3Z-8

UMLとデザインパターンによる分散制御シミュレーションモデルの設計と実装 *

戸村 豊明 †
旭川工業高等専門学校

金井 理 ‡
北海道大学大学院 工学研究科

上広 清, 山元 進 §
モトローラ株式会社

1. はじめに

近年，FA・BA（Building Automation）で，制御ノード数が数万に及ぶ分散制御ネットワークが導入されており，通信遅れや通信トラフィックに応じた制御特性を評価できる分散制御シミュレータが不可欠なものとなっている．

分散制御ネットワーク上の制御ノードに相当するデバイスは複数のセンサ・アクチュエータからなる集合体であり，類似した機器構成を持つデバイスが繰り返し大量に現れるモデルを効率的に設計する必要がある．我々は，オブジェクト指向に基づいて，UMLのstatechart図で記述されたセンサ・アクチュエータ単体の挙動を設計するデザインパターン[1]，デバイスを構成するセンサ・アクチュエータの種類・個数をパラメータとして，デバイスモデルを柔軟かつ効率的に設計するためのデザインパターン[2], [3]を提案してきた．現在，ネットワークを介したセンサ・アクチュエータ間のイベント連鎖を柔軟かつ統一的に設計でき，かつ定義済みのイベント連鎖を持つデバイスモデルを再利用できるデザインパターンが必要とされている．

本報では，UMLに基づき記述されたアクチュエータ・センサ・デバイスモデルにネットワーク変数を割り当てて，変数間のバインディング情報を定義する事により，ネットワークシミュレータ上でのモデル間イベント連鎖を実装するためのデザインパターンを提案する．さらに，このパターンに基づき，多重に階層化されたデバイスモデル，コンポジットデバイスモデル間のイベント連鎖を統一的に実装可能なJavaコードの開発手法を提案する．

2. 分散制御シミュレーションモデル構築のためのパターン

我々はSEMIのCommon Device Model[4]に基づき，分散制御システム（DCS）のシミュレーションモデルを構築するために，以下のデザインパターンを提案してきた．

- Statechart パターン[1]：statechart図で記述されたセンサ・アクチュエータの挙動を設計するパターン．
- Device-Constructor パターン[2]：センサ・アクチュエータの種類・個数をパラメータとして，デバイスモデルの構造を設計するパターン．
- Composite-Device-Constructor パターン[3]：デバイスの種類・個数をパラメータとして，コンポジットデバイスモデルの構造を設計するパターン．

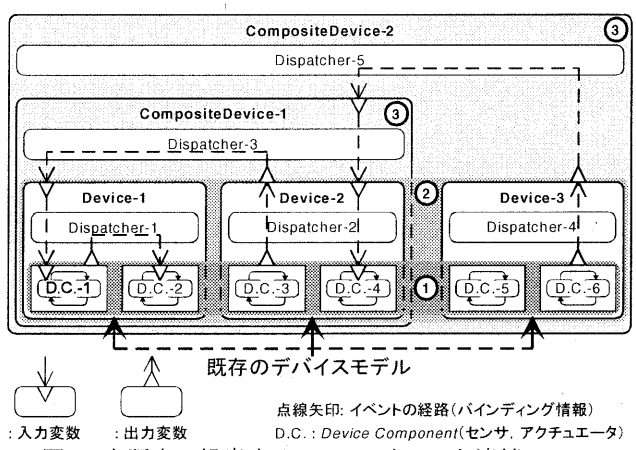

図1. 本研究で提案するDCSのイベント連鎖モデル

これらのパターンはモトローラ社開発のDCSモデラ，DCSシミュレータへJavaコードとして実装され，その有用性は実証されている．しかしながら，複雑な階層性を持つデバイスのイベント配信メカニズムを統一的に設計でき，かつデバイス単位で定義済みのイベント連鎖を再利用でき，さらに他の3種類のパターンと整合性の取れたデザインパターンが強く求められている．

