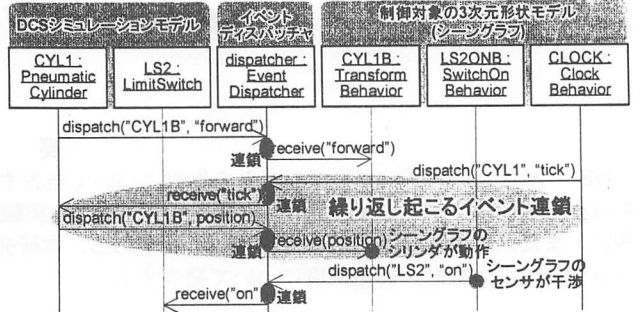


図 5. ピック & プレースユニット用 DCS におけるシミュレーションモデルと 3 次元形状モデル間のイベント連鎖

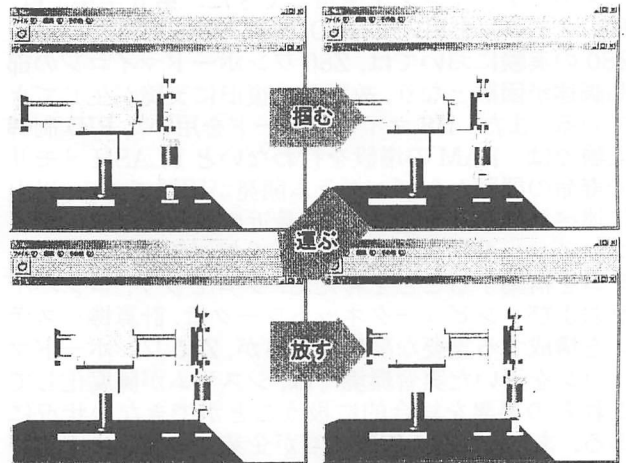


図 6. ピック & プレースユニット用 DCS の 3 次元協調シミュレーションの実行例

4. まとめと今後の課題

本研究では、FA 用 DCS の 3 次元協調シミュレーションを実現するメカニズムを提案し、そのメカニズムを内部に実装した Java3D ベースの 3 次元協調シミュレーションソフトウェアを開発した。

今後は、並列分散事象システムのシミュレーション機構であるタイムワープ機構を DCS シミュレーションモデルへ適用する方法について議論する予定である。

参考文献

- [1] E. Gamma, R. Helm, R. Johnson, J. Vlissides : オブジェクト指向における再利用のためのデザインパターン, ソフトバンク, 1999.
- [2] T. Tomura, S. Kanai, T. Kishinami, K. Uehiro, S. Yamamoto : Developing Simulation Models of Open Distributed Control System by Using Object-oriented Structural and Behavioral Patterns, Proceedings of Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2001), pp. 428-437, 2001.
- [3] H. Sowizral, K. Rushforth, M. Deering : The Java 3D API 仕様, 株式会社アスキー, 1999.
- [4] 戸村, 金井, 岸浪, 伊深, 上広, 山元 : UML とデザインパターンを利用した FA 用分散制御ソフトウェアの迅速開発, 2002 年度精密工学会春季大会学術講演会講演論文集, p. 54, 2002.
- [5] Object Management Group : OMG Unified Modeling Language Specification Version 1.4, OMG, 2002.