

外部設計

この資料は大阪情報コンピュータ専門学校における
学習用であり，将来この資料の全体または一部につ
き，変更の必要がある場合があります．

外部設計

目次

| | |
|---------------------------------|----|
| 学習目標・はじめに | 2 |
| 1.1 外部設計とは | 3 |
| 1.2 要求仕様の確認..... | 3 |
| 1.2.1 概要 | 3 |
| 1.2.2 分析技法 | 4 |
| 1.3 サブシステムの定義と展開 | 11 |
| 1.3.1 サブシステムへの分割 | 11 |
| 1.3.2 サブシステムの展開と新物理モデルの作成 | 11 |
| 1.3.3 ソフトウェアパッケージの組込み | 12 |
| 1.3.4 性能・規模の見積り | 12 |
| 1.4 画面設計..... | 13 |
| 1.4.1 画面設計とは..... | 13 |
| 1.4.2 画面設計の手順と作業内容 | 14 |
| 1.5 報告書設計 | 18 |
| 1.5.1 報告書の設計とは..... | 18 |
| 1.5.2 報告書設計の手順と作業内容 | 18 |
| 1.6 コード設計 | 21 |
| 1.6.1 コード設計とは | 21 |
| 1.6.2 コード設計上の留意点 | 22 |
| 1.6.3 コード設計の手順と作業内容 | 23 |
| 1.6.4 よく使われるコードの例 | 24 |
| 1.6.5 コードのチェック方法 | 26 |
| 1.7 論理データ設計..... | 27 |
| 1.7.1 論理データ設計とは | 27 |
| 1.7.2 論理データ設計の手順と作業内容 | 27 |
| 1.8 外部設計の作成..... | 30 |
| 1.8.1 外部設計書とは | 30 |
| 1.8.2 外部設計書の構成と内容..... | 30 |
| 1.9 理解度確認問題..... | 33 |

外部設計

学習目標

外部設計の主要な作業について学習する。外部設計はシステム設計の入口なので、システムの全体像を意識した設計が大切である。

プログラマは外部設計に直接携わるわけではないが、内部設計～プログラミングを担当するうえで、上流工程を担当する SE が設計した外部設計書に書かれている内容について理解できるようになることを目標として学習する。

1 外部設計とは

外部設計では、コンピュータシステムの内部にはかかわりなく、ユーザの要求・意見を反映した設計が行われることを確認する。

2 外部設計

「要求仕様の確認」から「外部設計書の作成」に至るまでの各フェーズの具体的な内容、手順、成果物(ドキュメント)について説明しているので、それぞれの概要をしっかりと把握する。

はじめに

外部設計は、ウォーターフォールモデルの開発工程でいうと、基本計画の次のフェーズに該当する。通常、このフェーズでは、システム開発に精通したシステムアナリストやアプリケーションエンジニアなどがユーザとの話合いのもとに、システムの分析や設計を行う。

外部設計の“外部”とは、「コンピュータ(システム)の外から見た」という意味である。「コンピュータ(システム)の外から見る」のはユーザであり、外部設計ではヒューマンインタフェースを中心にユーザの立場に立って設計が行われる。したがって、要求仕様の確認は非常に大切な作業の一つであり、ユーザとのトラブルもこの段階の確認不足に起因するところが多い。ユーザの要求を満足させることは、システム開発の最も重要な目的なので、基本計画フェーズで作成した要求仕様書の確認や、ユーザとの打合せは綿密に行わなければならない。

1. 1 外部設計とは

基本設計で、システム化のための調査・立案・要求定義などが行われる。外部設計では、それらの資料をもとにして、システムの機能やヒューマンインタフェースについての定義が行われる。ユーザの立場から見た設計が行われることになるので、ユーザと綿密に打合せをして、ユーザの意見を十分に反映した設計を行わなければならない。このときのユーザにとっては、コンピュータシステムの内部でどのようなシステムやプログラムが動いていても関係ない。「何を入力」すれば、「どのような出力」が「どのようなタイミング」で得られるかということが理解できれば、ユーザは滞りなく仕事を進めていくことができるからである。

ゆえに、担当者（アプリケーションエンジニアなど）は、ユーザにわかりやすいシステムの設計を心掛ける必要がある。

外部設計における作業手順は、次のとおりである。

- ① 要求仕様の確認 (DFD などによって求められたシステムの機能分析)
- ② サブシステムの定義と展開 (サブシステムの設計)
- ③ 画面設計・報告書設計 (画面および帳票の設計)
- ④ コード設計
- ⑤ 論理データ設計
- ⑥ 外部設計書の作成

また、外部設計で作成される成果物 (ドキュメント) は、次の二つである。

- ① 外部設計書
- ② 外部設計レビュー報告書

なお、このフェーズでは、運用前の最終段階のテストであるシステムテストや、システムの移行・運用などの計画も行われる。

1. 2 要求仕様の確認

要求仕様の確認は外部設計における最初の作業である。基本計画で作成した要求仕様書をベースに、さらに詳細にシステムの目的やユーザの要求などを確認する。

1. 2. 1 概要

要求仕様の確認とは、開発対象のシステムに求められる機能を分析する作業のことである。

〈作業の概要〉

- ① 基本計画で作成された要求仕様書を確認する。

- ② ユーザとの打合せを行い、システムに盛り込むべき機能やその他の要求を確認する（グループインタビューなどを行う）。その際、ユーザや基本計画担当者（システムアナリストなど）に必ず確認をとる。
- ③ 上記①・②をベースに、現在の業務をより詳細に分析し、システムに対するユーザの要求事項（ユーザ要件）をまとめる。
- ④ 基本計画で作成した現行論理モデルに上記事項を加味し、新業務フロー（新論理モデルという）を作成する。ここでは、ユーザの要求をもとにデータフローとその処理を整理しただけのモデルで、詳細な部分にはふれない。

1. 2. 2 分析手法

要求仕様の分析技法としては、DFD、HIPO、状態遷移図などがあるが、ここではDFDとHIPOについて説明する。

（1）DFD（Data Flow Diagram）

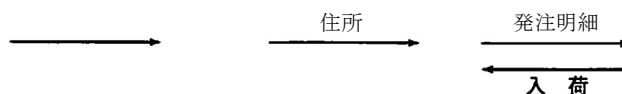
DFDは1970年代後半にアメリカで使用されるようになった技法で、業務の流れを図式表現する。業務の流れを表現する場合、使用されているデータがどこで発生し、どのように処理され、そしてどのように保管されているかに着目することから始まる。したがって、DFDは作成しやすく、ユーザにも理解しやすいため、最近では内部設計などの下流工程でも幅広く用いられている。

1）DFDの表記法

DFDで用いる記号は、次の4種類だけである。

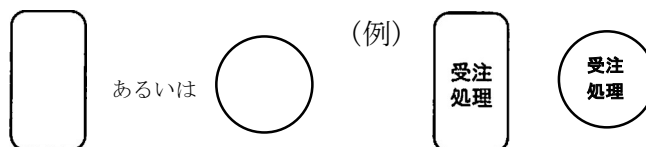
- ・データフロー（矢印）：情報の流れを示す。

（例）



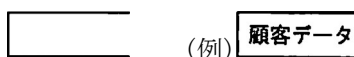
図表 1-1 データフロー

- ・処理（プロセスボックス）：データの加工や変換を表す。



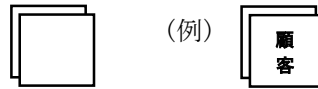
図表 1-2 処理（プロセスボックス）

- ・データストア：データの蓄積を示す。



図表 1-3 データストア

- ・外部:分析対象業務の範囲外にあり,データの発生源または行き先となる対象を示す.



図表 1-4 外部

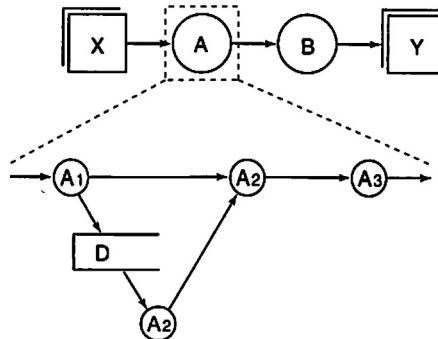
2) DFD の特徴

DFD の主な特徴を示すと,以下のとおりである.

- ・データの流れを中心に記述する.
- ・非常に簡単な表記法なので,ユーザにも理解しやすい.
- ・組織や入力媒体にとらわれずに,必要な業務の機能を抽出することができる.
- ・あらゆるレベルに対応したシステムモデルの記述が可能である.
- ・処理機能が分解されるため,修正しやすい.
- ・開発フェーズ間の接続が円滑で,しかもフェーズ化しやすい.

3) トップダウン展開

DFD は,処理機能を順次下位レベルの DFD に分解して,トップダウン展開で詳細化することができる.これは複雑な機能を取り扱う場合の解決策として効果的である.



図表 1-5 トップダウン展開

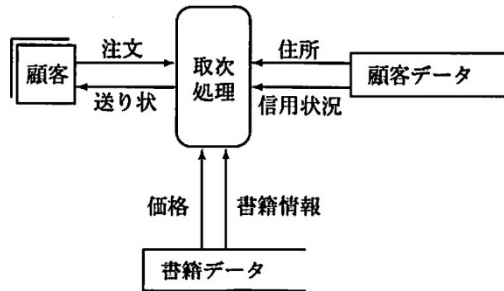
市販本の流通を例に, DFD のトップダウン展開について説明する.

① DFD の概略を作成

DFD を用いて業務分析を行う場合は,最初に業務の概略を DFD で表現する.