3. センサ・アクチュエータ間のイベント連鎖モデル

図1に示すように，LonWorksのようなオープンネットワークを用いたDCSでは，センサ・アクチュエータの入出力へネットワーク変数を割り当てて，変数間の情報の伝達をバインディング情報として定義する．このネットワーク変数値の変化がイベントとなり，バインディング情報に従って別のネットワーク変数値の変化として伝達される．この現象のシーケンスがイベント連鎖となる．

DCSの設計において，多数のセンサ・アクチュエータへ新規にネットワーク変数を割り当て，バインディング情報を定義するのは膨大な作業となる．しかし，DCSは互いに類似構造を持つデバイスから構成される事が多いため，ネットワーク変数とバインディング情報を含めた形で既存のデバイスモデルを再利用できれば，効率的にDCSシミュレーションモデルを設計できる．そのために，デバイス自身にも外部と通信可能なネットワーク変数を

---

* Design and Implementation of Distributed Control Simulation Model Using UML and Design Patterns
† Toyoaki Tomura, Asahikawa National College of Technology, 2-2 Shunkodai, Asahikawa, Hokkaido, 071-8142 Japan
‡ Satoshi Kanai, Hokkaido University, Kita-13 Nishi-8, Kita-ku, Sapporo, Hokkaido, 060-8628 Japan
§ Kiyoshi Uehiro and Susumu Yamamoto, Motorola, 2-9-1 Akedori, Izumi-ku, Sendai, Miyagi, 981-3206 Japan

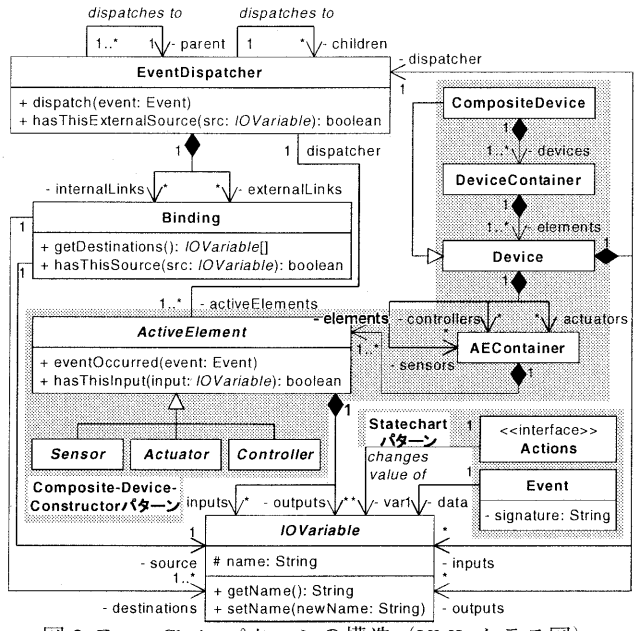

図 2. Event-Chain パターンの構造（UML クラス図）

仮想的に割り当てて，デバイスの内外へイベントを配信可能なメカニズムが，イベント連鎖の実現に必要となる．

これらより，イベント連鎖を実現するデザインパターンには，以下の条件が要求される．

1) DCS の規模に一切依存しない，同一のイベント配信メカニズムによって，全てのイベントを配信できる．
2) ネットワーク変数間のバインディング情報を含めた既存のデバイスモデルを相互接続する事で，効率的かつ階層的に大規模なデバイスモデルを設計できる．
3) Statechart ， Device-Constructor ， Composite-Device-Constructor パターンと有機的に関連付けられている．

4. Event-Chain パターン

上述の要求条件を満たす，センサ･アクチュエータ間のイベント連鎖モデルを設計するため，本研究では図 2 に示される Event-Chain パターンを提案する．本パターンにおける主なクラスの役割を以下に示す．

- **EventDispatcher** ：あるデバイス，コンポジットデバイスに関わる全てのイベント配信を管理する．
- **Binding** ：デバイスやデバイスコンポーネントが持つ入出力変数間のイベント連鎖の経路を定義する．
- **IOVariable** ：ネットワーク変数を表す．ネットワークの仕様に応じて，サブクラスを定義可能である．
- **Event** ：イベントを表す．イベントの発生源である IOVariable オブジェクトを参照している．

