3K−7　オブジェクト指向による仮想化外部装置を用いたFAソフトウェア開発手法　- 実現手法 -

硴崎賢一[+]　松本俊哉[+]　井上郁[++]　大畑浩司[++]　ペセモンテネグロマヌエルヘスス[+]　打浪清一[+]
[+]九州工業大学情報工学部　[++]安川情報システム（株）制御システム事業部

1．はじめに

FAソフトウェアの重要な位置を占める外部装置のプログラミングは，その対象が多種多様にわたり機能や操作法が統一されていないという問題があるために，非常に困難なものとなっている．本報告では，仮想化外部装置[1]の要求分析に基づき，その実現手法について示す．

2．仮想化外部装置の構成

仮想化外部装置に要求される以下の5項目を満たす仮想化外部装置の構成法を示す．

1）抽象化
2）標準化
3）個別化
4）自律化
5）組織化

仮想化された外部装置は，デバイスユニットと呼ばれ，ソフトウェア的に構成されるデバイスオブジェクトと，外部装置のハードウェアから構成される．以下の節では，自動倉庫に使用されるクレーン，コンベア，カートラックなどの外部装置を例として，これらの機能を簡単に説明する．

2．1　抽象化

自動倉庫の外部装置は，物品を運ぶ共通の機能を持ち，その機能は以下のように分類，定義される．

1）荷物の取得
2）指定位置への移動
3）荷物の格納

外部インターフェースとしては，以下の様なものが必要である．

1）動作の指示
2）状態の取得
3）制御情報の設定
4）割り込み

外部装置の仮想化には，C++ のクラス定義機能を用いる．

2．2　標準化

標準化は，C++ の継承機能で実現する．継承機能を用いた標準化では，外部装置の共通の特性を基底クラスとしてまとめ，各デバイスユニットはこのクラスを継承する派生クラスとして作成する．各派生クラスでは，そのクラスに特有な機能の処理を，基底クラスとの差分として追加，修正する．この様に継承機能を利用すると，基底クラスが共通するクラスのオブジェクトでは，機能や外部インターフェースが自然に標準化されると共に，新たなデバイスユニットの作成が容易に行えるという利点がある．

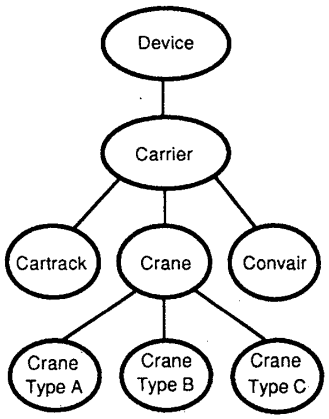

図1　継承による標準化

ここで，カートラック，コンベア，クレーンを仮想化したデバイスユニットをそれぞれ Cartrack, Convair, Crane とする．これらのデバイスユニットは物品を運搬する機能を共通に持つので，まず，運搬機能を持つクラスを Carrier クラスとして定義し，Cartrack, Convair, Crane の各クラスは，Carrier の派生クラスとして定義する．さらに，物品を運搬するものに限定せず，外部装置に共通する基本的な機能だけを持つクラスを Device クラスとして定義すると，Carrier クラスは Device クラスの派生クラスとして定義することができる．この様子を図1に示す．

Factory Automation System Development Methodology using Object-Oriented Virtual Devices, - Implementation -

Ken'ichi KAKIZAKI[+], Toshiya MATSUMOTO[+], Kaoru INOUE[++], Kouji OHATA[++], PECE MONTENEGRO Manuel Jesus[+], Seiichi UCHINAMI[+]
[+]Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, [++]Yasukawa Joho System K.K.