これは最上位の DFD で,業務の分析範囲と他の業務とのインタフェースを示す.

このような DFD を、コンテキストダイアグラムと呼ぶ。

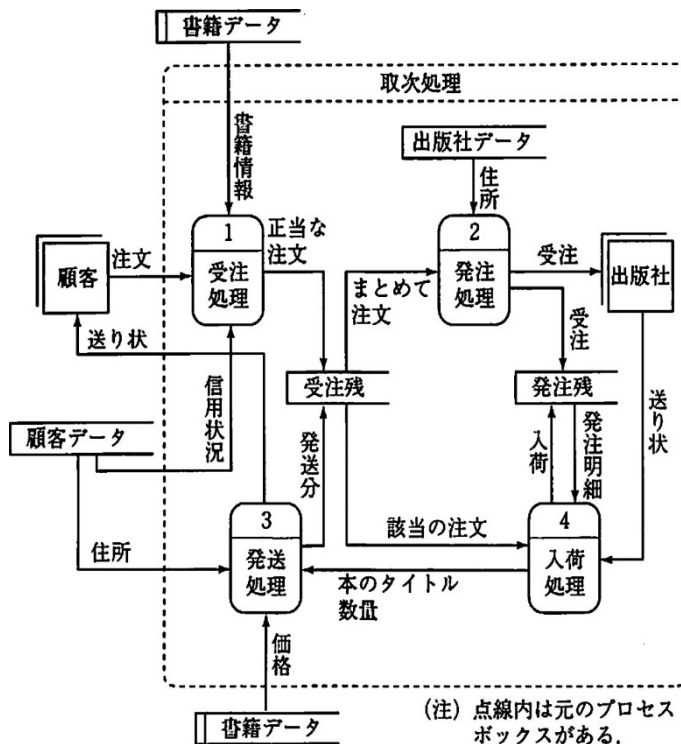


図表 1-6 コンテキストダイアグラム

②処理の分解

図表 1-6 の「取次処理」をより詳細に分解する. 分解は処理機能を明確にするために行うので, 場合によってはジョブ単位になるまで分解する.

ただし, どの程度まで分解するかは, 全体的に統一しなければならない.



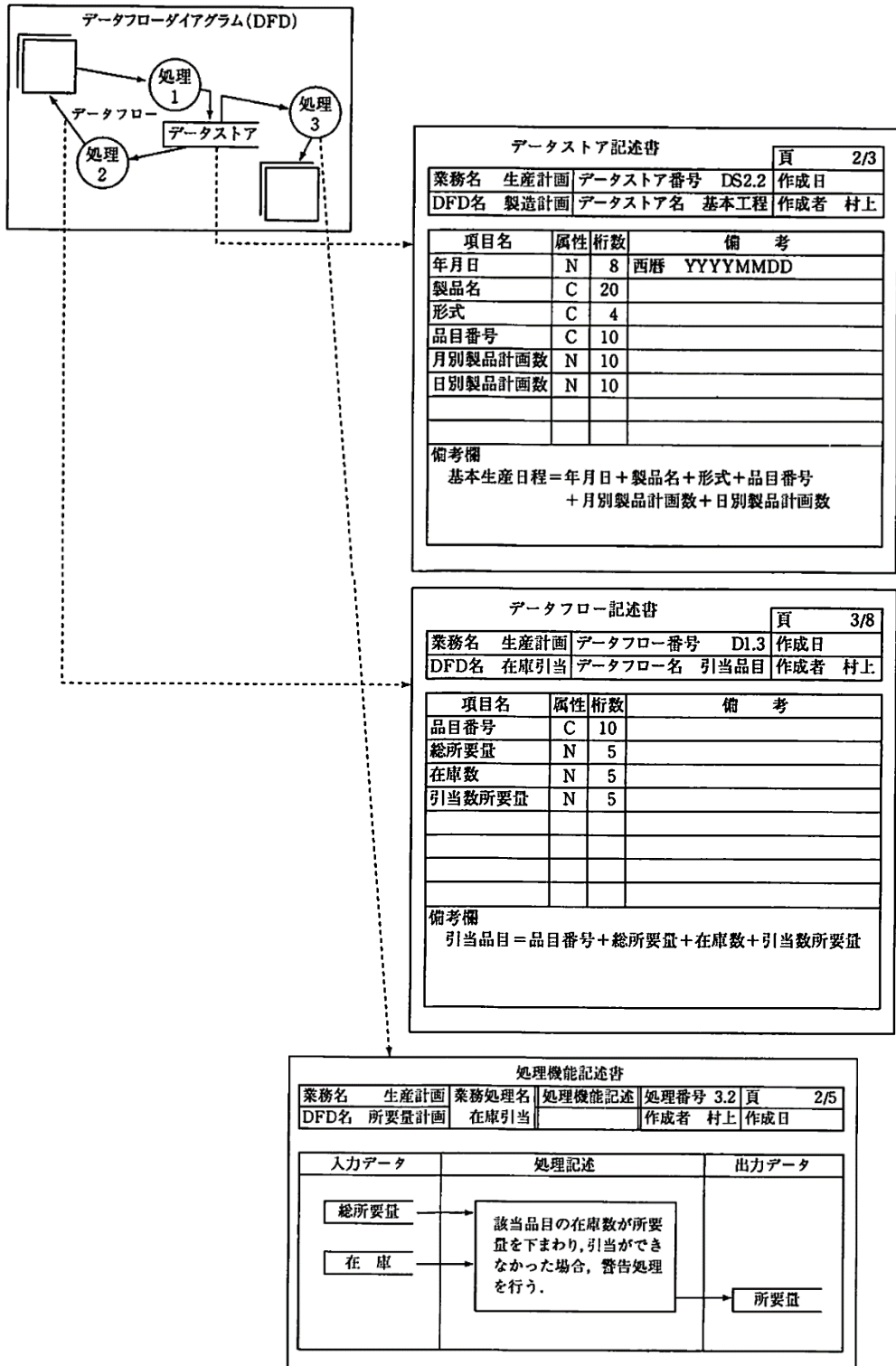
図表 1-7 プロセスボックスの分解

4) DFD の記述書

現行業務の流れを DFD で記述しても, それだけでは業務の詳細を表現することができない. DFD を中心とした仕様書を作成するためには, 以下に示す四つの記述書が必要である.

- ・データフローダイアグラム
- ・データフロー記述書(データフローの詳細を記述したもの)
- ・データストア記述書(データストアの詳細を記述したもの)

- ・ 処理機能記述書(処理(プロセスボックス)の詳細を記述したもの)



図表 1-8 DFD 記述書

なお、要求仕様を確認する段階で作成される DFD を新論理モデルという。

(2) HIPO (Hierarchy Plus input Process Output:階層入力処理出力記述法)

HIPO は設計の補助手段とドキュメント化を支援するための手法である。

従来のドキュメント化は、設計が終了した段階で事後処理として文書化する場合が多かった。それに比べて、HIPO はシステムやプログラムの機能が設計者の頭の中にはっきりと残っている間に、それらの機能を文書化するために用いる作図法である。

システムやプログラムの機能を図によってわかりやすく表現し体系的に順次詳細化していくので、管理者・設計者・利用部門・開発担当者や保守担当者のさまざまな要求に応えることができる。

1) HIPO の特徴

HIPO には次のような特徴がある。

①管理・理解のしやすさ

システムの機能を理解しやすい形に表現する。組織図によく似た階層構造で表現されており、あるレベルのダイアグラムは、それぞれその上位レベルのサブセットになっているので、管理しやすい。

②入出力の明確化

個々の機能の入力と出力が見やすい形で記述されるので、入力データから出力データへの変化の過程を容易にとらえることができる。

③機能の記述

ある機能を実行するための論理(How)を記述するのではなく、実行する機能(What)を記述する。補足説明欄には、次工程の担当者への手引きとなるように、機能に関する追加情報を記入する。

2) HIPO パッケージの構成

HIPO は次の三つの書式によって構成されている。

- ・図式目次(VTOC)
- ・総括ダイアグラム(IPO ダイアグラム)
- ・詳細ダイアグラム

これらの三つの書式は、システムやプログラムの機能を図形によって順次詳細に解説していく。

①図式目次(VTOC)

図式目次は、HIPO によって記述するシステムやプログラムの機能の階層構造を表現する。パッケージ全体を通読しなくても、この図式目次によって特定の情報や図式を探し出すことができる