このパターンは，以下のような特徴によって，上述の要求条件を満たしている．

1) Device，Composite Device ごとに EventDispatcher を持つので，モデルの階層性が複雑になっても，常にイベント配信は EventDispatcher 間のみで行えば良く，配信メカニズムが統一できる．
2) Device，CompositeDevice は IOVariable を持つため，他の Device の IOVariable とバインドするだけで，大規模な Composite Device を効率的に設計できる．
3) Device と CompositeDevice は EventDispatcher と IOVariable，ActiveElement は IOVariable を持ち，IOVariable は Event と Actions に関連付けれている．

5. Event-Chain パターンを用いた DCS シミュレーションモデルの実装手続き

Event-Chain パターンを用いて，ネットワーク変数とバインディング情報を持つ DCS シミュレーションモデルを Java 言語で実装する手続きは，図 1 の①，②，③に適用される．①の手続きを以下に示す．

1) 図 2 の *Sensor*, *Actuator*, *Controller* のサブクラス SC_1 を定義した後，各 SC_1 の入出力変数を特定し，*IOVariable* のサブクラス SC_2 として定義する．
2) 各 SC_1 のコンストラクタへ，SC_2 のオブジェクトを生成する手続きを実装する．
3) 各 SC_1 の statechart 図において，入力変数の変化をイベント，出力変数の変化をアクションとして記述し，Statechart パターンを用いて，それを実装する．

②の手続きを以下に示す．

1) 各 Device オブジェクト O_1 における SC_1 の全オブジェクト O_2 について，O_2 のネットワーク変数が同じ O_1 に属する O_2 と通信する時は，両変数を結合する internalLinks の Binding オブジェクトを生成する．そうでなければ，O_1 に対して新規に SC_2 のオブジェクト O_3 を生成し，O_3 と O_2 の両変数を結合する externalLinks の Binding オブジェクトを生成する．

③の手続きは②と同様であり，これを再帰的に繰り返す事で，既存の Device オブジェクトをクローン化によって効率的に再利用しながら，より大規模な DCS シミュレーションモデルを設計する事ができる．

6. デザインパターン適用の効果

本報で提案する Event-Chain パターンを含む 4 つのデザインパターンは，筆者らが開発した DCS シミュレータへ応用されている．これらのパターンに基づきシミュレータ開発を行った結果，開発期間を 4 ヶ月から 2 ヶ月へ大幅短縮する事ができた．また，これまでイベントキューにより実装されていたセンサ･アクチュエータ間のイベント連鎖を Event-Chain パターンへ置き換えた事により，シミュレーションの実行速度は大幅に改善された．現在，本シミュレータは，空調制御システム等の設計に活用されており，実用的な規模および精度でシミュレーションが可能な事が確認されている．

この結果から，提案する分散制御シミュレーション用デザインパターンの有効性が確認された．

参考文献

[1] 戸村，金井，岸浪，上広，山元："超分散システム制御ネットワークシミュレータの開発（第 2 報）—状態遷移仕様のデザインパターンを用いた制御対象モデルの迅速開発—"，2000 年度精密工学会春季大会学術講演会講演論文集，p. 71，2000.

[2] 戸村，金井，岸浪，上広，山元："オブジェクト指向デザインパターンを活用した超分散システム制御ネットワークシミュレータ用デバイスモデルの迅速構築"，2000 年度精密工学会秋季大会学術講演会講演論文集，p. 499，2000.

[3] 戸村，金井，岸浪，上広，山元："オブジェクト指向デザインパターンに基づく超分散システム制御ネットワークシミュレータ用コンポジットデバイスモデルの迅速構築"，2000 年度精密工学会北海道支部学術講演会講演論文集，pp. 28-29，2000.

[4] SEMI："SEMI E54.1-0298 Standard for Sensor/Actuator Network Common Device Model"，SEMI，1996.