3K－8　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　-評価システム-

硴崎賢一[+]　井上郁[++]　大畑浩司[++]　ペセモンテネグロマヌエルヘスス[+]　松本俊哉[+]　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．この様な問題を解決するために，我々はオブジェクト指向に基づき仮想化した外部装置を用いた開発手法を提案している．本報告では，仮想化外部装置[1]を用いた開発の有効性を確認するために作成した，自動倉庫のシミュレータの実現手法とその評価について示す．

2．シミュレーションの概要

シミュレーションは，自動倉庫の基本的かつ一般的な構成を選び，以下の3種6台の外部装置を用いたものとなっている．

1）クレーン1台
2）カートラック1台
3）コンベア4台

3種類の外部装置を用いることにより，抽象化や自律化だけではなく，標準化，個別化，組織化の評価も行えるように考慮している．

シミュレートする自動倉庫の構成を図1に示す．この構成では，入出庫のスケジューリングの自由度が高くなるように，コンベアを入庫用と出庫用に別系統として用意している．また，入出庫口とクレーンの設置位置を自由に設定できるようにするために，それらの中間にカートラックを設置している．

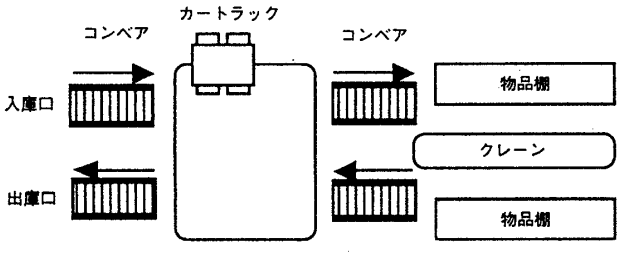

図1　シミュレーションシステム

シミュレータでは，マウスを利用して指定した物品棚に対する入出庫の指定ができるようになっており，指示に従って複数の物品の入出庫を並行して行えるようになっている．

3．システム構成

自動倉庫のシミュレータは，UNIX　ワークステーション上で作成した．

3．1　クラス構成

3種類の外部装置に対応するデバイスオブジェクトCartrack, Crane, Convairは物品を運ぶという共通の機能を持つため，図2に示すように物品を運ぶ外部装置の抽象クラスであるCarrierの派生クラスとして定義している．

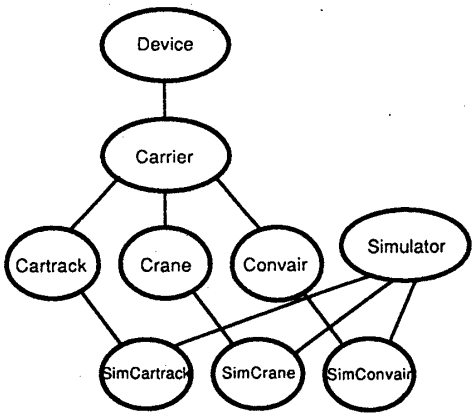

図2　クラス階層

図2のクラス階層では，物品を受け取り，移動し，受け渡すための基本的な機能や，指定された位置まで自律的に移動する機能などは，物品を運ぶ外部装置に共通の機能としてCarrierクラスに定義している．また，デバイスオブジェクトが連係して処理を行うための組織化機能は，デバイスオブジェクトに共通の機能として，Deviceクラスに定義している．

Factory Automation System Development Methodology using Object-Oriented Virtual Devices, - Evaluation -
Ken'ichi KAKIZAKI[+], Kaoru INOUE[++], Kouji OHATA[++], PECE MONTENEGRO Manuel Jesus[+], Toshiya MATSUMOTO[+], Seiichi UCHINAMI[+]
[+]Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, [++]Yasukawa Joho System K.K.

