

## エンドユーザが理解できるシステム設計書を S E に要求する方法

林田 熙                      飯倉 道雄                      吉岡 亨  
日本工業大学 工学部

システム設計書を渡され、「これに従って開発するので、承認してください」と S E に云われ、内容が十分に理解できないまま、「S E が何とかしてくれるだろう」と開発を承認したが、システム納入後に要求していたものと違うシステムであることが判明して、その対応に苦勞したエンドユーザも多いと思われる。

エンドユーザがシステム設計書を正確に理解する方法として、次の 2 つ考えられる。

エンドユーザが情報処理技術を習得する

エンドユーザが理解できるようなシステム設計書を S E に要求する。

筆者らは、 に対応するために、エンドユーザが理解できるシステム設計書の雛型について検討してきたので報告する。

### A method to ask system engineers for a system specification understandable to end-users.

Hiroshi Hayashida, Michio Iikura and Tohru Yoshioka  
Nippon Institute of Technology

An end-user didn't fully understand the system specifications supplied by the system engineers, and approved it believing that the engineer wrote it with a clear understanding of his/her requirement. But at system delivery, it came out that the system doesn't meet the requested specification. Not a few end-users should have had this kind of trouble.

There are 2 possible solutions for end-users to avoid troubles like this:

solution 1: acquire information-processing technology skills

solution 2: ask their SEs to supply specifications that they can understand

To achieve 2 above, authors of this document have studied a model of system specifications which is understandable to end-users and are going to report it.

## 1. はじめに

情報処理システム（以下、システムと記す）が納入されたものの、機能不足でシステムを廃棄したり、多額の追加費用を負担したエンドユーザの事例は多い<sup>1)</sup>。

このようなことが起こる原因の 1 つは、エンドユーザがシステム設計書の内容を正確に理解できないことにある。

多くの場合、システム設計書は、S E が属する組織が採用した設計技法<sup>2) 3) 4)</sup>による文書をまとめて作っている。システム設計書は、情報処理技術の専門的な用語や図があふれ、役員や管理者を含むエンドユーザは、システム設計書から「要求が反映されているか」「開発費の見積りは妥当か」などを把握することが難しい。

しかし、期待していたシステムが出来上がらなくて最も困るのは、エンドユーザである。開発の原点であるシステム設計書をエンドユーザも理解するように努める必要がある。

エンドユーザがシステム設計書の内容を正確に理解する方法として、次の 2 つが考えられる。

エンドユーザが情報処理技術を習得する  
エンドユーザが理解できるようなシステム設計書を S E に要求する

は、コンピュータ利用の初期の時代、専門性の高い分野、例えば、原子力や気象関連のシステム開発において、S E にその分野の理論を教えてシステムを開発してもらうより、その分野の専門家がプログラム言語を習得してシステム開発を行ったこともあったようだが、現在では、習得する情報処理技術が莫大で、現実的ではない。

は、エンドユーザが理解できるシステム設計書の雛型があれば、雛型に従ってまとめてもらうことを、エンドユーザが S E に要求できるようになり、検討する価値がある<sup>6) 7)</sup>。

本稿は、図面（以下、S D D (System Description Diagram) と記す）を用いた、エンドユーザが理解できるシステム設計書の雛型の提案であり、

- ・ S D D によるシステムの表し方
- ・ S D D によるシステム設計書の雛型
- ・ S D D によるシステム設計書の使い方について述べる。

## 2. SDDによるシステムの表し方

システムを、以下の6つの質問に沿って整理し、その結果を10種類の図面で表す。

- システムの使用者は？  
使用者構成図
- システムの機能は？  
サブシステム遷移図、機能遷移図
- 機能の使い方は？  
機能操作手順図、流れ情報レイアウト図、情報構造図
- システムの蓄積情報は？  
蓄積情報参照図、情報構造図
- 機能の仕組みは？  
機能処理手順図
- 必要なハードウェアとソフトウェアは？  
ハードウェア構成図、ソフトウェア構成図

