

パッケージ設計専用CADの開発*

長谷川 辰雄**、中村 吉信***、上野 育子***、藤澤 充**

包装用パッケージ(紙箱)の製作は、中にいれる商品の形状や重量、紙材質、外見の良さ、色彩など、様々な項目を検討して行われる。その工程は、はじめにパッケージの意図するデザインを検討し形式を決定する。その形式に基づいてドラフター製図を行う。この製図の段階で、組み立てに必要な切れ込み、糊代幅、色づけの際のベタ幅など、細部の寸法計算を行わなければならない。この寸法計算は、紙厚や紙材質によって異なってくるため、熟練者でも30分から1時間の時間を要する。本研究では、時間を要する製図工程の省力化を図るため、パソコンによる寸法計算で図面作成を行うソフトウェアを開発した。

キーワード: パッケージ、CAD、ソフトウェア

A Development of CAD Software for drawing Package Design

HASEGAWA Tatsuo, NAKAMURA Yoshinobu, UWANO Ikuko
and FUJISAWA Mitsuru

There are many processes of making packaging box. The design is decided based on a purpose of package, type, weight, material of paper, looking and color. The drawing process is required many time because of designers have to make detail drawings about measuring the size, calculating area size of a tab for sticking and colorizing. This process needs from 30min. to 60min. for an expert. This paper reports a development of CAD Software that can make package drawing with easy operations.

key words: Package, CAD, Software

1 緒 言

包装用パッケージ(紙箱)のデザイン及び製作は多くの工程を経て行われており、パソコン利用による製作作業の省力化が試みられている。パッケージには、商品の可搬性・安全性を保つほかに、商品アピールなどの宣伝効果が期待されている。既製品パッケージは、形状や材質が固定されているため、特殊な形状、安全性をより高めたいなどの柔軟な要求を満たすことができない。そのため、新たなパッケージ製作が必要となる。単純形状と思われがちなパッケージであるが、その製作には時間を要する。この作業工程を次に述べる。「①パッケージに入る商品の寸法を測り、パッケージの形式を決定する。②寸法計算及びドラフターで製図し、型紙を作成する。③型紙を基に紙を切り、組み立ててパッケージの試作をする。④発注先からの要求があった場合に、イラストレータ (Adobe Illustrator) でCAD展開図面を作成する」の工程を経て行われている。これらの作成には、10分から1時間を要する。さらにイラストレータで展開図面を書く場合は、30分から1時間ほど要する。

また、パッケージの形状は20種類に分類され、紙材の

種類や寸法と組み合わせると数百種類になり、その選択が製図時間に大きく影響を及ぼしている。作業工程の中で多くの時間を費やすのが、②の寸法計算及びドラフターでの製図、④の展開図面の作成であり、これらを自動化することで大幅な時間短縮となる。本報では、パッケージ種類の選択をマウス選択で簡素化し、寸法入力のみでパッケージのデザイン図面を自動生成する新規CADについて述べる。

2 開発方法

2-1 開発概要

本CADの原型は1987年に(株)でんにより開発され、当時のプログラムはMS-DOS上のBASICで作成された。しかし、現在は全く異なったOS環境であるため、開発するプログラムはWindows95/98上でVisual C++⁽¹⁾⁽²⁾⁽³⁾⁽⁴⁾⁽⁸⁾及びAccessデータベース⁽⁵⁾⁽⁷⁾を使って、オブジェクト指向型開発⁽⁵⁾を行った。オブジェクト指向型開発は、膨大・複雑化するソフトウェア開発の効率化、信頼性向上を目的とした設計手法であり、これを本研究の開発プログラムに適用した。これは、データベースを管理するためのAccessと、

*技術バイオニア養成事業

**電子機械部

*** (株)でん

計算プログラムにはVisual C++というように、別々に開発することで作業分担ができ、開発効率が向上するメリットがある。Accessを用いてデータベース化することは、デザイン(寸法)変更が容易であり、型紙のデジタルアーカイブ(デジタル保管)や、再利用性向上が図れる。開発CADは、(株)でんの経営システムとコンピュータ・ネットワークで連携管理できるようにデータベース設計を行った。本システムの基本となる設計方針を次に示す。

- ① 画面の遷移階層は5階層以内で設計する。
- ② データの入力形態は、可能な限り選択式で設計する。
- ③ 寸法計算はオブジェクト指向型で設計する。
- ④ 図面データはファイルとして保存する。
- ⑤ CADの入力データはデータベースとして構築する。