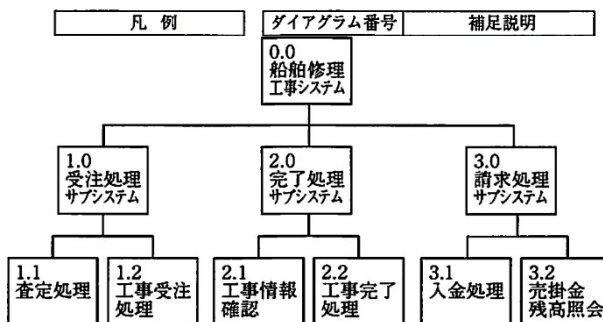
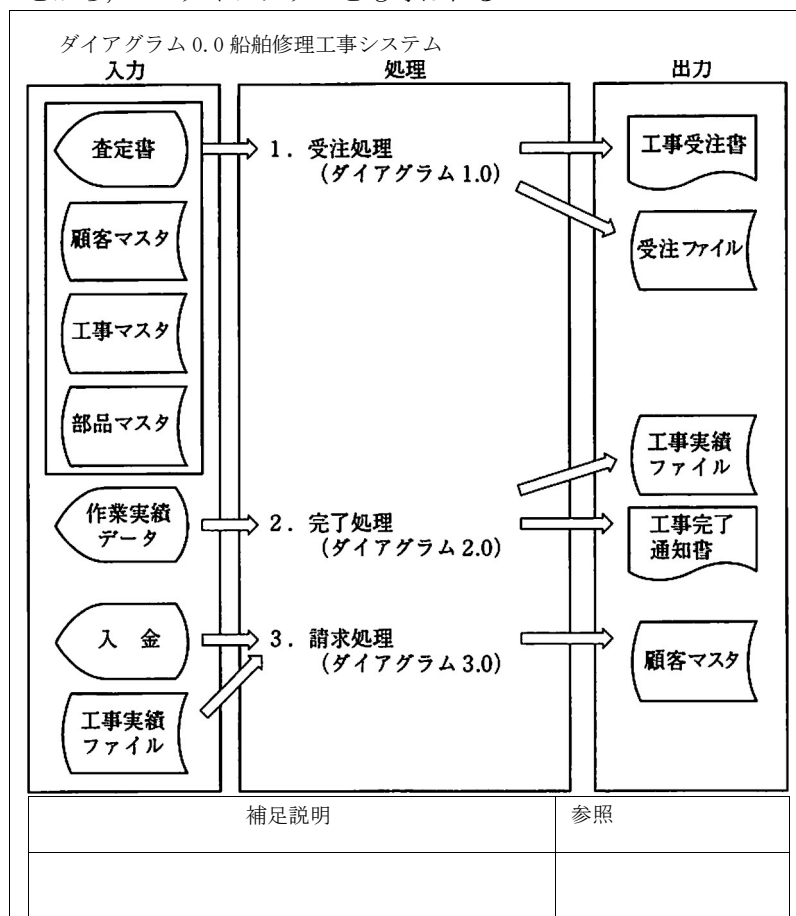


図 1-9 図式目次(VTOC)

②総括ダイアグラム（IPO ダイアグラム）

システムやプログラム全体の機能をまとめて(総括して)説明するために使用する。左側に入力部分(Input), 中央に処理部分(Process), 右側に出力部分(Output)を記入する欄があることから, IPO ダイアグラムとも呼ばれる。



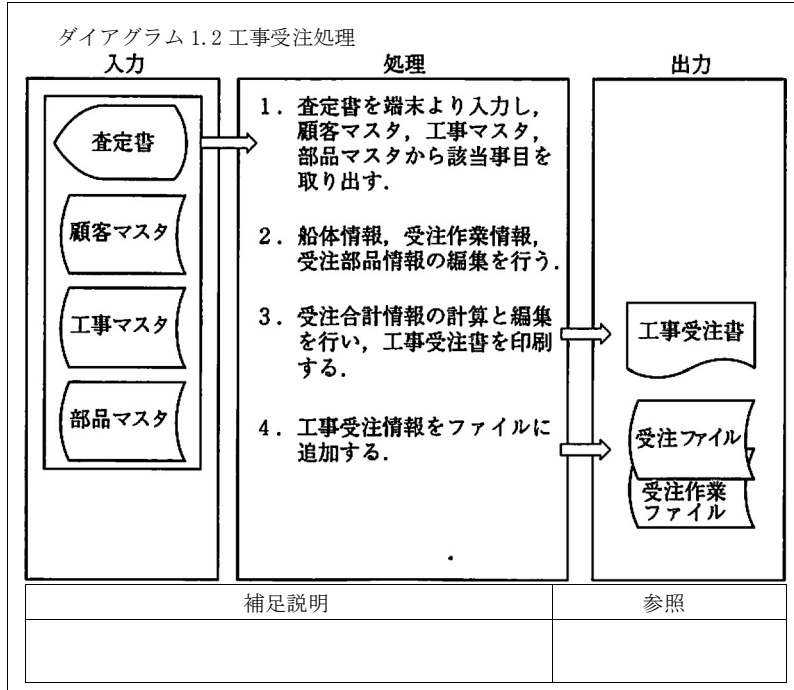
図表 1-10 総括ダイアグラム (IPO ダイアグラム)

③詳細ダイアグラム

総括ダイアグラムと同様の形式であるが、総括ダイアグラムよりも下位レベルの詳細な情報を記述する。

〈注意事項〉

- ・一般に、処理欄には機能を中心に記述し、処理手順や処理条件および次工程に対する手引きとなるような追加事項は、補足説明欄に記述する。
- ・参照欄は、コード表などの関連文書との対応関係を調べるときなどに利用する。



図表 1-11 詳細ダイアグラム

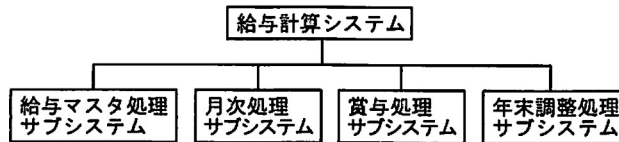
1. 3 サブシステムの定義と展開

ここでは、システムの最初のブレイクダウンである、サブシステムの定義と展開について学習する

1. 3. 1 サブシステムへの分割

大規模なシステムの場合は、基本計画で作成したシステムの流れを時間やデータによって関連性の高い、いくつかのサブシステムに分割する。

分割基準は大きさではなく、サブシステム間の関連性が最も弱くなる（インタフェースが少ない）ようにまとめる。

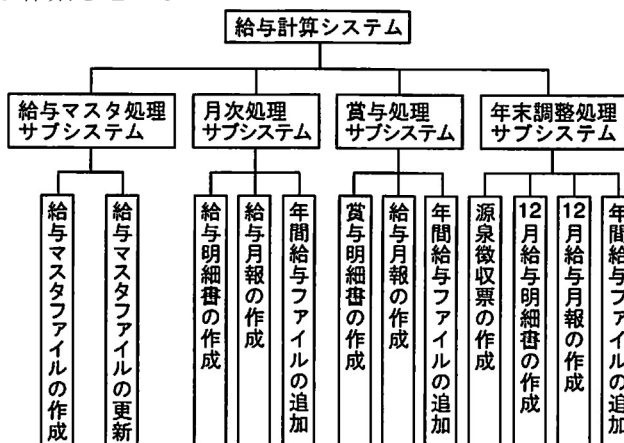


図表 1-12 システムとサブシステムの関係

1. 3. 2 サブシステムの展開と新物理モデルの作成

(1) サブシステムの展開

サブシステムの機能をさらに詳細化していき、結果として個別の”処理”の段階にまでブレイクダウンしていく。これをサブシステムの展開という。その際に、ユーザの要件が漏れていないかを確認しながら作業を進める。



図表 1-13 サブシステムの展開

(2) 新物理モデルの作成

上記(1)の結果を受け、それぞれ業務処理手順を検討し、要求仕様の確認で作成した新論理モデルに対して新物理モデル(新業務フロー)を作成する。このモデルには、具体的な内容が記述される。

1. 3. 3 ソフトウェアパッケージの組込み

サブシステムレベルの開発では、システム開発の生産性の向上や開発コストの削減などの面から、ソフトウェアパッケージの導入を考慮し、柔軟に対応する。仮に、サブシステムがソフトウェアパッケージのみで対応できなかったり、機能的に不十分な場合には、不足部分のみ開発したり、場合によってはカスタマイズを行うなどして対応する。

また、既存のソフトウェアを改良したり、部品化されているソフトウェアをサブシステムに利用できるのであれば、積極的に利用する。なお、これらの場合においては、必要ならば利用許諾契約を結ぶ。

1. 3. 4 性能・規模の見積り

システムの性能や開発規模を、サブシステム単位で見積もる。

(1)システム性能の見積り

サブシステムに対する次のような性能要件を見積もることである。

- ・アクセス頻度(単位時間当たりのアクセス数)
- ・ハードウェア構成に基づく基本機能ごとの応答時間や処理能力

算出結果と要求仕様書の性能要件を比較し、性能要件に満たない場合には対応策を検討する。

(2)開発規模の見積り

ファンクションポイント法などを用いて開発規模の見積りを行う。

(3)クライアントサーバシステムにおける性能・規模の見積り

クライアントサーバシステムなどにおいては、外部設計フェーズでサーバやクライアント、データベースなどの分割単位を決定する。大規模データベースが必要なのか、またパソコンレベルで十分対応できるのかなど、さまざまな観点から検討し、最適な資源を配置する。なお、この場合の単位とは、ネットワーク(LAN など)に接続するときの単位のことである。

1. 4 画面設計

画面設計は、ヒューマンインタフェースの設計のなかでも特に重要な位置を占めている。画面の使い勝手でシステムの品質を評価してしまうことがあるくらい、ユーザにとって画面は重要なものである。ここでは、画面設計の意義とその手順について学習する。

1. 4. 1 画面設計とは

(入力)画面は、ユーザがシステムに接することが多い、最も身近な部分であり、最近では、GUI (グラフィカルユーザインタフェース)を駆使した画面が多く用いられている。

GUI とは、直感的に識別できるアイコン(絵文字)やドロップダウン・ポップアップメニューなどを用いた、対話処理が容易なインタフェース方式である。画面上のポインティングデバイス(マウスなど)を用いて、矢印などを画面上の任意の位置に移動させ、アイコンやメニュー項目を指示し、処理を行う。

パソコンでは GUI 機能が OS に組み込まれているが、UNIX では GUI 機能を別に導入する必要がある。また、アイコンの表示形式や意味および操作方法はそれぞれの GUI により若干異なるが、GUI ごとに標準化されている。