これらの図面を、次の図書館業務支援システムを例にして示す。

### [例] 図書館業務支援システム

#### システム概要

図書館業務を支援するシステム。主な機能は

- ・ 図書の貸出と返却
- ・ 図書の有無に関する問い合わせ
- ・ 蔵書の管理

である。

#### (1) システムの使用者は？

##### 使用者構成図 (図1参照)

使用者構成図は、システムの利用者、流れ情報の授受の有無、流れ情報を授受する媒体とその数を表している。

使用者とはエンドユーザ、流れ情報はシステムと使用者で授受する情報である。

図1から次のことが分かる。

- ・ 使用者は、「図書館利用者」「窓口係」などである
- ・ 「図書館利用者」は、「図書館業務支援システム」と流れ情報を授受する
- ・ 「図書館利用者」が、「図書館業務支援システム」と流れ情報を授受する媒体は、「検索用端末機」であり、20台ある (他の使用者についても同様)

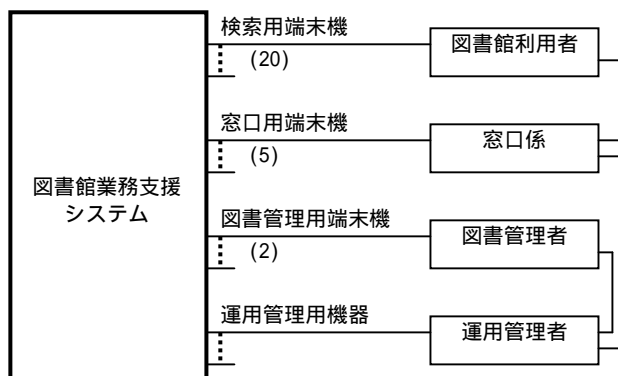


図1 使用者構成図の例

#### (2) システムの機能は？

##### サブシステム遷移図 (図2参照)

サブシステム遷移図は、システムに含まれるサブシステム、サブシステムの使用者と流れ情報を授受する媒体、サブシステムを使う順序と時期を表している。

サブシステムとは、システムの機能を、「だれが」「どの媒体で」「いつ」使うかによって分類した、機能の集まりである。

図2から、次のことが分かる。

- ・ サブシステムは、「初期設定」「運用準備」などである

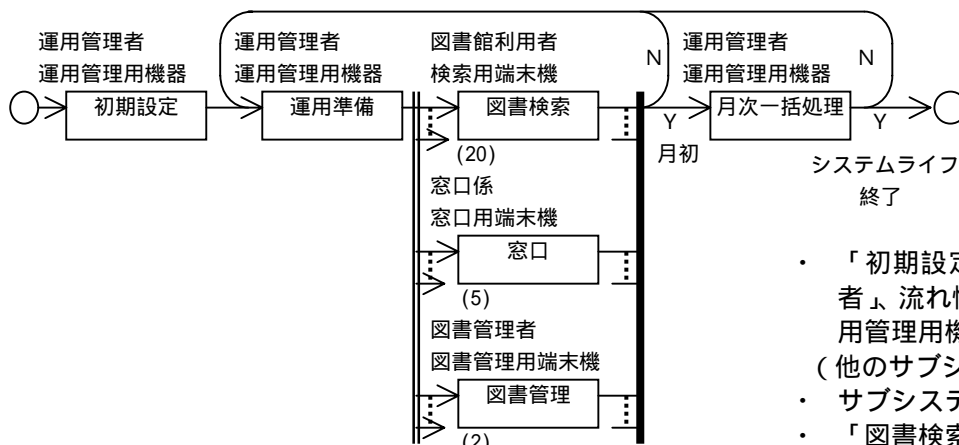


図2 サブシステム遷移図の例

- ・ 「初期設定」の利用者は「運用管理者」、流れ情報を授受する媒体は、「運用管理用機器」である (他のサブシステムについても同様)
- ・ サブシステムは、矢印の順に使える
- ・ 「図書検索」「窓口」「図書管理」は、並行して使える

