

# ソフトウェア開発高度化に関する研究

長谷川 辰 雄\*、藤 澤 充\*\*

## A Simplification of Making Computer Software Using Simple Diagram and Line

HASEGAWA Tatsuo\*, FUJISAWA Mitsuru\*\*

The computer programmers have to make great efforts to develop software. Hardware and operating systems are complicated every year. Computer language is only written in characters. It caused programmers to confuse with making programs. We apply a simple diagram to programming. A simple diagram made it easier to develop programs. This paper describes diagram programming.

Keyword : Software, Program-Parts, Diagram

### 1 緒 言

コンピュータを利用して物体の構造解析や測定を行おうとした場合、ソフトウェアを自作して実験することはごく希になってきている。まずは市販のソフトウェアやハードウェアを購入して行うのが一般的である。市販のソフトウェアは、ハードウェアの高速化、大容量化に伴って機能が複雑化し、何人もの人が分担作業で作成している。

複雑な計算や処理をおこなう特殊な市販ソフトウェアは、多機能ではあるが、ユーザーはその機能全てを使用する訳ではない。ユーザーが本当に行いたい解析や測定は、2機能か3機能だけで十分な場合がほとんどである。少数の機能であれば、個人で目的に合致したソフトウェアを作るほうが、コスト的にも良いと思われる。しかし、ソフトウェアを1から作るとなると、プログラミングの知識が必要となり、その習得に多大の時間を要する。

著者らは、プログラミングの知識習得を最小限にし、容易にソフトウェアを作成可能とすることを目的とした開発システムを研究中である。本稿ではその進捗状況を報告する。

### 2 開発システム

#### 2-1 開発システムの考え方

ソフトウェアを容易に作成する考え方は、以前から多数研究されているが、最近注目されはじめた考え方に「オブジェクト指向」がある。従来の考え方は「手続き型」と呼ばれる手法であり、データの処理があら

かじめ決められた順番で行われる。これは、ソフトウェアをある機能単位に分割して設計する「構造化手法」と共に、現在のソフトウェア開発の主流となっている。この場合、データ部と処理部が明確に分かれており、データ部がいくつかの処理部によって何らかの処理がなされ、決められた順番に従って処理が行われる。

一方、「オブジェクト指向」は従来方法とは異なり、データ部と処理部が一緒になった構造をしている。オブジェクトは日本語で「もの」を指すが、あるオブジェクトに何らかのメッセージを送ることで、ある振る舞いを自ら行う「もの」のことをいう。例えば、手帳というオブジェクトにたいして「表示せよ」というメッ

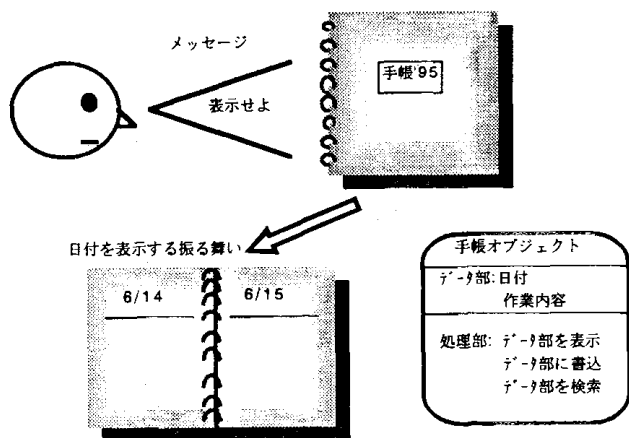


図1 オブジェクト指向の概念

現在 \* 岩手県工業技術センター 電子機械部 岩手県盛岡市飯岡新田 3-35-2  
 \*\* 岩手県工業技術センター 企画情報部 岩手県盛岡市飯岡新田 3-35-2