図 1-14 GUI

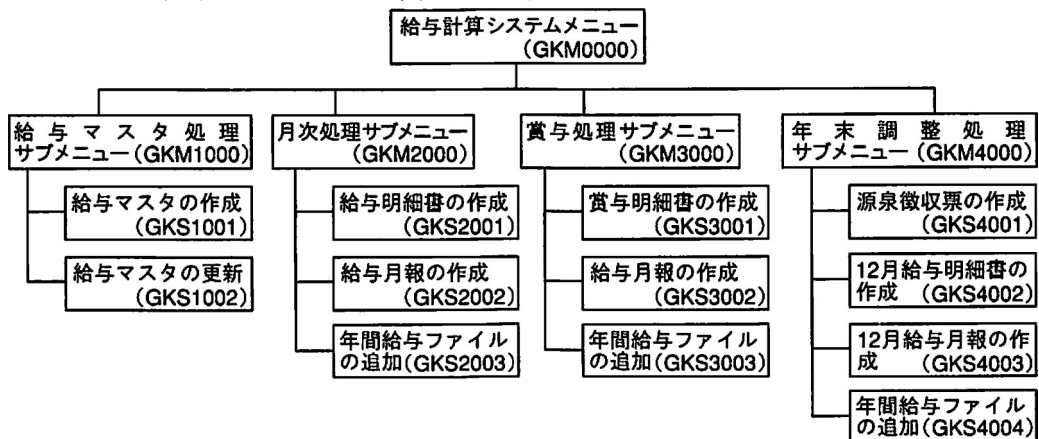
1. 4. 2 画面設計の手順と作業内容

画面設計は次のような手順で行われる。

- ①全体像の作成(使用するすべての画面の全体構成図を作成する)
- ②画面設計の標準化(画面のレイアウトや流れを標準化する)
- ③画面フローの設計(どのような順序で対話処理を行っていくかを設計する)
- ④画面の表示方法の検討(画面の使用頻度, 利用者の熟練度などを考慮して行う)
- ⑤画面レイアウトの設計(画面の構成・配置などを設計する) 画面設計の作業内容を以下に簡単に説明していく。

(1)全体像の作成

開発対象システムで用いる画面を, 関連性を含めてすべて洗い出し, 全体の構成図を作成する. 一般的には, メニュー・サブメニュー・処理などで構成されることが多い. このとき, 全体を通じて画面番号をつけておくと, 管理しやすい



(画面番号の意味) GKMxxxx

G : 画面の略

K : 給与計算システムの略

M : M=メニュー S=処理

m : サブシステムの識別番号

xxx : 連番

図表 1-15 画面全体像の作成

(2)画面設計の標準化

画面レイアウト(項目の配置場所など)や画面フローを標準化することにより, 開発側・ユーザ側ともに生産性の向上を図ることができる。

〈標準化対象〉

- ① 表示位置(タイトルの位置や項目の表示場所など)
- ② 表示方法(色彩・表示する語句の統一など)
- ③ 入力方法(選択式か, 直接入力かなど)

④ PF キー(プログラムファンクションキー)の使用方法

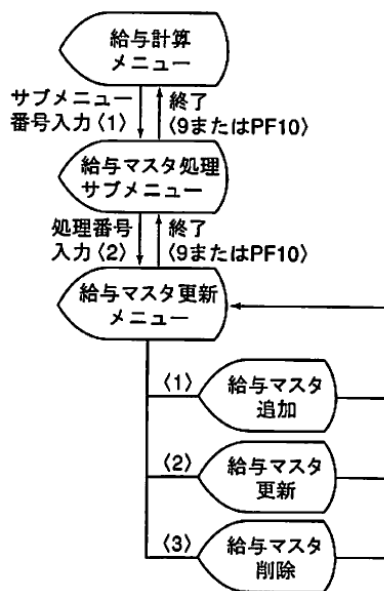
| (画面番号) | 適用業務システム共通域 | (端末番号) |
|---|-------------|-----------------|
| 見出し | | |
| A A A A | B B B B C C | D D D D E E E E |
| 適用業務入出力域 | | |
| x x コード [] | | |
| x x 名 [] | | |
| 入力領域を [] で囲んだりすると分かりやすい | | |
| 操作ガイドメッセージ | | |
| エラーメッセージ(適用業務メッセージ) | | (日時) |

図表 1-16 画面レイアウトの標準化(例)

(3)画面フローの設計

業務処理はいくつかの画面を使用して行われる. それらの画面をどのような順序で表示し, 処理を実行させるのかを設計する.

その際に, 登録処理, 検索処理, 更新処理, 削除処理などの画面の流れをパターン化しておけば, 開発者・ユーザ共に生産性を上げることができる.

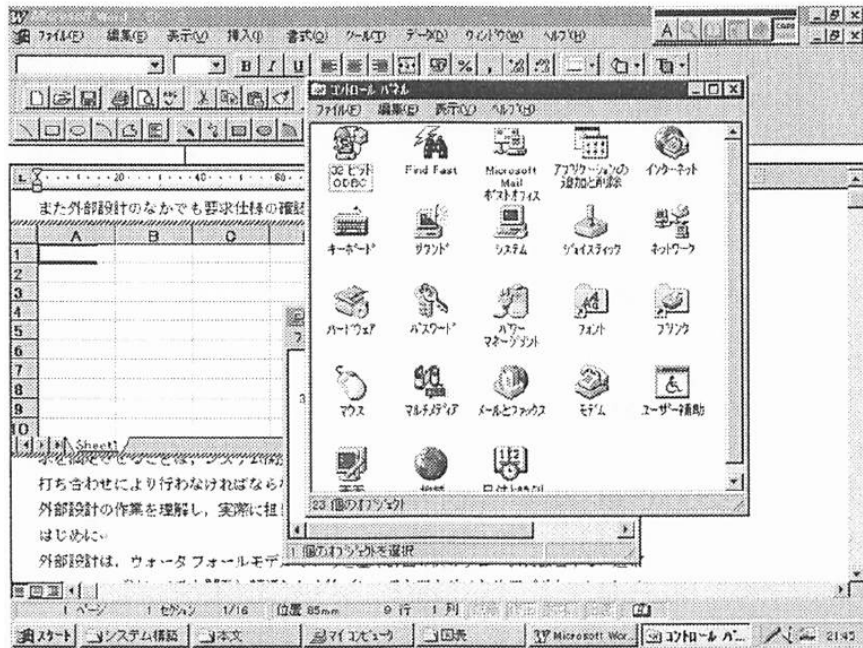


図表 1-17 画面フローの例

(4) 画面の表示方法の検討

システムを使うユーザの使用頻度や熟練度によって、画面の表示方法を決める。たとえば、初心者が画面を操作する場合には、コマンド入力方式(画面に直接コマンドを入力する方法)では生産性が上がらない。あくまでもユーザの立場に立った設計が必要である。

最近では、同時に複数の画面を表示し、同時処理を可能にしたマルチウインドウ形式の画面も多い。この形式はユーザにとっても使いやすいが、ポインティングデバイス(マウスなど)による操作が必要なので、システム講習会の際にはその操作練習なども考慮しなければならない。



図表 1-18 マルチウインドウの例

(5) 画面レイアウトの設計

表示画面のレイアウトを行う. その一例を図表 1-19 に示す. なお, ケースによっては画面のイメージだけを外部設計で確認し, その後の内部設計で専用用紙を用いて設計することもある.

| | | | | | | | | | | | |
|---------------------------------------|-------------|----------------|------------|------------|------------|------------|------------|------------|---------|----|--|
| 画面レイアウト | | | | 日付 | / | / | 作成担当 | | 承認 | | |
| 画面名称 | 給与マスタ更新メニュー | | | | | | | 画面ID | GKS1002 | | |
| | 1-10 | 11-20 | 21-30 | 31-40 | 41-50 | 51-60 | 61-70 | 71-80 | | | |
| | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | | | |
| 1 | GKS1002 | | | | | | | | TTT010 | 1 | |
| 2 | | *** | 給与 | マスタ | 更新 | メニュー | *** | | | 2 | |
| 3 | | | | | | | | | | 3 | |
| 4 | | | | | | | | | | 4 | |
| 5 | | | | | | | | | | 5 | |
| 6 | | | 1. 更新 | | | | | | | 6 | |
| 7 | | | 2. 追加 | | | | | | | 7 | |
| 8 | | | 3. 削除 | | | | | | | 8 | |
| 9 | | | | | | | | | | 9 | |
| 10 | | | | | | | | | | 10 | |
| 1 | | | 9. 終了 | | | | | | | 1 | |
| 2 | | | | | | | | | | 2 | |
| 3 | | | | | | | | | | 3 | |
| 4 | | | | | | | | | | 4 | |
| 5 | | | | | | | | | | 5 | |
| 6 | | | | | | | | | | 6 | |
| 7 | | | | | | | | | | 7 | |
| 8 | | | | | | | | | | 8 | |
| 9 | | | | | | | | | | 9 | |
| 20 | | | | | | | | | | 20 | |
| 1 | PF10 | 給与マスタ処理サブメニューへ | | | | | | | | 1 | |
| 2 | | | | | | | | | | 2 | |
| 3 | | | | | | | | | | 3 | |
| 4 | | | | | | | | | | 4 | |
| | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | 1234567890 | | | |
| 特記事項 | | | | | | | | | | | |
| ・ 該当する処理番号を入力することにより, それぞれの処理画面に遷移する. | | | | | | | | | | | |
| ・ 9を入力するかPF10を押すことにより, サブメニューへ遷移する. | | | | | | | | | | | |

図表 1-19 ディスプレイレイアウト

最近では, GUI 構築ツールやビジュアル言語を用いてプロトタイピングを作成し, 検討・評価することによりユーザの承認を得て, レイアウトを決定している.
なお, GUI による画面設計の詳細は, 「内部設計」で行う.