**機能遷移図** (図3参照)

機能遷移図は、サブシステムに含まれる機能、機能を使う順序を表している。

図3は、図2のサブシステム「図書管理」の機能遷移図で、次のことが分かる。

- ・ 「図書管理」の機能は、「図書管理開始」「機能選択」「購入図書登録」などである
- ・ 機能は、矢印の順に使える

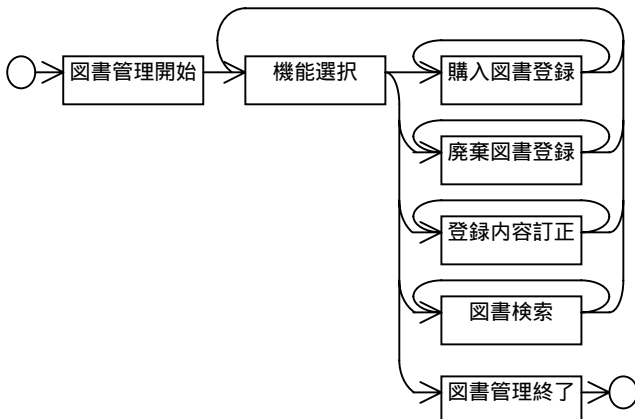


図3 機能遷移図の例

**(3) 機能の使い方は？**

**機能操作手順図** (図4参照)

機能操作手順図は、機能の使用者、システムと授受する流れ情報、流れ情報を授受する順序を表している。

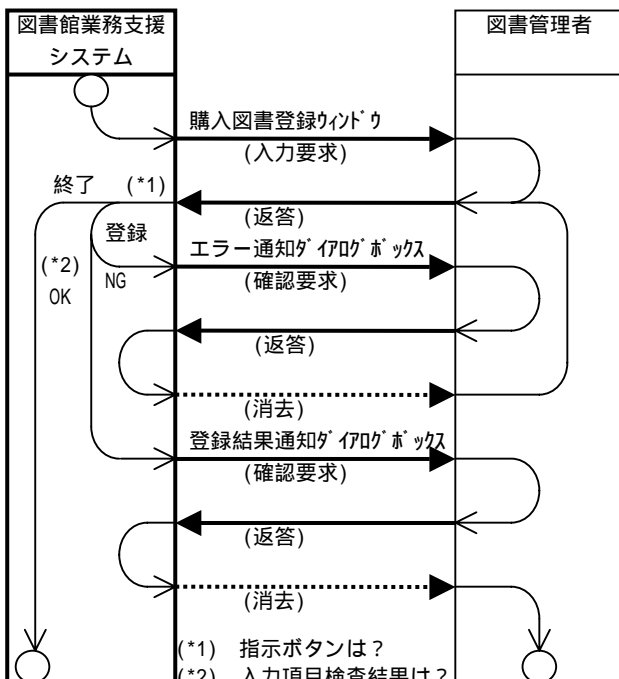


図4 機能操作手順図の例

図4は、図3の機能「購入図書登録」の機能操作手順図で、次のことが分かる。

- ・ 機能の使用者は、「図書管理者」である
- ・ 授受する流れ情報は、「購入図書登録ウィンドウ」「エラー通知ダイアログボックス」などである
- ・ 流れ情報は、矢印の順に授受する

**流れ情報レイアウト図** (図5参照)

流れ情報レイアウト図は、流れ情報の具体的な形式を表している。

図5は、図4の流れ情報「購入図書登録ウィンドウ」の流れ情報レイアウト図で、次のことが分かる。

- ・ 入力項目は、「図書コード」「整理コード」「書名」などである
- ・ 「登録」「終了」ボタンがある

図書館業務支援システム (図書管理) YYMMDD  
 機能名: 購入図書登録  
 登録データを入力してください。

図書コード	<input type="text"/>
整理コード	<input type="text"/>
書名	<input type="text"/>
著者	<input type="text"/>
要約	<input type="text"/>
貸出区分	<input type="text"/>
キーワード	<input type="text"/>

図5 流れ情報レイアウト図の例

**情報構造図** (図6参照)

情報構造図は、流れ情報と蓄積情報のデータ要素の構造をPAD (Problem Analysis Diagram) で表している<sup>5)</sup>。

図6は、図5の流れ情報「購入図書登録ウィンドウ」の情報構造図で、次のことが分かる。

- ・ 「購入図書登録ウィンドウ」は、「入力要求」と「返答」からなる
- ・ 「入力要求」のデータ要素は、「実行年月日」「図書コード」などである(「返答」も同様)
- ・ 「キーワード」は、同じデータ要素が10ある
- ・ 「返答」の「指示ボタン」が、「終了」の場合、「図書コード」などのデータ項目は、存在しない