シミュレータで使用するクラスとして，3種類のデバイスオブジェクトから派生させて，SimCartrack, SimCrane, SimConvairの3種類のデバイスオブジェクトを定義している．これらのクラスでは，外部装置を制御するメンバ関数を上書きにより無効化するとともに，デバイスオブジェクトの状態変化を引き起こすメンバ関数にウィンドウへのアニメーションなどを行う処理を付加し，各デバイスオブジェクトをシミュレータに適するように再構成している．

実際の自動倉庫で各外部装置が並列に動くように，シミュレータでも各デバイスオブジェクトを並列に動かす必要がある．各デバイスオブジェクトは，実システム上ではデバイスドライバあるいはタスクとして実現されるために並列に動作することができるが，シミュレータ上では，すべてのデバイスオブジェクトが一つのUNIXプロセス上で実行されるようにしているために並列に動作することができない．このため，一定時間ごとに各デバイスオブジェクトに制御を移し，疑似的に並行動作できるようにしている．この様な処理に必要な機能を持つ抽象クラスとしてSimulatorを定義し，シミュレータ用の各クラスに継承させている．なお，Simulatorクラスには，アニメーション用の基本的な機能も組み込んでいる．

シミュレータの実現だけを考えた場合には，DeviceクラスにSimulatorクラスを継承させるのが単純であり望ましいが，この様にすると，実際のシステムで用いられるデバイスオブジェクトにシミュレーション用のオーバーヘッドが組み込まれてしまうという問題があるために，図2のようなクラス階層となっている．

3．2　シミュレータの実現手法

自動倉庫のシミュレータは，X-WindowとX-Toolkitを利用して実現している．Simulatorクラスの継承によって実現される疑似並行動作機能は，X-Toolkitに用意されているバックグラウンドワークプロシージャを利用することにより実現している．バックグラウンドワークプロシージャとして定義している関数では，各デバイスオブジェクトの並列処理用のメンバ関数を定期的に呼び出すようにしている．

4．シミュレータの応用

4．1　テストとデバッグ

システムのテスト，デバッグ時には，実際に使用するデバイスユニットとシミュレータを置き換えることによって，外部装置がない状態で応用プログラムの機能検査を行うことができる．このため，外部装置を開発現場に持ち込んだり，外部装置の設置場所に出向いてテストやデバッグを行う必要がなくなり，非効率的な作業を軽減できる．

FA分野における外部装置は可動部分が多いため，本報告で示したようにグラフィックス表示できるようなシミュレータを作成すれば，技術者が視覚的な情報を入手して大局的な監視を行いながら，効率良くシステムのデバッグと調整を行うことができる．シミュレータは，応用プログラムから受けた指示を実行し，その結果を表示するだけでなく，その逆の機能も必要である．シミュレータに，システムの設置環境から外部装置に与えられる事象を発生させる機能を付加することにより，予測される様々な事象系列に対して，システムがどのように反応するかということをテストすることができる．さらに，シミュレータに外部装置の定格を逸脱したり，定義されていない機能の実行を要求する指示が与えられた場合には，指示の内容とその指示が与えられた時点での外部装置の状態を報告すると共に記録する機能を付けると効果的であると考えられる．

4．2　プロトタイピング

シミュレータの利用は，開発されたプログラムのテストやデバッグなどの，システム開発の後半の工程だけではなく，プロトタイピングや外部装置間のスケジューリングなどのアルゴリズムの評価などの，システム開発の早い時期でも有効に利用することができる．これにより，技術的な実現可能性の評価だけではなく，プロトタイプを参考にして発注者との仕様の確認を早期に行えるという利点がある．

5．まとめ

オブジェクト指向に基づく仮想化外部装置を利用した，自動倉庫のシミュレーションシステムの実現手法に関して示した．

シミュレーションシステムの開発には，自動倉庫の開発経験はあるが，オブジェクト指向による開発経験のない技術者に作業を行ってもらった．この結果，システムの分析，設計，プログラミングの各段階で，オブジェクト指向の考え方に慣れるのに時間がかかるものの，いったん理解した後は従来よりも効率良く開発できることが明らかになった．また，シミュレータも，オブジェクト指向の継承を用いることにより，実際に使用するデバイスオブジェクトを基本として，比較的容易に開発できることが明らかになった．

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).