2．3　個別化

個別化の機能は，C++ の多重継承を含む継承機能，差分の追加と修正で実現する．多重継承を利用すると，継承による標準化だけではなく，複数のクラスを同時に継承することによる個別化をはかることも可能になる．

2．4　自律化

自律化を実現するために，まず，比較的低機能の処理を保護されたメンバ関数として定義し，公開するメンバ関数は，これらの関数の組み合せで機能を実現するようにする．この様にすると，外部からは，自律的にふるまう高機能のインターフェースのみが見えるようになり，低機能の関数を隠蔽することができる．また，この様にして作成されたデバイスユニットを基底クラスとして継承するデバイスユニットでは，低機能の処理を組み合わせて新たな機能を容易に作成できるようになる．

2．5　組織化

デバイスユニットの組織化を実現するための方式として，コールバックと呼ばれる機能を用いる．コールバック機能は状態，手続き，手続きに渡すデータの組み合せをオブジェクトに設定するもので，オブジェクトの内部状態が指定した状態になった場合に，指定した手続きが指定したデータを伴って呼び出される．この様にして呼び出される手続きはコールバックルーチンと呼ばれる．コールバックルーチンの起動条件としての状態は，抽象化の際に定めた状態をそのまま利用できる．

コールバック機能を利用する方式では，システムの初期化の多くの部分は，各デバイスオブジェクトへのコールバックルーチンの設定によって行われることになる．この方式は，デバイスオブジェクトの組織化を実現できるだけでなく，コールバックルーチンを設定する系列が，プログラムにおける外部装置間の関係を示すことになり，その系列を参照することによって，システムの全体的な構成を一目で把握することができるという利点も生じる．

コールバックルーチンの設定は，必要に応じて実行時に行うこともできる．また，コールバックルーチンの内容はどのような処理でも記述できるために，大局的な判断を行いながら処理を進めたい場合には，コールバックルーチンの内容として，スケジューラ等の管理機構に問い合わせを行いながら処理を遂行するような手続きを記述することで柔軟に対処することができる．

3．仮想化外部装置を用いた開発の支援

3．1　仮想化外部装置の使用者支援システム

オブジェクト指向によるシステム開発では，各クラスの機能やインターフェースを把握するために，クラスの階層関係や利用できるメンバ関数などを容易に参照できる必要がある．また，メンバ関数の上書きや多態性により，処理の実際の定義の参照が困難であったり，メッセージの流れや，データメンバの定義と使用を把握しにくいといった問題が生じる．したがって，仮想化外部装置を用い，オブジェクト指向でシステムを設計，記述するためには，オブジェクト指向に特有なこれらの情報を効率良く参照するための支援システムが必須と考えられる．

ソフトウェアの生産性向上のために CASE ツールを用いた設計開発手法が注目されている．CASE ツールでは，システムの分析や設計を行うために，データフロー図を記述して，データの流れと処理主体の関係を定義する．データフロー図で表されるデータの流れをメッセージ，処理主体をオブジェクトととらえると，オブジェクト指向の考え方を視覚的に表現したものと考えることができる．

データフロー図に示される処理主体の間の関係は，デバイスユニット間のコールバックルーチンと対応付けることができる．この様な対応付けを行うと，CASE で記述したデータフロー図からコールバックルーチンの定義系列を生成し，実行可能なコードを直接生成できると考えられる．

3．2　仮想化外部装置の開発者支援システム

外部装置の仮想化を進めるためには，仮想化外部装置を開発するための支援システムも必要である．この様な支援システムでは，外部装置の機能とインターフェースを分析すると共に，すでに定義されている近い機能を持つ仮想化外部装置との関係を調べ，標準化に適し，プログラムの新規の記述が少なくてすむようなクラス階層，機能，インターフェースを効率よく定義するための機能が望まれる．また，仮想化外部装置の定義から，シミュレータの基本部分を自動的に生成する機能も必要であると考えられる．

4．まとめ

FA ソフトウェアの生産性と品質を向上させる上でボトルネックとなっている外部装置をオブジェクト指向により仮想化する実現手法を示した．

参考文献

[1] 砥崎賢一 他: オブジェクト指向による FA 用外部装置の仮想化手法，情報処理学会，ソフトウェア工学研究会資料，83-5, (1992).