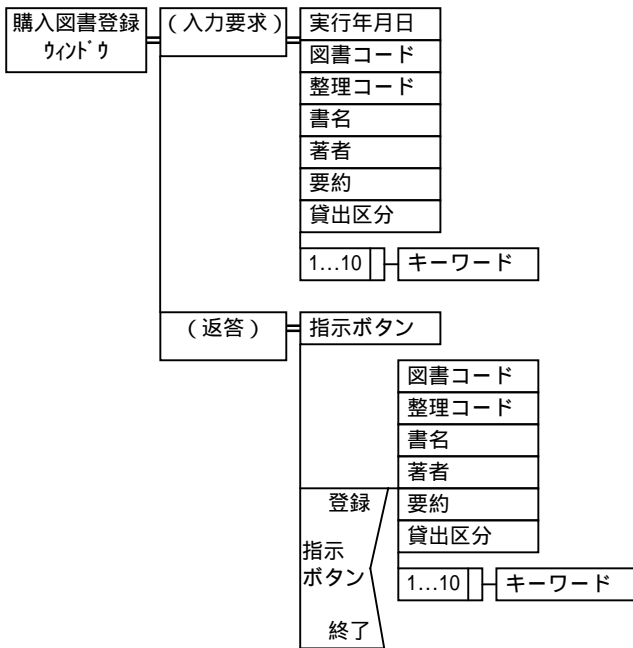


図6 情報構造図の例

#### (4) システムの蓄積情報は？

##### 蓄積情報参照図 (図7参照)

蓄積情報参照図は、システムの蓄積情報、蓄積情報に含まれるデータの最大数、蓄積情報間の参照の有無を表している。

蓄積情報とは、システム内に蓄積している情報である。通常、同一形式の複数のデータからなる。

図7から、次のことが分かる。

- ・ システムの蓄積情報は、「貸出図書」「図書館利用者」などである
- ・ 「貸出図書」のデータ数は、最大 10,000 である

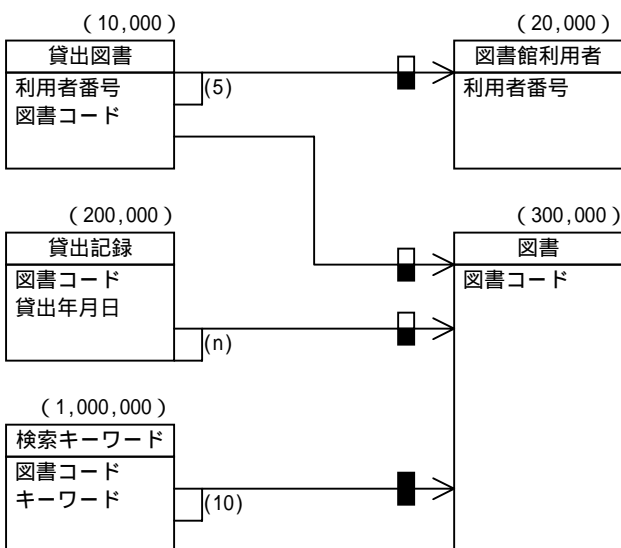


図7 蓄積情報参照図の例

- ・ 「貸出図書」のデータを識別する項目は、「利用者番号」「図書コード」である
- ・ 「貸出図書」から「図書館利用者」を参照できる
- ・ 「図書館利用者」のデータは、最大5つの「貸出図書」のデータから参照される
- ・ 「図書館利用者」のデータには、「貸出図書」のデータから参照されないものがある (他の蓄積情報についても同様)

##### 情報構造図 (図8参照)

図8は、図7の蓄積情報「図書」の情報構造図で、次のことが分かる

- ・ 「図書」は、図書の数だけ同一形式のデータがある
- ・ データ要素は、「図書コード」「整理コード」などである
- ・ 「図書コード」は、「図書」のデータを識別する項目である