これらの基本設計を元に詳細なフローチャートを作成し開発を行った。図1に詳細フローチャートを示すとともに、その内容を以下に述べる。

形式を選ぶ

パッケージの基本となる形式を選択する。形式は直接選択することも出来るが、縮小されたパッケージの展開図面を見ながら選択することも可能である。

紙の種類を選択

パッケージに使う紙の種類を選択する。(紙の厚さの違いにより、計算方法も変わってくる。)

寸法基準の指定

パッケージの寸法の基準となる指定方法を選択する。内

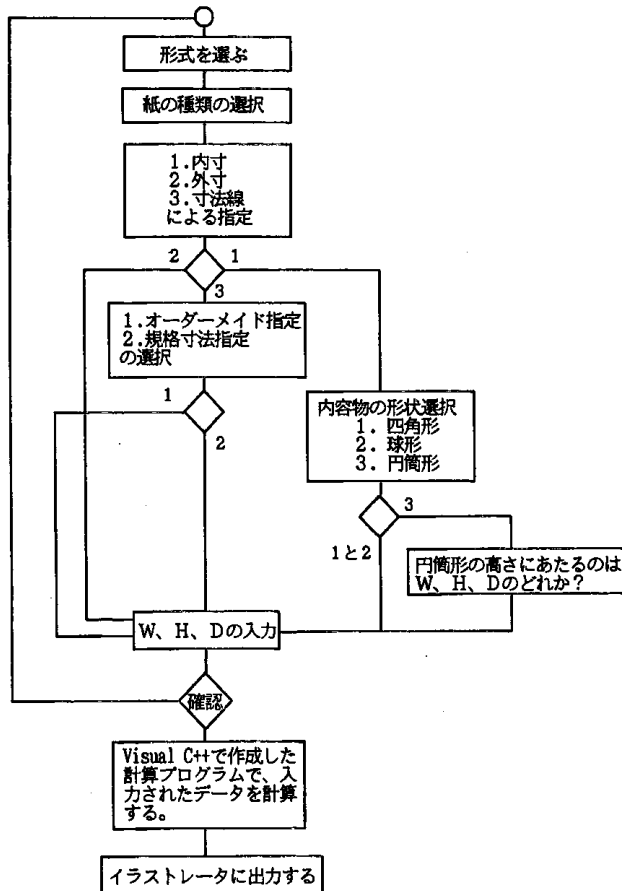


図1 フローチャート

寸の場合は、パッケージの内側の寸法を基準にし、外寸はパッケージの外側を基準にしている。寸法線による指定の場合は、2つの選択肢があり、W、H、D以外の細かい寸法を入力するのがオーダーメイド指定で、W、H、Dだけを入力するのが規格寸法指定である。

図面化データの計算

各形式の選択や寸法の入力が終了後、Visual C++で作成されたプログラムがデータを読み込んで図面化データの数値計算を行う。

ファイル保存

イラストレータ形式の展開図面がファイルとして出力される。

2-2 開発手法

代表的なプログラム開発手法に、構造化開発とスパイラル型開発の2種類が挙げられる。構造化開発の特徴は、プログラムを細分化しグループ分けすることで分散開発を可能とした。これによって、複数人による大規模開発が可能となる。現在でも主流の開発手法であるが問題もある。例えば、あるデータ形式が変更になった場合、それを処理する全ての関連モジュールを見つけだし修正しなければならない。修正したモジュールが、他のデータ処理に対して悪影響となる場合がある。スパイラル型開発は、最初に核となる小さなプロトタイププログラムを作成し、それに対して少しずつプログラム追加しながら開発する手法である。これは、開発内容をすべて把握可能な個人レベルの小規模なプログラム開発に向いている。開発の容易性の反面、開発者以外にはプログラム内容がわからず、複数人で改良や拡張することができない傾向にある。オブジェクト指向型開発は、クラス階層化やインヘリタンス(継承)という概念を用いることで、モジュールの追加・修正の容易性を追求した手法である。本研究では、この手法に基づき、寸法計算プログラムをクラス階層化し、メソッドをプログラミングした。

