

C++による仕様記述について

4S-5

南摩英明, 浜野博, 有沢誠

山梨大学

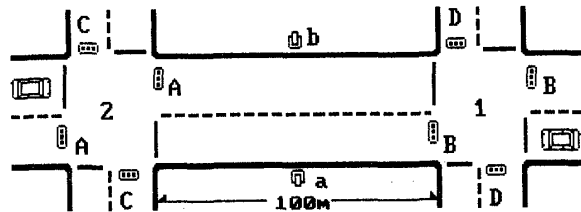
1. はじめに

本稿では, オブジェクト指向言語C++により2つの例題の仕様を記述し, 他の手法と比較することにより, 仕様記述の可能性と限界について考える.

2. 交通信号の実時間制御システム

(0) 問題

図のような道路を考える. 2ヶ所の十字路にA, B, C, Dの交通信号機がある. 道路はすべて両方向1車線で, 交差点の右折も許す. 右折信号はない. 信号の切り替えは, 赤-緑-黄-赤のサイクルとする. 1秒の整数倍の時間ごとに切り替える. 緑の継続時間は12秒-27秒の範囲で可変, 黄は3秒で固定, 赤は15秒-30秒の範囲で可変. AとB, CとDはそれぞれどちらか一方は必ず赤. お互いに赤であることは禁止. 中央の道路は, 両方向についてそれぞれ, その混雑度情報を, 区間に入っている車両台数で表し, 任意の時刻に十分早く計量できる. 車がどちら方向もふたつの信号をスムーズに通過できるようにタイミングをとる. 制御システムからA-Dへ指令した信号切り替えは, 十分速く伝達実行できる. システム誤動作に関するフェイルセーフ処理等は省略. 不十分な点は, 適宜補ってよい.



(1) オブジェクトに分ける

まず, この要求書からオブジェクトの候補をあげる. 車, 道路, 信号機, 制御システム

である. 車に関しては, シミュレーションの段階で初めて必要とし, しかも, 乱数で発生させればよいことである. 道路は, なにもすることがない. したがって, 信号と制御システムについて, 内部変数や機能を考える.

(a)信号クラス: 内部変数として, 各信号の色の継続時間と現在の色をもつ. 機能として, 信号の色変え, 継続時間変え, をとりあげる.

(b)制御システム: 内部変数として, 一定時間の両方向の車両台数をいれておくカウンタと, 一定時間の両方向の車と車の時間差をいれておく配列をもつ. 機能として, 車量台数計量, 時間差計量, 混雑度計量をとりあげる. 車と車の時間差は要求書にはでてこないが, 実際に混雑度を計る場合に, 車両台数だけでは混んでいるかどうかの判断をする事はできない. 例えば, とても混んでいて車が動かない場合は, 車両台数は少ない. そこで, 車と車の時間差も計り, 車両台数との相互関係で混雑度を計る.

(2) 機能の詳細

各機能の詳しい中身, すなわち, 何を引数にもち, どのように動き, 何を返すかを記述する. ここでは, 「void 信号機::色変え(int& s)」のように記述する. これは, クラス信号機のメンバ関数「色変え」はint&の引数をもち, 返すものはないことを意味する.

```
void 信号機::色変え(int& s)
{sは信号が代わってからの経過秒を示す. この値が現在の色の継続時間に等しいかそれ以上だったら, 次の色に変える. さもなくば, 経過秒に1加える. 参照を引数としているので, ここでの実行が, 本体に影響する.}
void 信号機::時間変え(int x, int y)
{xの値に12を加えたものを緑の継続時間, yに
```

15を加えたものを赤の継続時間とする。x,yはそれぞれ混雑度の値で、0-15の値をとる}

```
void システム::台数計量(int x)
{ xが1のときは、カウンタ1に1加え、2のときは、カウンタ2に1加える。カウンタ1は右方向の車両台数、カウンタ2は左方向である。 }
```

```
void システム::時間差計量(int x,int y)
{ xは上と同様、yの値を配列のカウンタ-1番目にいれる。yは時間差の値、配列は0番目から始まるためカウンタの値より1小さい }
```

```
void システム::混雑度計量(int& x1,int& x2,int& y1,int& y2)
{ x1,x2は両方向の緑信号に加える値で、y1,y2は赤信号に加える値。加える値を決めるのに次の表を考える。
```

通過台数	時間差の平均	混雑の状態
多い	小さい	混んでいる
多い	大きい	ありえない
少ない	小さい	混みすぎ
少ない	大きい	すいている

2番目のありえないを除く3つのそれぞれを5段階に分けると、1-15の値ができ、それ以外の0を加えて0-15の値ができる。その値を両方向もとめ、x1,x2に代入する。y1,y2は15からその値を引き代入する。両方向の同期をとるために、その値の差をすいている方の赤信号の時間に加える。多い、少ないの目安は、検討中である。 }

3. 在庫管理問題 [2]

(1) オブジェクトに分ける

まず、要求書からオブジェクトの候補と機能についてあげる。

(a) 受付係：内部変数として、在庫リスト、在庫不足リストをもつ。機能として以下のものがある。出庫依頼受け、在庫確認、在庫があれば、倉庫係へ出庫指示、在庫不足のときは、依頼者へ電話連絡し、在庫不足リストに記入、倉庫係へ空予定コンテナ知らせ、積荷票到着後、在庫不足リストを確認し、当該品の必要量があったら、不足品の出庫指示。

(b) 倉庫係：内部変数として、コンテナをもつ。機能として、コンテナ受け取り、受付係に積荷票送付、出庫指示により、内蔵品出庫、がある。

(c) 依頼者：内部変数はない。機能としては、

出庫依頼、がある。

(2) 機能の詳細

```
void 受付係::依頼受付(出庫依頼票 x)
{ 在庫を確認し、もしあれば、在庫リストからxを削除、出庫指示書を作成、出庫指示、なければ、依頼者へ電話、在庫不足リストを更新 }
```

```
void 受付係::出庫指示(出庫指示書 x, 倉庫係 y)
{ y.発送(x); }
```

```
void 受付係::空予定の指示(倉庫係 x)
{ 在庫ファイルから空予定のコンテナを調べ、x.空予定(コンテナリスト); }
```

```
void 受付係::受取(積荷票 x)
{ もし、在庫不足リストを確認し、あれば、在庫不足リストからxを削除、出庫指示書を作成、出庫指示、なければ、在庫リストを更新 }
```

```
void 倉庫係::発送(出庫指示書 x)
{ コンテナからxの内容を削除 }
```

```
void 倉庫係::受ける(コンテナ x)
{ コンテナへxを追加 }
```

```
void 倉庫係::渡す(受付係 x, 積荷票 y)
{ x.受取(y); }
```

```
void 依頼者::依頼(受付係 x)
{ 出庫依頼表作成,x.依頼受付(出庫依頼表); }
```

その他に、リストの検索、更新、削除をする、関数も必要である。ここで、y.発送(x)というのは、クラス倉庫の変数yが引数xで発送する意味である。

4. おわりに

C++で仕様を記述するときの問題として、メッセージのやりとりを関数呼び出しで行い、クラスは型であり、その型の変数を用意しなければならないために、きれいに記述することが難しいことがわかった。

5. 参考文献

- [1] B. Stroustrup: The C++ programming Language, 1986
(訳、斎藤信男、トッパン、1988)
- [2] 山崎利治：設計方法解説のための共通例題、プログラム設計技法の実用化と発展シンポジウム、情報処理学会、1984