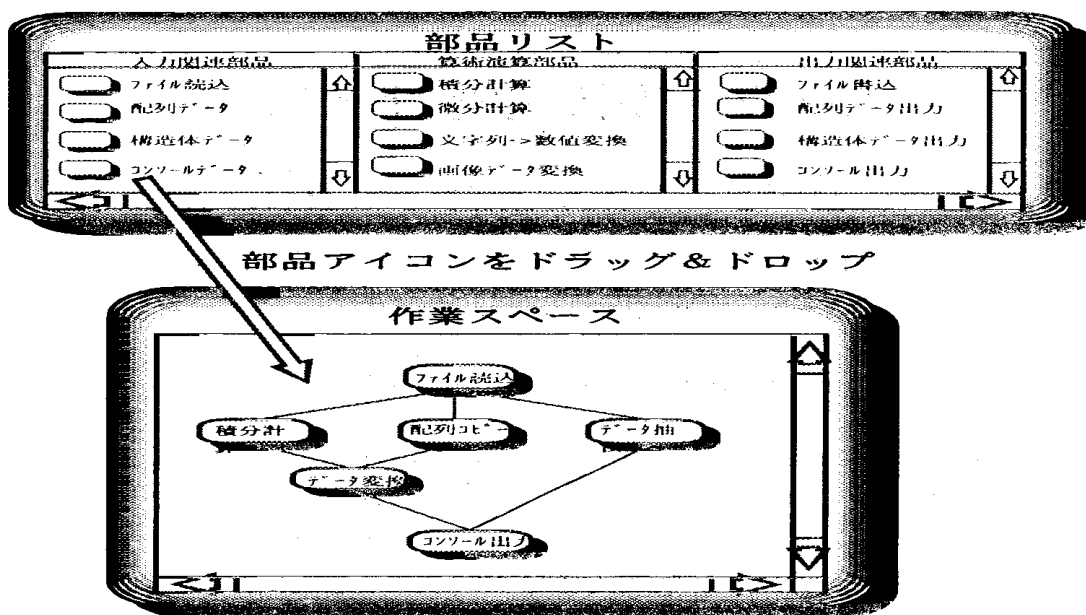


図2 ソフトウェア開発構築ツール概念図

メッセージを送る。すると、手帳は今日の日付を表示する。この場合、手帳は「日付」というデータ部と「日付を表示する」という処理部を一緒に持っているオブジェクトであり、メッセージ「表示せよ」により「日付を表示する」という振る舞いをする。概念を図1に示す。

オブジェクト指向では、従来プログラミングよりもプログラムの部品化及び再利用が容易であると言われている。しかし、オブジェクト指向言語でプログラミングすれば、容易にプログラムが可能であるという訳ではない。いくつかの問題点を以下に記述する。

- 問題点
- ・ プログラムの部品化はしやすいが、1からプログラムを作成する場合、プログラム部品が存在しない。
  - ・ オブジェクト指向言語を習得するのに時間がかかる。
  - ・ オブジェクト指向の1つの概念である「継承」は、部品化に役立つが、あとからプログラム・コードを分析しようとする場合に非常に分析しにくい。
  - ・ オブジェクト指向プログラミングが分析しにくいということは、その部品の再利用も困難である。

本研究は、上記の問題を軽減するようなソフトウェア構築ツールを作成することを目的とする。

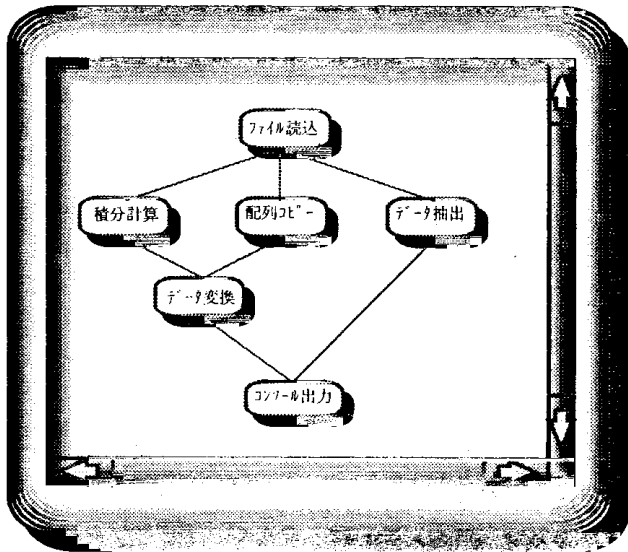
## 2-2 ソフトウェア構築ツールの仕様

ソフトウェア構築ツールの仕様を以下に記述する。

- ・ C言語を対象とする。
- ・ 基本となるソフトウェア部品をあらかじめ搭載する。

- ・ ソフトウェア部品は新規登録、修正、削除を可能とする。
- ・ ソフトウェア部品はわかりやすい図的表現にする。
- ・ ソフトウェアの作成は、図で表した部品を線でつなぎ合わせることで行う。
- ・ 図と線で表したソフトウェアから、C言語ソースコードを生成する。
- ・ 生成されたC言語コードのコンパイル・リンクは行わない。
- ・ 初級C言語習得者以上を対象とする。