1. 5 報告書設計

報告書の設計は画面設計と同様、ユーザがシステムを利用する際の重要なポイント部分であり、ユーザにとってわかりやすい、利用しやすい報告書を設計する必要がある。

ただし、報告書の設計というと、プリントアウトするための帳票設計をイメージしがちであるが、画面に出力した処理結果も報告書であるということに注意しておこう。

1. 5. 1 報告書の設計とは

報告書は、システムでの処理結果を具体的な形で、しかもわかりやすく表現するためのものである。したがって、ユーザの立場に立って、フォーマットなどを検討する必要がある。なお、ハードウェアの制約を受ける部分(印刷方式など)や詳細な部分の設計については内部設計で行うので、ここでは報告書の大きさや項目の配置など、出力のイメージを設計するだけでよい。

1. 5. 2 報告書設計の手順と作業内容

報告書の設計は次の手順で行う。

- ①出力に関する総合的な検討
- ②出力方式・媒体の決定
- ③レイアウトの作成

各作業の詳細は次のとおりである。

(1)出力に関する総合的な検討

報告書を作成するにあたっての目的や時期などについて、総合的に検討する。

検討項目としては、

報告書名、使用目的、作成周期・時期、期限、配付先、数量、形式、出力項目の位置などがある。

1)報告書名

設計内容にふさわしい報告書名をつける。

2)使用目的

外部提出用なのか、内部資料用なのかを明確にする。外部提出用なら専用のフォームが決まっている場合もあるので、注意が必要である。

3)作成周期・時期

報告書がいつ必要なのか検討する(日単位か、週単位か、月次単位か、それとも決算で必要なのかなど)。

4)期限(制約)

報告書をいつ出力するのかを検討する。

5)配付先

作成した報告書をどこに配付するのかを明確にする。

6) 数量

報告書のページ数や画面数がどのくらいになるのかを検討する。

7) 形式

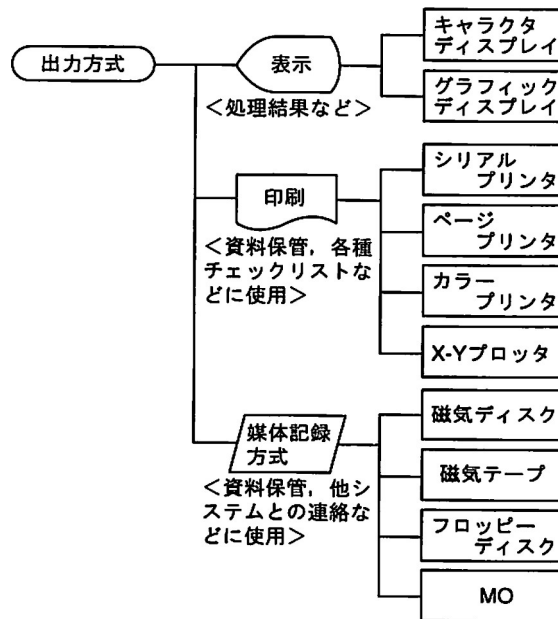
報告書に印字(表示)される項目(フィールド)のけた数やタイプ(数字か, 漢字かなど)を検討する。

8) 出力項目の位置

項目の配置場所などの出力イメージを明確にする。

(2) 出力方式・媒体の決定

報告書の性質や使用目的などを考慮し, 最適な出力方式と媒体を決定する。データとして記憶媒体に記録されるケースもよくある。



図表 1-20 出力方式と媒体

(3) レイアウトの作成

上記(1)～(2)をもとに出力仕様をまとめ, レイアウト(イメージ)を作成する。出力レイアウト(イメージ)についても, 文字間隔や項目レイアウトの配慮, 出力年月日の自動出力など, ユーザの立場に立った設計が必要である。

| | | | |
|----------|--|------|-------|
| サブシステム名称 | | 資料名 | 作成年月日 |
| 給与マスタ処理 | | 出力仕様 | |

| | | | |
|---|-------------|---------|--|
| 情報名 | | 業務名 | |
| 給与マスタ更新リスト | | 給与マスタ更新 | |
| 使用目的 | | | |
| 給与マスタ更新時のチェックリスト | | | |
| 周期・時期 | 出力方法 | 配付先 | |
| 逐次 | バッチ | | |
| 出力装置 | 出力媒体 | 期限 | |
| ページプリンタ | プリンタ用紙 (B4) | | |
| 情報量 | | | |
| 備考 | | | |
| ・ 1 ページ 7 人分出力 (1 ページ 55 行出力) ・ 社員コード順に情報を印字 | | | |

| NO | 項目名 | 文字種類 | 桁数 | 備考 |
|----|--------|------|----|----|
| 1 | 所属コード | N | 4 | |
| 2 | 社員コード | N | 5 | |
| 3 | 役職コード | N | 2 | |
| 4 | 氏名 | K | 10 | |
| 5 | 住所 | K | 40 | |
| 6 | 生年月日 | N | 8 | |
| 7 | 配偶者コード | N | 1 | |
| 8 | 扶養者親族数 | N | 1 | |
| 9 | 性別コード | N | 1 | |
| 10 | 基本給 | N | 8 | |
| 11 | 職務手当 | N | 6 | |
| 12 | 家族手当 | N | 6 | |
| 13 | 住宅手当 | N | 6 | |
| 14 | 通勤手当 | N | 6 | |
| 15 | 住民税 | N | 6 | |
| 16 | | | | |
| 17 | | | | |

図表 1-21 出力仕様

| 給与マスター一覧表 | | | | | | 999 頁 |
|-------------|------|------------|----|----|--|-------|
| 社員コード 9...9 | | 氏名 N...N | | 性別 | | |
| | | 住所 N.....N | | | | |
| 所属コード | 所属名 | | | | | |
| 役職コード | 役職名 | 等級 | 号級 | | | |
| 基本給 | 役職手当 | | ~ | | | |
| | | | ~ | | | |
| ... | | | | | | |
| 社員コード 9...9 | | 氏名 N...N | | 性別 | | |
| | | 住所 N.....N | | | | |
| 所属コード | 所属名 | | | | | |
| 役職コード | 役職名 | 等級 | 号級 | | | |
| 基本給 | 役職手当 | | ~ | | | |
| | | | ~ | | | |
| ... | | | | | | |
| 社員コード 9...9 | | 氏名 N...N | | 性別 | | |
| | | 住所 N.....N | | | | |
| 所属コード | 所属名 | | | | | |
| 役職コード | 役職名 | 等級 | 号級 | | | |
| 基本給 | 役職手当 | | ~ | | | |
| | | | ~ | | | |

- ・ 1 ページ 7 人分出力 (1 ページ 55 行出力)
- ・ 社員コード順に情報を印字
- ・ 氏名が変わったときに空白行を 1 行出力

図表 1-22 出力イメージ

1. 6 コード設計

都道府県コードや郵便番号、車のナンバーや製品番号など、まわりを見渡すと、さまざまなコードがあふれている。コンピュータで処理し管理するために情報をコード化すれば「データ」として扱いやすくなるからである。ここでは、コード化について学習する。

1. 6. 1 コード設計とは

(1)コード化の目的

情報をコード化することにより、データの識別や分類・配列が容易になる。しかし、ただ単にコード化すればよいというものではなく、将来を見すえた拡張や変更など、柔軟性に富んだコード設計を行わなければならない。コード設計を誤ると、関係するすべてのシステム(プログラム)やデータを修正しなければならないので、慎重に設計する必要がある。

外部設計で使用するデータの入出力コードを、特に外部コードという。

(2)コードの機能

一般に、コード設計を行う場合には、次の四つの機能が要求される。

1)識別機能

他のデータと区別するための機能(ユニーク性)。

たとえば、顧客管理を行う場合、同姓同名の顧客を顧客コード(同姓同名であっても別のコードを付与)によって識別することができる。

2)分類機能

データを分類する機能。

たとえば、顧客を年代別、性別に分けるなど、データを整理し体系的に分類することができる。

3)配列機能

データの順番を決定する(並べ替え)機能。

たとえば、生年月日順とか、五十音順とか、都道府県順などに並べ替えることによって、データを分析・整理することができる。

4)チェック機能

入力されたコードが正しいかどうかをチェックできる機能。

通常は、本来のコード部分に対して、モジュラス 11 などの計算を行い、その計算結果(チェックディジット)を の最下位に付け加える。

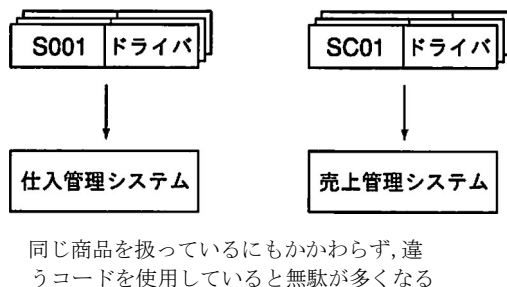
1. 6. 2 コード設計上の留意点

間違ったコード設計は、システム全体を左右するような大きな問題に発展する場合もあるので、コードを設計する際は、以下に述べる事項に注意して慎重に行わなければならない。