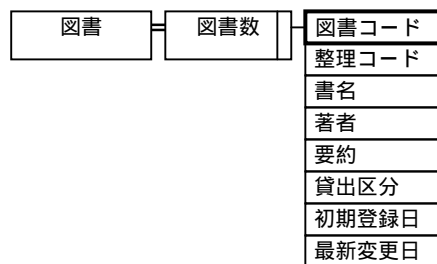


図8 情報構造図の例

#### (5) 機能の仕組みは？

##### 機能処理手順図 (図9参照)

機能処理手順図は、機能を実現するためのシステム側のアルゴリズムをPADで表している。

図9は、図4の「購入図書登録」の機能処理手順図で、次の手順で情報を処理することが分かる。

「購入図書登録ウインドウ」を作成し出力する  
 「購入図書登録ウインドウ」を入力する  
 入力した「指示ボタン」が「登録」なら へ、  
 「終了」なら へ  
 入力データの検査を行い、検査結果がOKなら へ、NGなら へ  
 蓄積情報「図書」と「検索キーワード」にデータを追加し、次に、「登録結果通知ダンプボックス」を作成、出力、入力、消去し、 へ  
 「エラー通知ダンプボックス」を作成、出力、入力、消去する  
 「指示ボタン」=「登録」で検査結果=NGなら へ、それ以外は へ  
 処理を終了する

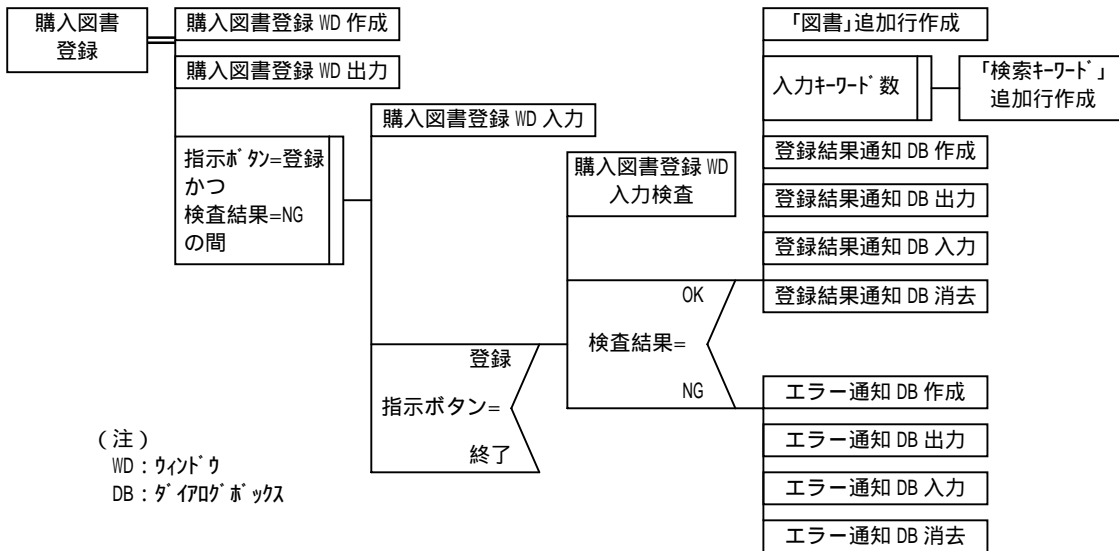


図9 機能処理手順図の例

### (6) 必要なハードウェアとソフトウェアは？

#### ハードウェア構成図 (図10参照)

ハードウェア構成図は、システムに必要なハードウェアとハードウェアの接続を表している。

図10から、次のことが分かる。

- ・ 検索用端末機は、開発するハードウェアである
- ・ 既存のハードウェアは、サーバ、スイッチングハブ、デスクトップPCなどである
- ・ サーバには、プリンタ、スイッチングハブなどが接続している
- ・ スwitchングハブ1には、検索用端末機が20台接続している  
(他のハードウェアについても同様)