3 結果

開発したCADプログラムを起動すると、図2に示すデータ入力画面が表示される。ここで、パッケージの形式や蓋の種類に関するデータを入力する。この入力データは同時にAccessデータベースへ保管される。パッケージ形式の選択は、名前を選択する方法と画像化した展開図を視覚的に選択する方法の2つの選択方法を可能とした。視覚的に選択する方法では、図3の細分化されたパッケージ種類を選択した後、図4に示す画像リストから選択が可能である。ここまで入力したデータを元に図5に示すレイアウト確認画面が表示される。この画面で紙の種類や折りかたの方法、外寸・内寸の寸法設定を行う。これらの作業が完了後、図面化のための計算プログラムを立ち上げ、入力されたデータを計算することで、イラストレータの展開図面が図6のように出力される。入力作業から展開図面が出力するまでの時間は約1分程度であったが、従来の設計時間である30分~1時間を大幅に短縮することができた。

4 今後の課題

試作段階であるが、形式と寸法を与えるだけでCADデータを求めることができた。作業時間も従来の30分の1から60分の1と大幅に短縮することができ、目標を達成することができた。本開発におけるプログラムのテストは、1~2人のプログラム開発者で行っているため、実用化のための実用化試験を実施する必要がある。このため、作業現場での複数人によるテストが必須となる。また、本ソフトウェアの本格稼働には、形式種類の追加、カッタープロッター出力、信頼性向上テスト、営業データとの連携など多くの機能追加が必要となる。本

格稼働のためのプログラム全体構造を図7に示す。核となる処理プログラムをオブジェクト指向型のC++言語で開発し、入力データはAccessデータベースへ構築され経営システムで管理される。図面をカッタープロッターで出力し、図面の切り取り作業の自動化を目指す。本ソフトウェア開発の手法に、オブジェクト指向型開発を採用したが、C言語やBASIC言語とプログラムの流れが異なっているため、データ構造のクラス化や処理フロ