(1) コードの適用範囲と使用期間

1) 適用範囲

コードの適用範囲によりコード体系が変わってくる。たとえば、外部との連携(他の会社とのネットワーク)で使用する場合は、業界などで用いている標準的なコード体系を用いる必要がある。また、組織内で使用する場合でも、全社共通で利用できるコード設計が望ましい。

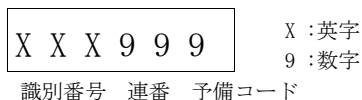


図表 1-23 商品コードの例
(仕入管理システムと売上管理システムで違っている例)

2) 使用期間

どのくらいの期間にわたってコードを使用するのか、将来業務量がどのくらい増加するのかなど、予測を誤ってコード設計すると、コードが不足する恐れがある。そのような場合は、コードの体系を見直さなければならず、結果としてシステム全体を修正する事態も起こりかねない。コード設計には余裕をもたせる必要がある。

具体的には、コードの末尾に拡張用の予備コードを設けたりする。



図表 1-24 拡張性をもたせたコード設計

(2) わかりやすさ

コードを設計するときの心がまえとしては、わかりやすさが大きなポイントの一つとなる。

わかりやすさとは

- 扱いやすさ(できるだけ短く、また単純に)
- 体系化(グループに分類することができる)
- 明瞭性(数字を原則とし必要に応じて英字や仮名を入れる)

を備えていることである.

| | | | | | |
|---|---|---|---|---|---|
| 9 | 9 | 9 | 9 | 9 | 9 |
|---|---|---|---|---|---|

学科コード 年制 連番

図表 1-25 コードの体系化

1. 6. 3 コード設計の手順と作業内容

コード設計は次のような手順で行われる.

- ①コード化対象の選定
- ②コード化の目的の明確化
- ③使用期間とデータ量の予測
- ④利用範囲の決定
- ⑤コード化作業とコード表作成
- ⑥コードファイルの作成

各作業の詳細は次のとおりである.

(1)コード化対象の選定

入出力データの調査・分析,作成した画面レイアウトなどの結果から,コード化の対象となる候補を抜き出す.そして,業務処理上あるいはコンピュータ処理上,コード化したほうがよいと思われる項目を決定する.

(2)コード化の目的の明確化

コード化の目的によりコード体系が変わってくるので,前提として,コード化の目的を明確にする必要がある.たとえば,データを参照(検索)するためにコード化が必要なのか,または大量のデータを分類するためにコード化したいのか,などである.

(3)使用期間とデータ量の予測

現在のデータ件数と将来の予想増加数により,コード化の対象となるデータ件数を予測する.これは非常に大切であり,システム稼働中に,実際のデータ件数がコード設計上のデータ件数を上まわってしまうと,ファイルやプログラムの変更などシステム全体の修正が必要になってくる.使用期間を考慮して,件数に余裕をもたせた設計が必要である.

(4)利用範囲(適用範囲)の決定

設計したコードをどの業務に適用するのか,他の業務についても利用できるのかなど,コードの適用範囲を決定する.適用範囲によっては,コードのけた数を増やさなくてはならないからである.また,前述のように他企業との連携(ネットワークなど)で使用する場合は,JISコードやJANコードなどの標準的なコードを用いたほうがよい.

(5)コード化作業とコード表作成

一般的なコードの種類と特徴を知り,これまでの(1)～(4)の結果をふまえて,コード体系やけた数(データ件数を考慮),チェックディジットなどを考慮し,具体的にコードを設計

しコード設計書を作成する.そして、具体的にデータにコードを割り当てたものをコード表にまとめる.このコード表を集めて1冊にしたものを、コードブックという.

図表 1-26 にコード設計書の例を示す.

| | | | | | | | |
|--------|----------|---|----------|-----|--|----|--|
| コード設計書 | 日付 | / | / | 作成者 | | 承認 | |
| システム名称 | 給与計算システム | | サブシステム名称 | | | | |
| コード名 | 所属コード | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|--|----|--|--|--|--|--|-----|----|-----|---|------|---|------|---|------|---|-----|---|------|---|------|---|------|---|-------|---|-----|---|-----|---|-----|---|-----|---|-----|---|
| 目的 | ・所属（本支社，部，課）を識別し，分類する。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 構成 | 9 9 9 9 （4桁） ① ② ③ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | ①本社コード（10～70） ②部コード（1～3） ③課コード（1～5） | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 付番要領 | <div style="text-align: center;"><table border="1"><tr><td>本 社</td><td>10</td></tr></table></div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"><div style="text-align: center;"><table border="1"><tr><td>営業部</td><td>1</td></tr></table><div style="display: flex; flex-direction: column; align-items: center;"><table border="1"><tr><td>営業1課</td></tr><tr><td>1</td></tr></table><table border="1"><tr><td>営業2課</td></tr><tr><td>2</td></tr></table><table border="1"><tr><td>営業3課</td></tr><tr><td>3</td></tr></table></div></div><div style="text-align: center;"><table border="1"><tr><td>開発部</td><td>2</td></tr></table><div style="display: flex; flex-direction: column; align-items: center;"><table border="1"><tr><td>開発1課</td></tr><tr><td>1</td></tr></table><table border="1"><tr><td>開発2課</td></tr><tr><td>2</td></tr></table><table border="1"><tr><td>開発3課</td></tr><tr><td>3</td></tr></table></div></div><div style="text-align: center;"><table border="1"><tr><td>総務人事部</td><td>3</td></tr></table><div style="display: flex; flex-direction: column; align-items: center;"><table border="1"><tr><td>庶務課</td></tr><tr><td>1</td></tr></table><table border="1"><tr><td>人事課</td></tr><tr><td>2</td></tr></table><table border="1"><tr><td>経理課</td></tr><tr><td>3</td></tr></table><table border="1"><tr><td>企画課</td></tr><tr><td>4</td></tr></table><table border="1"><tr><td>秘書課</td></tr><tr><td>5</td></tr></table></div></div></div> | | | | | | | 本 社 | 10 | 営業部 | 1 | 営業1課 | 1 | 営業2課 | 2 | 営業3課 | 3 | 開発部 | 2 | 開発1課 | 1 | 開発2課 | 2 | 開発3課 | 3 | 総務人事部 | 3 | 庶務課 | 1 | 人事課 | 2 | 経理課 | 3 | 企画課 | 4 | 秘書課 | 5 |
| | 本 社 | 10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 営業部 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 営業1課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 営業2課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 営業3課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 開発部 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 開発1課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 開発2課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 開発3課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 総務人事部 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 庶務課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 人事課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 経理課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 企画課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 秘書課 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | | |
|-------|----|-----|----|----|----|
| 各 支 社 | | | | | |
| 札幌 | 仙台 | 名古屋 | 大阪 | 広島 | 福岡 |
| 20 | 30 | 40 | 50 | 60 | 70 |

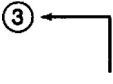
図表 1-26 設計書の例

(6)コードファイルの作成

業務で実際に使用するために、コード表の内容を記憶媒体(磁気ディスクなど)に記録する.これをコードファイルという.

1. 6. 4 よく使われるコードの例

われわれの身のまわりでよく用いられているコードの例を、図表 2ー27 に示す.

| コード名 | コードの内容 | 使用例 |
|-----------------------|--|---|
| 連番コード (シーケンスコード) | 先頭から順番に番号をつけていく. 作業としては簡単であるが. コード全体は体系化されない. | 〈JIS の都道府県コード〉 01 北海道 02 青森・・・47 沖縄 |
| 区分コード (ブロックコード) | ブロックごとに番号をつけた各ブロック内で順番に番号をつけていく. 分類には便利であるデータを追加する場合や件数が多い場合には不便である. | 〈JIS の大学・高等専門学校〉 国立大学 0001～ 公立大学 1001～ 私立大学 2001～ |
| 10 進コード | コード化の対象を 0 ～ 9 でコード化し各数字ごとにさらに 0 ～ 9 でコード化し順次細かくコード化していく. | 〈日本十進分類法(図書分類コード)〉 000 総記 010・・・図書館 020・・・図書. 書誌学 100 哲学 110・・・哲学各論 120・・・東洋思想 200 歴史 |
| けた別コード (グループ分類コード) | コードの各けたが意味をもっており, 分類ごとに集計する場合には便利であるが. データが多い場合はけた数が大きくなってしまう. | 〈JIS の産業分類コード〉 ①②③④ 12 84 1 1 ソフトウェア業 12 84 1 2 情報処理サービス業 12 84 1 3 情報提供サービス業 12 84 1 9 その他の情報サービス業 ①大分類コード(サービス業) ②中分類コード(情報サービス・調査・広告業) ③小分類コード(情報サービス業) ④細分類コード |
| 表意コード (モニックコード) | 商品の略称や記号などをコードとすため覚えやすいが. 分類には不便な場合がある. | 18CTV (18 インチのカラーテレビ) Y Y . M M. D D (年月日) |
| チェックディジット | コードのチェック用の数字であるチェックディジット(検査文字)をつけたコードで, 間違えてはならない重要なコードなどにつける. | 〈銀行の口座番号〉 1 2 6 5 X) 1 4 2 3 1 8 12 15 (1 + 8 + 12 + 15) ÷ 11 = 3 余り 3  |

図表 1-27 体系の例

1. 6. 5 コードのチェック方法

コードのチェック方法としては、チェックディジット法がある。ただし、このチェック法はコードの誤りを発見するためのものであり、訂正機能はない。

なお、コンピュータのメモリでは、ビット誤りを検出するだけでなく、自動訂正する誤り訂正符号(Error Correcting Code ; ECC)を採用するのが一般的になっている。