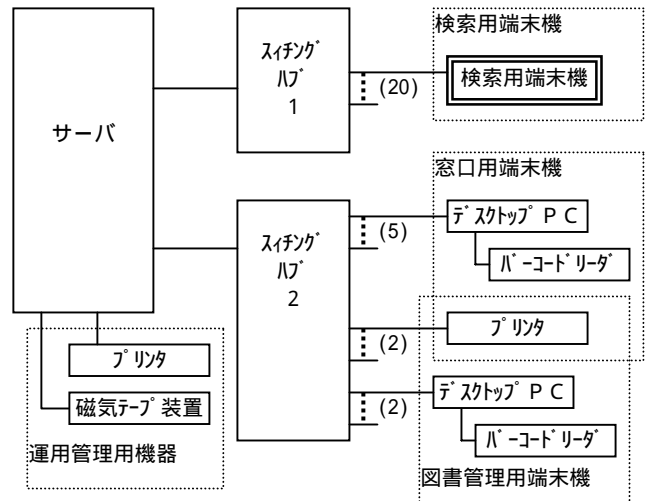


図10 ハードウェア構成図の例

#### ソフトウェア構成図 (図11参照)

ソフトウェア構成図は、システムに必要なソフトウェアとソフトウェア間のインターフェースの有無を表している。

図11は、図10の「サーバ」のソフトウェア構成図で、次のことが分かる。

- ・ 「図書館業務支援」は、開発するソフトウェアである
- ・ 既存のソフトウェアは、オペレーティングシステム、データベース管理システムなどである
- ・ 「図書館業務支援」は、オペレーティングシステムとデータベース管理システムとインターフェースがある  
(他のソフトウェアについても同様)

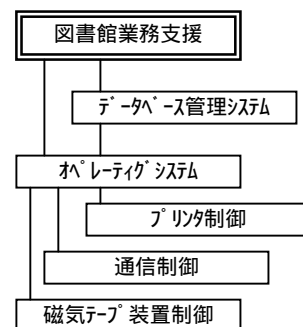


図11 ソフトウェア構成図の例

このように、SDDは10種類の図面でシステムを表している。

### 3 . S D Dの構成

S D Dの構成を図12に示す。

**使用者構成図**は、システムに対し1つある。分散システムのように複数のシステムがあるときも、全体で1つのシステムとみなす。

**サブシステム遷移図**も、システムに対し1つある。サブシステム遷移図に現われる、「使用者」と「流れ情報を授受する媒体」は、使用者構成図のそれぞれと対応している。

**機能遷移図**は、サブシステム遷移図のサブシステムごとに1つある（図の中の 参照）。

**機能操作手順図**、**流れ情報レイアウト図**、**情報構造図**および**機能処理手順図**は、機能遷移図の機能ごとに1組ある（ 参照）。

機能操作手順図に現われる「システム」と「使用者」は、使用者構成図のそれぞれと対応している。

流れ情報レイアウト図とその情報構造図は、機能操作手順図の流れ情報ごとに1組ある（ 参照）。

流れ情報レイアウト図とその構造図は1対1に対応する。

機能処理手順図は、機能操作手順図に現われる「システム」ごとに1つある。（ 参照）。

**蓄積情報参照図**は、システムに対し1つある。

蓄積情報の**情報構造図**は、蓄積情報参照図の蓄積情報ごとに1つある（ 参照）。

**ハードウェア構成図**は、システムに対し1つある。

**ソフトウェア構成図**は、ソフトウェアが必要なハードウェアごと1つにある（ 参照）。

S D Dは、このような構成になっている。

S D Dによるシステムの表し方は、以下の特長がある。

- 図面の表す内容を常識で理解できる
- 表すシステムに制限がない
- 規模の大きいシステムに対応できる
- 内容が必要最小限である
- 内容を参照しやすい

は、システムの表し方に、特に難しい概念もないので、図面の表す内容を、システム開発にかかわりのある人々が、常識で理解できる。

エンドユーザも、図面から「要求が反映されているか」「どのように操作するのか」などを、かなり正確に判断できる。また、文字ではなく図面なので、外国人技術者も内容を理解できる。