ソフトウェア開発構築ツールの概念図を図2に示す。ソフトウェア開発者(以下ユーザー)は、コンピュータ画面上に表示されたソフトウェア部品リストから、求める部品をマウスにより作業スペースへドラッグ&ドロップする。ユーザーは、部品と部品を線でつなぎ合わせることでプログラムの処理の流れを記述する。図で表現されたプログラムは自動的にC言語ソースコードに展開される(図3 C言語コード生成)。このシステムでは、C言語のコンパイル・リンクは行わない。理由は、企業における開発環境(コンパイル環境)がそれぞれ異なっているためである。そのため、本ツールのユーザーはC言語コンパイラを使用して実行可能アプリケーションを生成しなければならない。C言語に限定した理由は、コンピュータ・プログラム言語として一般的であり普及していることである。オブジェクト指向言語「C++」は、先に述べた「習得時間がかかる」、「プログラムの分析が困難」などの理由により一般的ではないと判断した。



↓ C言語コードを生成

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    file_open();          // ファイル読込
    integral();          // 積分
    array_copy();        // 配列コピー
    convert_ascii();     // データ変換
    get_valid_data();    // データ抽出
    display();           // 表示
}
/*****
/* Spec: file open .....
/* input: none
/* output: file pointer
*****/
void file_open()
{
    FILE *fp;
    fp=fopen("xxxxx.xxx",w);
}
```

図3 C言語コード生成

### 3 問題点と考察

現在は本システムの開発中であり、仕様に基づきプログラミングを行っている段階である。複雑化していくプログラムを、より簡単にわかりやすく記述したいという要求は当然であり、過去には、様々なプログラムの視覚化の研究がなされてきた。しかし、従来のテキストベースのプログラムを視覚化するには様々な課題が存在する。現在、当システムの開発において、直面している問題点は次の通りである。

- (1) テキスト情報を全て図で表現しようとするとう煩雑になり、逆にわかりにくくなる。
- (2) テキスト情報の一部をで表現した場合、図とテキスト情報とのリンクが難しい。
- (3) 各社のコンピュータ機種 (OS) の違いより、共通のGUI (Graphical User Interface) を作成することは困難である。

(4) ソフトウェア部品を登録する際、図的表現の制約により従来のソースコードをそのまま登録することはできない。