コードの誤りは、主に入力時に発生する。たとえば、似ている数字(3 と 8 など)の入力ミスや入力順番ミス(132 を 123 と入力)などである。

チェックディジットの計算方法を図表 1-28 で説明する。

<モジュラス 10 によるチェックディジットの計算>

①重みづけ(コードは“1011”とする)

1 0 1 1

5 4 3 2 <各けたの重み> ※下位のけたから軽い順に重みづけをする。

②各けたごとの乗算と加算

1 0 1 1

X X X X

※乗算が 2 けたになった場合には、1 けたずつに

5 4 3 2

分解して加算を行う。

5+0+3+2 = 10

③除算(モジュラスによる除算)

10 = 1 余り 0 → 0 がチェックディジット

(M :モジュラス)

(なお、モジュラスから余りを引いたものをチェックディジットする場合もある。)

④チェックディジットの付加

1 0 1 1 0

チェックディジット

図表 1-28 チェックディジットの計算方法

1. 7 論理データ設計

システムを利用する際にプログラムと同様に欠かせないのが、データベースやファイルなどのデータである。これらのデータを効率よく入出力させることによって、システム自体の性能が高くなる。そのための第1段階の設計として、この論理データ設計が行われる。

1.7.1 論理データ設計とは

要件仕様の確認, サブシステムの分解・展開, 入出力の設計で検討された入出力データから, システム内に蓄積すべきデータの構造とデータ項目を決定する。ただし, ファイル編成などの物理的な(具体的な)検討は次の内部設計で行うので, ここではデータ項目を洗い出し, さらにファイルやデータベースの候補を洗い出す。

1.7.2 論理データ設計の手順と作業内容

論理データ設計は次のような手順で行われる。

- ①データ項目の洗い出し
- ②データの関連性の分析
- ③ファイル候補の決定
- ④ファイル候補内のデータ項目と機能との対応づけ
- ⑤ファイル仕様の作成

各作業の詳細は次のとおりである。

(1) データ項目の洗い出し

まず、画面・報告書の設計結果や現在の入力伝票類などからデータ項目を洗い出して、それぞれにデータの内容を示す名称や必要なけた数、タイプ(数字か、漢字か、など)を明確にしておく。

次に、それらの検討結果をデータ項目検討表にまとめる。この表は、次の「データの関連性の分析」と「ファイル候補の決定」の作業を経て完成させる。

| データ項目検討表 | | 日付 | | ／ | | ／ | | 作成者 | | | | 承認 | | | |
|----------|--------|----------|----|-------|------|--------|---------|-------|--|--|--|------|--|--|--|
| システム名称 | | 給与計算システム | | | | | | | | | | | | | |
| サブシステム名称 | | 給与マスタ処理 | | | | | | | | | | | | | |
| データ項目 | | | | 入力情報名 | | | | 出力情報名 | | | | ファイル | | | |
| No | 名称 | 属性 | 桁数 | 人事情報 | 給与情報 | エラーリスト | 給与マスタ一覧 | 給与マスタ | | | | | | | |
| 1 | 所属コード | N | 4 | × | × | × | × | × | | | | | | | |
| 2 | 社員コード | N | 5 | × | × | × | × | ◎ | | | | | | | |
| 3 | 役職コード | N | 2 | × | × | × | × | × | | | | | | | |
| 4 | 氏名 | K | 10 | × | × | × | × | × | | | | | | | |
| 5 | 住所 | K | 40 | × | | × | × | × | | | | | | | |
| 6 | 生年月日 | N | 8 | × | | × | × | × | | | | | | | |
| 7 | 等級 | N | 1 | | × | × | × | × | | | | | | | |
| 8 | 号級 | N | 1 | | × | × | × | × | | | | | | | |
| 9 | 配偶者コード | N | 1 | × | | × | × | × | | | | | | | |
| 10 | 扶養者親族数 | N | 1 | × | | × | × | × | | | | | | | |
| 11 | 性別コード | N | 1 | × | | × | × | × | | | | | | | |
| 12 | 基本給 | N | 8 | | | | × | × | | | | | | | |
| 13 | 職務手当 | N | 6 | | × | × | × | × | | | | | | | |
| 14 | 家族手当 | N | 6 | | × | × | × | × | | | | | | | |
| 15 | 住宅手当 | N | 6 | | × | × | × | × | | | | | | | |
| 16 | 通勤手当 | N | 6 | | × | × | × | × | | | | | | | |
| 17 | 住民税 | N | 6 | | × | × | × | × | | | | | | | |
| 18 | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | | |
| 32 | | | | | | | | | | | | | | | |
| 33 | | | | | | | | | | | | | | | |
| 34 | | | | | | | | | | | | | | | |

×：使用するところを示す ◎：キー項目を示す

図表 1-29 データ項目検討表

(2) データの関連性の分析

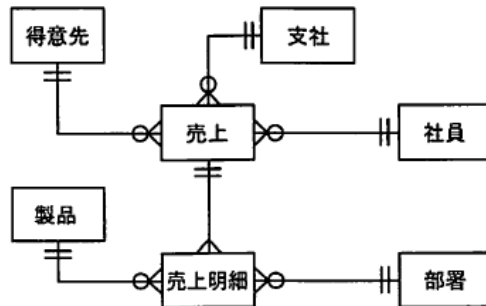
洗い出された複数の項目間には、次の例に示すようにそれぞれ関連性がある。

(例)社員コード:役職コード:社員名:住所

これらの項目は関連性ととともに、同時に使用することも多いので、ファイルとして定義される可能性が強いということになる。

ここでは、画面や報告書などをもとにデータ間の関連性を分析し、その結果を引き続きデータ項目検討表に記入する。なお、データ間の関連性分析には ER-D (Entity-Relationship Diagram)などが用いられる。

また、関連性ごとにデータの集合を分割・統合することを正規化という。



図表 1-30 ER-D

(3) ファイル候補の決定

(1) , (2) の過程を経てまとめられたファイル候補を決定し, 作業終了時にデータ項目検討表を作成する.

(4) ファイル候補内のデータ項目と機能との対応づけ

ファイル候補の各データ項目と, サブシステムの展開で分割された機能(処理単位)とを対応させ, ファイル項目の抜けや機能の定義漏れを再度チェックする. その際, データ項目と機能の対応表を用いる. これらの作業を行うことにより, 機能とファイルが決定する.

(5) ファイル仕様の作成

決定されたファイルについて, ファイルの仕様(ファイル候補検討表)を作成する.

| ファイル候補検討表 | | 日付 | / | / | 作成者 | | 承認 |
|-----------|-----------------------------|----|----|-----------------------|-----|--|----|
| システム名称 | 給与計算システム | | | | | | |
| サブシステム名称 | 給与マスタ処理, 月次処理, 貸与処理, 年末調整処理 | | | | | | |
| ファイル名称 | 給与マスタファイル | | | | | | |
| No | データ項目 | 属性 | 桁数 | 補足説明 | | | |
| 1 | 所属コード | N | 4 | 本支社 2 桁, 部 1 桁, 課 1 桁 | | | |
| 2 | 社員コード | N | 5 | 最後の 1 桁はチェックディジット | | | |
| 3 | 役職コード | N | 2 | | | | |
| 4 | 氏名 | K | 10 | | | | |
| 5 | 住所 | K | 40 | | | | |
| 6 | 生年月日 | N | 8 | 西暦 4 桁, 月 2 桁, 日 2 桁 | | | |
| 7 | 等級 | N | 1 | | | | |
| 8 | 号級 | N | 1 | | | | |
| 9 | 配偶者コード | N | 1 | | | | |
| 10 | 扶養者親族数 | N | 1 | | | | |
| 11 | 性別コード | N | 1 | | | | |
| 12 | 基本給 | N | 8 | | | | |
| 13 | 職務手当 | N | 6 | | | | |
| 14 | 家族手当 | N | 6 | | | | |
| 15 | 住宅手当 | N | 6 | | | | |
| 16 | 通勤手当 | N | 6 | | | | |
| 17 | 住民税 | N | 6 | | | | |
| 18 | | | | | | | |
| 19 | | | | | | | |
| 20 | | | | | | | |

(属性) X : 英数字 N : 数字 K : 漢字

図表 1-31 ファイル候補検討表

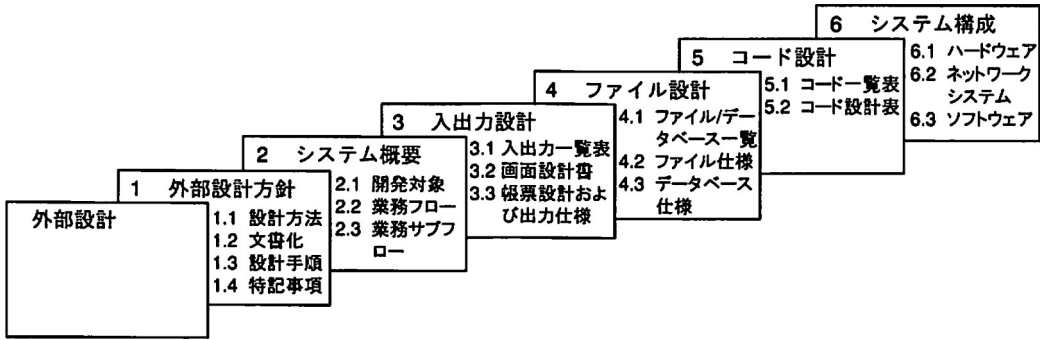
1. 8 外部設計書の作成

外部設計フェーズで作成したさまざまな資料をもとにして、外部設計書を作成する。外部設計はユーザの要求に基づいた設計であり、その結果をまとめた外部設計書は経営トップやユーザの承認を得なければならない。したがって、外部設計書を作成するときにはなるべく専門用語を避け、わかりやすかつ明瞭に記述しなければならない。