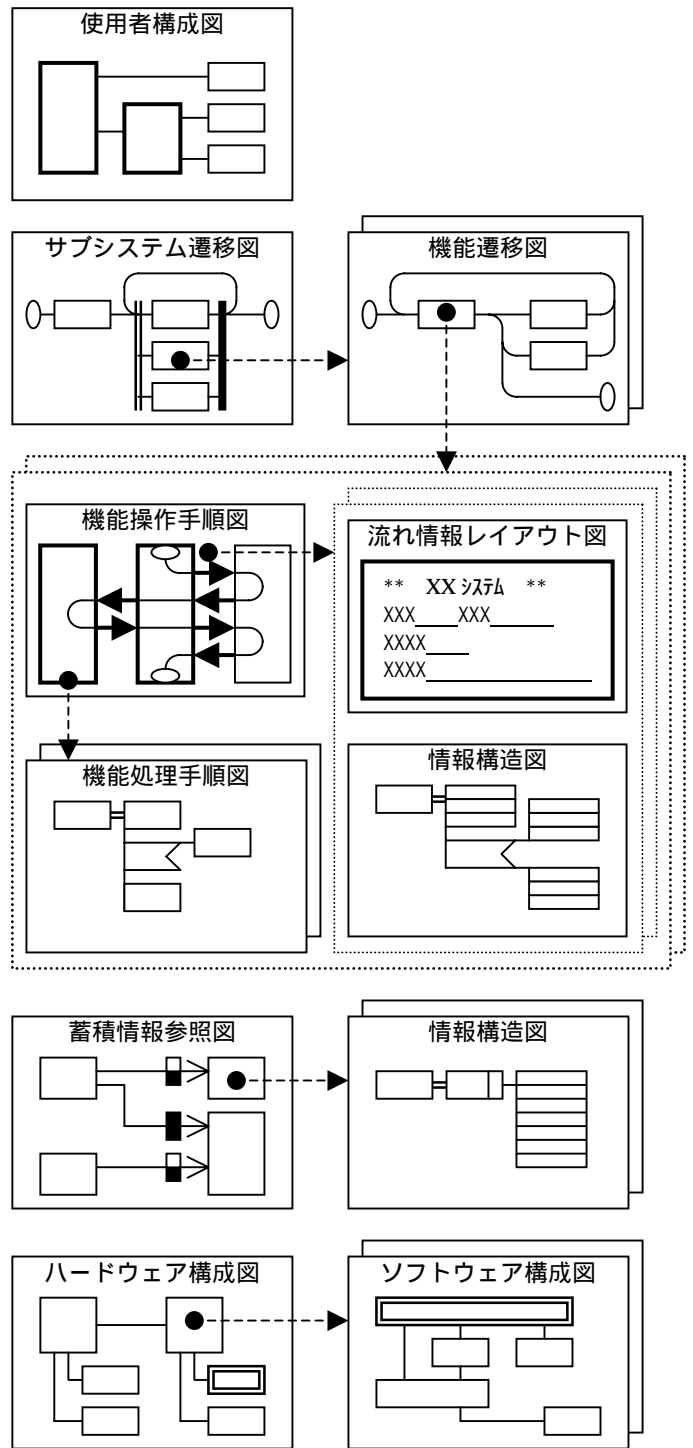


図12 図面の構成

は、どのシステムにも共通する6つの質問に沿ってシステムを表したので、表すシステムに制限はない。事務処理システム、通信システム、組み込みシステムなど、どんなシステムでもこの方法で表せる。

は、システムの機能を「だれが」「どの媒体で」「いつ」使うかによって分類し、システムを細分化することで対応している。

は、図面の構成から、内容の重複している図面がないこと、および図面が1つでも欠ければシステムを表せないことから、必要最小限と考えられる。

は、サブシステム遷移図、機能遷移図、蓄積情報参照図およびハードウェア構成図が、詳細内容のインデックスの役割を果たし、内容を参照しやすくしている。

#### 4. SDDによるシステム設計書の雛型

SDDで表したシステムを、図13の目次に従ってまとめ、システム設計書とする。

「1. 基本構成」は、使用者構成図、サブシステム遷移図、機能遷移図、および蓄積情報参照図を集める。これらの図面で、システムの概要がわかる。

「2. 稼働環境」は、ハードウェア構成図とソフトウェア構成図を集める。これらの図面で、システムに必要なハードウェアとソフトウェアがわかる。

「3. 蓄積情報」は、蓄積情報の情報構造図をすべて集める。蓄積情報の詳細がわかる。

「4. 機能」は、機能操作手順図、流れ情報レイアウト図、流れ情報の情報構造図、および機能処理手順図を集め、サブシステム・機能単位にまとめる。これらの図面で、機能の詳細がわかる。

このように、システム設計書の雛型を定めることにより、エンドユーザは、SEに、

- ・ SDDによりシステムを表す
- ・ この目次に従ってシステム設計書を作成することを要求できるようになる。

また、システム設計書の目次と使用する図面が決まっているので、エンドユーザは、

- ・ 目次の項目ごとに、例えば、「1.基本構成」の図面ができたとき、SEと設計内容のレビューができ、要求が反映されているか、事前に確認できる
- ・ 図面の作成状況でシステム設計の進捗が把握できるようになる。