本システムでは上記問題点を解決するために、実験的にC言語の関数を部品として提案している。C言語関数はブラックボックス的に表現される。また、入出力が明確なため、構造化プログラム技法 (図4 構造図例) では1つの箱で記述される。この箱に着目し、構造化プログラム技法で記述された箱を、視覚化する方法として利用したものが本システムのイメージである。

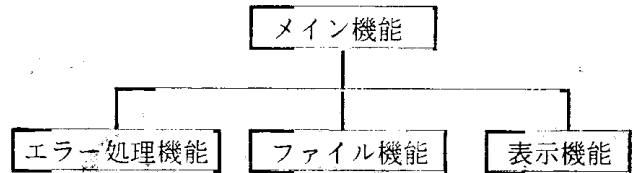


図4 構造図例

本システムの箱はC言語関数を表し、実際のソースコードは部品データベースとして別の場所に保管される。部品データベースから作業スペース上へドラッグ&ドロップされた箱は、単なるデータベースのインデックスとして機能する。箱と箱を接続する実線は、データ及び機能の流れ (順番) を表す。具体的な例を図5に示す。

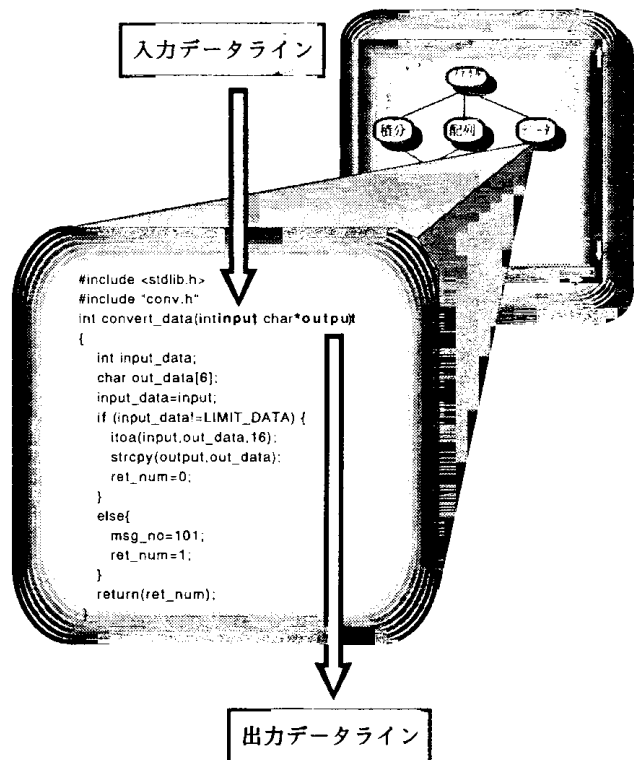


図5 データの流れ

一般的にソフトウェアは目に見えないといわれている。その理由は、プログラムはアルゴリズムの結論のみ

を数式や文字で表現し、その結論にいたった理由、証明は影に隠れているからである。しかし、実際には結論にいたった理由などはプログラムで記述する必要はない。

このような考えで、本システムは、プログラムの内容そのものを図で表現するよりも、単なる象徴としての図を使用しようと試みた。しかし、単なる象徴としての図も画面上にいくつも存在すると煩雑で逆に分かりにくくなる。

そこで、いくつかの箱をまとめて、1つの箱にするグループ化(図6 グループ化)を行って煩雑さを避けようと計画している。

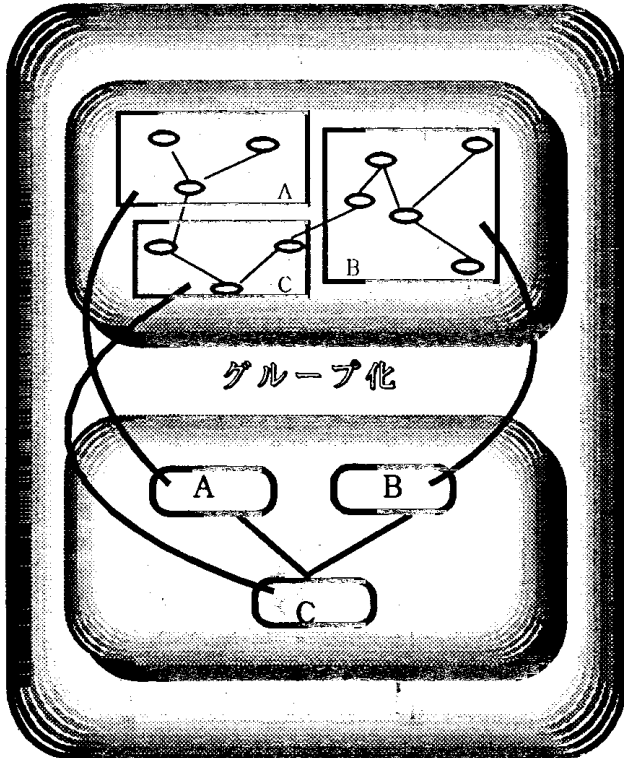


図6 グループ化

#### 4 結 言

本システムでは、部品データベースが重要な機能を果たすが、ここに保管される部品をパターン別に選別し登録することが非常に重要な意味を持つ。現在の進捗状況は、両面とドラッグ&ドロップ処理の設計を行っており、全体の20%程度である。本製品の中核となる部品データベースは未着手であり、今後早急に作成を進める予定である。

#### 5 要 約

ソフトウェア開発の効率化を目指し、コンピュータ・プログラムを部品としてデータベースへ登録し、図的表現によって容易にソフトウェアを構築する「ソフトウェア構築支援ツール」を開発中である。プログラムの中身を図的に表現するのではなく、「ファイル名」のようなインデックス的に図を利用する手法を用いた。

キーワード：ソフトウェア プログラム部品 図的表現

#### 参考文献

- (1) 竹内彰：インタラクティブシステムとソフトウェアI, 近代科学社 (1994)
- (2) 松田純一：画像データベース構築・活用技法, オーム社 (1991)
- (3) 河西朝雄：C, ナツメ社 (1994)
- (4) Kurt J. Schmucker：オブジェクト指向プログラミング 上下巻, 日本ソフトバンク (1990)
- (5) 原田実：CASEのすべて, オーム社 (1991)
- (6) 柴田望洋：CプログラマのためのC入門, ソフトバンク (1993)
- (7) S.B. リップマン, 富士ゼロックス情報システム訳：C プライマー, トッパン (1993)