1. 8. 1 外部設計書とは

外部設計において行われた数々の作業の成果が外部設計書としてまとめられ、この外部設計書をもとに次の工程である内部設計が行われる。また、外部設計書はユーザと開発側との重要な接点となるドキュメントなので、ユーザの要望を配慮した記述内容でなければならない。

い。たとえば、外部設計書は図表 2-32 に示すような項目から構成される。



図表 1-32 外部設計書

1. 8. 2 外部設計書の構成と内容

ここでは、外部設計書の詳細について説明する。

(1) 業務フロー

サブシステムを展開し、作成した DFD（新物理 DFD）を記述する。この段階の DFD は、業務の大きな流れを示す最上位のコンテキストダイアグラムである。

(2) 業務サブフロー

コンテキストダイアグラムのプロセスボックスがサブシステムであり、そのサブシステムの流れを DFD で詳細に記述する。さらに、DFD のプロセスボックスを詳細化したものが基本機能であり、必要に応じて再度詳細化していく。基本機能については処理機能記述書を作成し、入力・処理・出力の詳細を記述する。（詳細は、図表 1-8 を参照のこと）

(3) 入出力一覧表(レイアウトを含む)

画面および入出力帳票(伝票など)の一覧表を作成し, 画面・入出力イメージも添付する.

(4) ファイル一覧表

論理データ設計で決定したファイルの一覧表およびファイル仕様を添付する.

(5) コード設計表

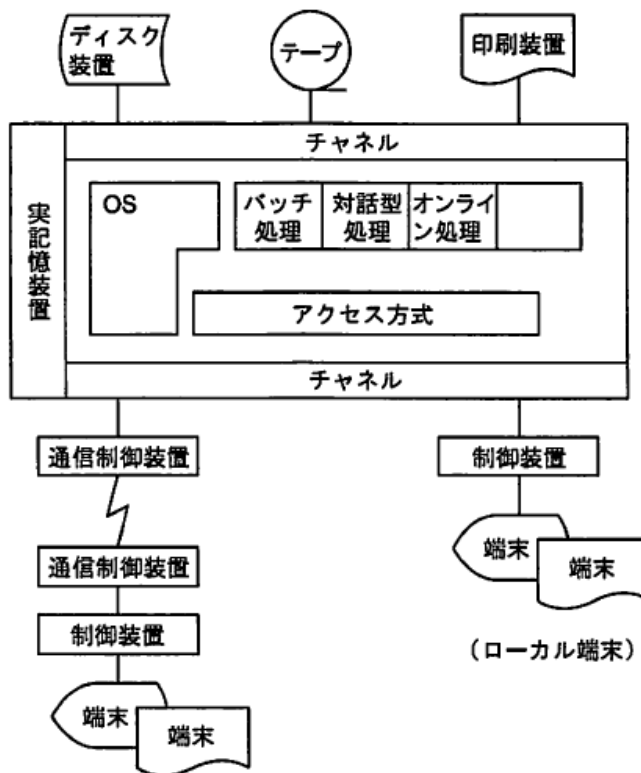
コード設計によってコード化されたコードの一覧表およびコード設計書を添付する.

(6) システム環境の記述

システムで使用するハードウェア, ネットワーク, ソフトウェアなどについて記述する.

1) ハードウェア

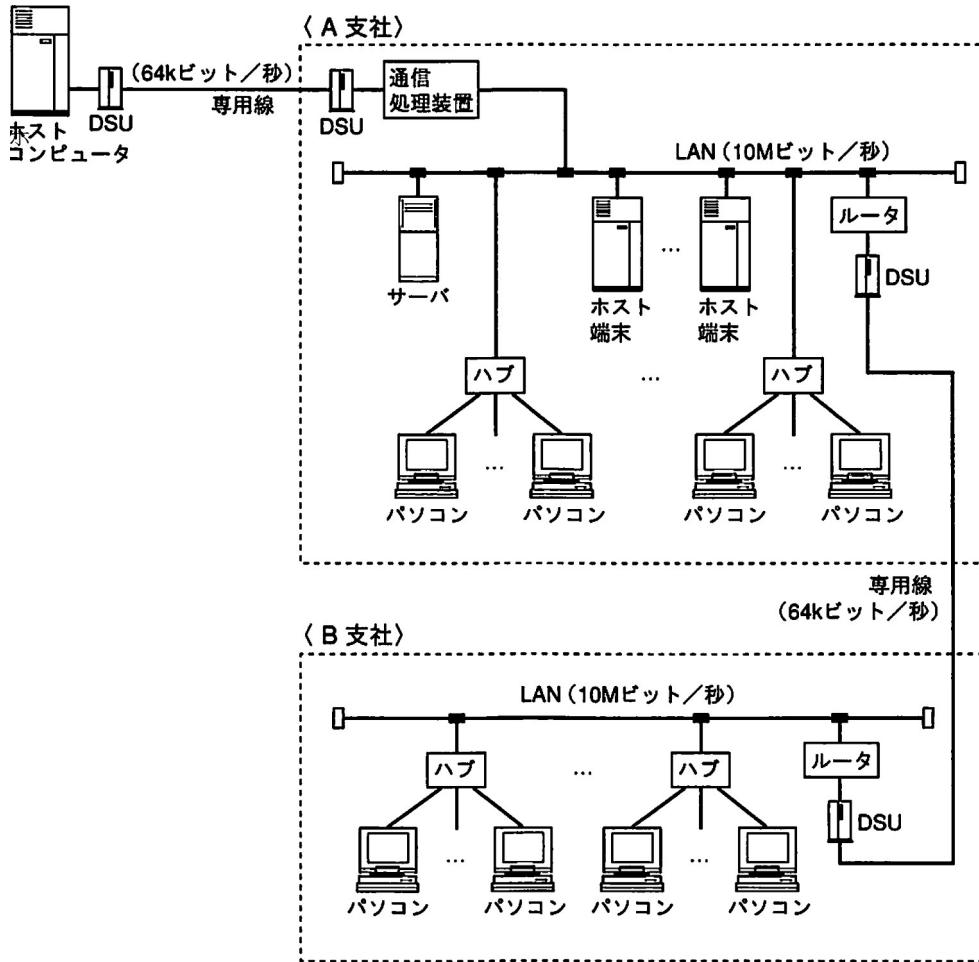
システムの機器構成の概略図を記述する. チャンネルの数や通信回線の本数なども記入しておくといよい.



図表 1-33 システムの機器構成図

2) ネットワーク

クライアントサーバシステムや, WAN などのネットワークを利用したシステムの場合には, ネットワーク構成図を示す必要がある. 特に, 回線容量やプロトコルのほか, LAN の種類(イーサネットなど)や WAN の詳細(ISDN 使用か, 高速デジタル回線使用かなど)も記入しておく. ネットワーク構成図の例を図表 1-34 に示す.



図表 1-34 ネットワーク構成図

3) ソフトウェア

システムで使用する(あるいは使用予定の), 次のようなソフトウェアを記入する.

- ① OS
- ② ユーティリティプログラム
- ③ 使用予定のプログラム言語
- ④ DBMS (データベース管理システム)
- ⑤ 開発支援ツール(CASE ツール, GIII構築ツール, ビジュアル言語など)
- ⑥ サブシステムで導入予定のソフトウェアパッケージ
- ⑦ 通信関係のソフトウェア

なお, ④~⑥については, マニュアルやパンフレットに記述してある程度の簡単な概要(必要なメモリ容量, CPU のグレード, 適用 OS など)を併せて記述しておくといよい.

1. 9 理解度確認問題

【問題】

外部設計の作業に関する次の記述中の□□に入れるべき適切な字句を、解答群の中から選べ。

外部設計では、システムの機能やヒューマンインタフェースについての定義が行われる。それらの作業を次の①～⑥に示す。

- ① □ a □ : 各種分析技法を用いて求められているシステムの機能を分析
- ② □ b □ : システムをサブシステムおよび機能単位に分割・展開
- ③ □ c □ : 画面および報告書の設計
- ④ □ d □ : システムで使用するコードの設計
- ⑤ □ e □ : ファイルやデータベースの基本設計
- ⑥ □ f □ : 外部設計フェーズで作成される成果物

□ a □ は、基本計画において作成した要求仕様書をベースに、さらに詳細にシステムの目的やユーザの要求などを確認する。その分析技法としては□ g □、HIPO、状態遷移図などがある。

□ b □ は、先の基本計画で作成したシステムの流れを時間的・データの関連性の高いものをまとめて、いくつかのサブシステムに分割し、さらにサブシステムの機能を詳細化していく。

□ c □ は、□ h □ を決定する重要な作業である。なお、□ c □ においては、近年ユーザの使い勝手を改善した□ i □ の採用が急増している。

□ d □ は、情報をコード化する作業である。この作業を行うことによりデータを識別し、分類・配列がしやすくなる。

□ e □ は、□ a □ ・ □ b □ ・ □ c □ など検討された入出力データから、システム内に蓄積すべきデータの構造とデータ項目を決定する。

解答群

- (ア) DFD
- (イ) GUI
- (ウ) 外部設計書作成
- (エ) 画面設計/報告書設計
- (オ) コード設計
- (カ) サブシステムの定義と展開
- (キ) ユーザインタフェース
- (ク) 要求仕様の確認
- (ケ) 論理データ設計