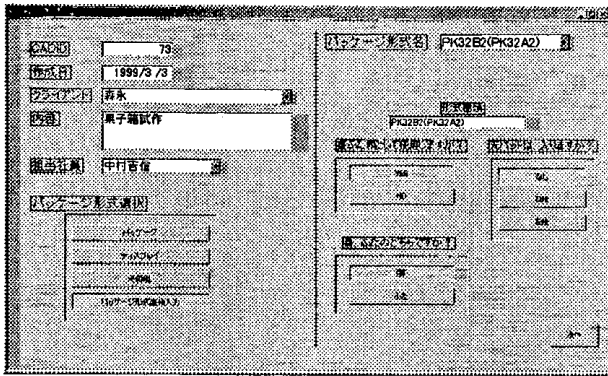


図2 初期データ入力画面

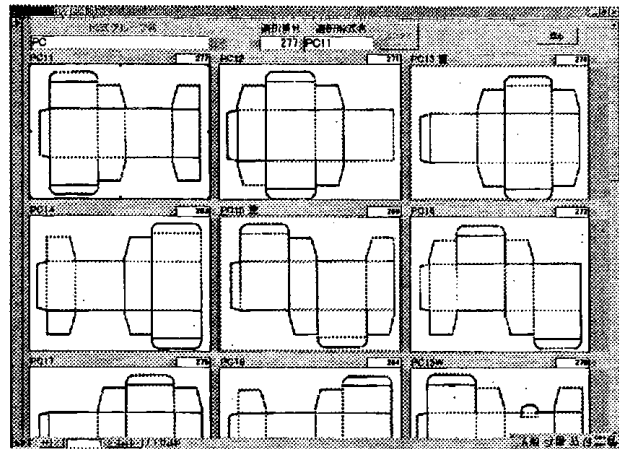


図4 パッケージ種類の画像リスト



図3 細分化されたパッケージ種類

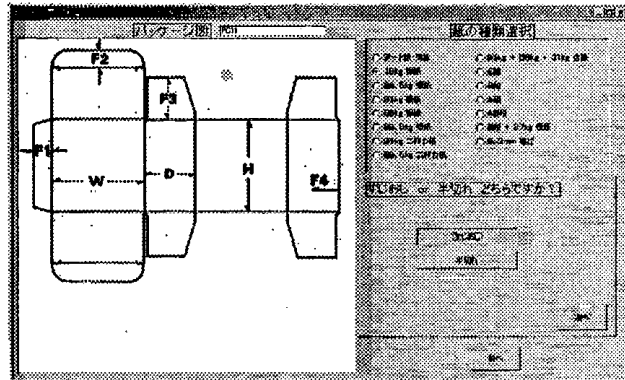


図5 レイアウト確認画面

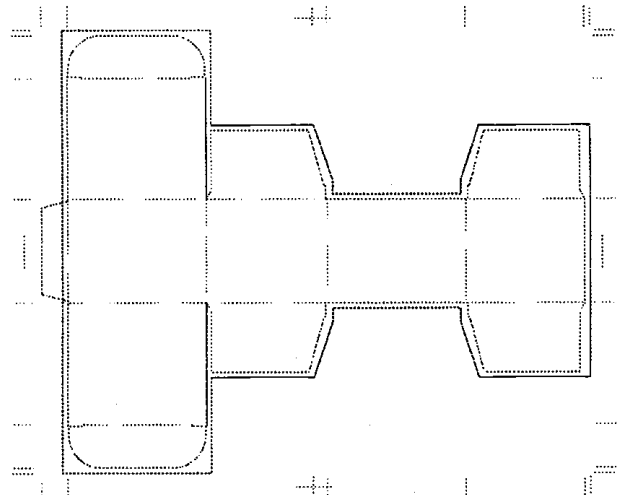


図6 イラストレータ展開図面 (実行結果)

一の作成方法に時間を要した。この結果として開発効率は従来開発よりも低下したが、オブジェクト指向型開発技術の修得の時間を含んでいるため単純には比較できない。開発効率や信頼性の向上、保守時間の低減など、実際に効果が現れるのは今後のプログラム開発からであると期待される。

文 献

- 1) 桜田幸嗣, 田口景介: Visual C++ 5.0 プログラミング入門, アスキー, 1998
- 2) 横井与次郎: Visual C++ 5.0 パワープログラミング, ソフトバンク, 1996
- 3) David J. Kruglinski: Inside Visual C++ Version 5, アスキー, 1998
- 4) 林晴比古: 新 Visual C++ 5.0 入門, ソフトバンク, 1998
- 5) 石塚圭樹, 横手靖彦: オブジェクト指向プログラミング, アスキー, 1993

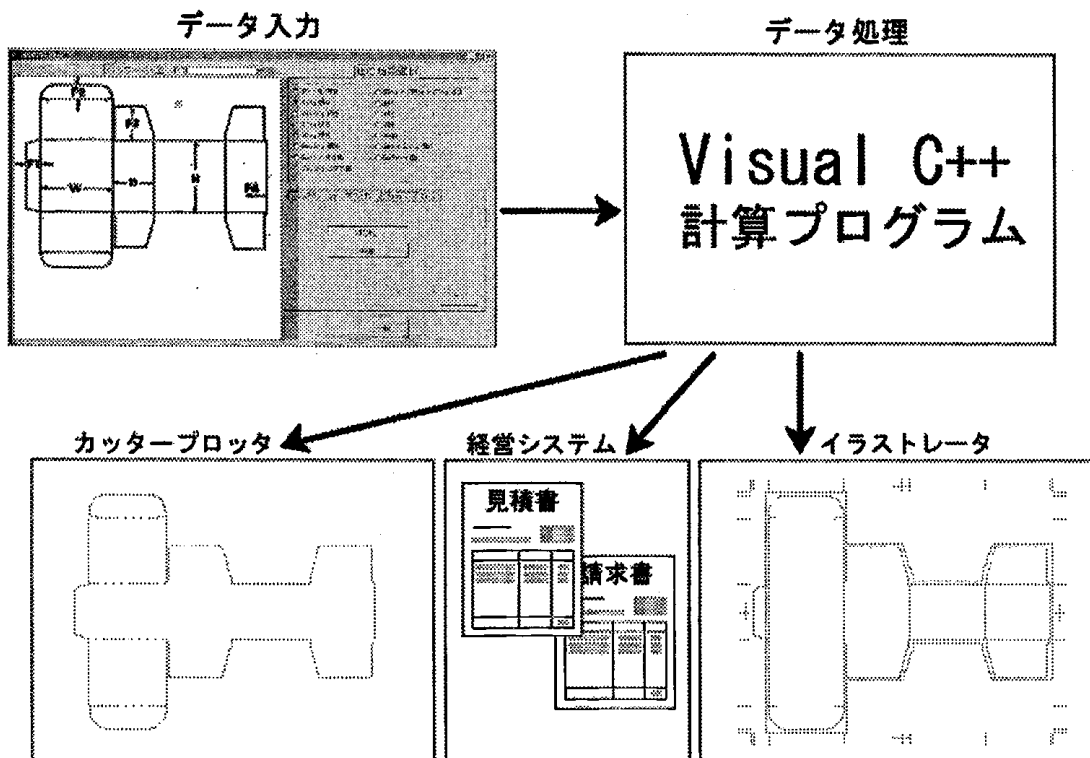


図7 プログラム全体構造図