#### 5. SDDによるシステム設計書の使い方

システム設計書の承認において、エンドユーザが確認しなければならない、重要な事項

- ・ 設計結果は要求を反映しているか？
- ・ 開発費の見積りは妥当か？

目次	図面
1. 基本構成	
1.1 使用者	使用者構成図
1.2 機能	
1.2.1 サブシステム遷移	サブシステム遷移図
1.2.2 機能遷移	
1.2.2.1 サブシステム - 1	機能遷移図
:	
1.3 蓄積情報	蓄積情報参照図
2. 稼働環境	
2.1 ハードウェア構成	ハードウェア構成図
2.2 ソフトウェア構成	
2.2.1 ハードウェア - 1	ソフトウェア構成図
:	
3. 蓄積情報	
3.1 蓄積情報 - 1	情報構造図
:	
4. 機能	
4.1 サブシステム - 1	
4.1.1 機能 - 1 1	
4.1.1.1 操作手順	機能操作手順図
4.1.1.2 流れ情報	流れ情報レイアウト図
4.1.1.3 処理手順	情報構造図
:	機能処理手順図
:	
5. 索引	

図13 システム設計書の目次と図面

における SDD によるシステム設計書の使い方を、「図書館業務支援システム」で示した図面を用いて述べる。

#### (1) 設計結果は要求を反映しているか？

要求の反映は、目次「1. 基本構成」と「4. 機能」の使用者構成図、サブシステム遷移図、機能遷移図、機能操作手順図および流れ情報レイアウト図を使って、次の手順で確認する。

使用者構成図(図1)から、エンドユーザである使用者、すなわち、「図書館利用者」「窓口係」「図書管理者」「運用管理者」の代表を要求反映の確認者として選ぶ。「図書館利用者」のように外部の人の参加が得られないときは、図書館の職員が代行する。

各代表者は、サブシステム遷移図(図2)から、自分が担当するサブシステムを確認する。「図書管理者」代表の担当は、「図書管理」サブシステムである。

各代表者は、担当するサブシステムの機能遷移図(図3)から、担当する機能を確認し、機能の過不足を判断する。「図書管理者」代表は、担当する機能が「購入図書登録」「廃棄図書登録」など7つであることを確認する各代表者は、担当する機能ごとに機能操作手順図(図4)、流れ情報レイアウト図(図5)を使って、機能の使い方をシミュレートし、使い勝手を予想する。

ここまでの確認で、要求を反映していない点があれば、SEに再設計を依頼する。一方、不都合な点がなければ、「機能が欠けている」、「使い勝手が悪い」などは起こらないであろう。

## (2) 開発費の見積りは妥当か

開発費の、主な見積り項目である、

- ・ ハードウェア購入費
- ・ ソフトウェア購入費
- ・ ソフトウェア開発費

の妥当性は、次の手順で確認する。

ハードウェア購入費は、目次「2.稼働環境」のハードウェア構成図(図10)の各ハードウェアに対して、単価と数量が見積もられていることを確認する。図10の場合、サーバは1台、検索用端末機は20台など、ハードウェア構成図と見積りを対応させて確認する。

ソフトウェア購入費も、目次「2.稼働環境」のソフトウェア構成図(図11)から、ハードウェアと同様に、単価と数量が見積もられていることを確認する。図11の場合、開発するソフトウェアである「図書館業務支援」以外のソフトウェアの単価と数量が見積もられていることを確認する。

ソフトウェア開発費は、目次「4.機能」のサブシステムごとに、機能遷移図(図3)のすべての機能が、見積もられていることを確認する。機能ごとの見積りの差は、機能処理手順図(図9)から機能の複雑さの程度を確認する。

この手順で、見積り対象とその数量が妥当か確認できる。金額の妥当性は、価格表を取り寄せ確認する。見積り額がエンドユーザの予算を超えたときは、必須でないハードウェア、ソフトウェア、機能を図面で探し、再見積りを求めることもできる。また、SDDによるシステム設計書で、複数の業者から見積りを取り、比較することも可能である。

## 6. おわりに

よいシステムを作るには、エンドユーザが、システム設計に積極的に参加することが不可欠である。システム化する仕事について詳しいのは、SEではなく、エンドユーザである。当然、できあがったシステムを使うのもエンドユーザである。また、期待していたシステムが出来上がらなくて、最も困るのもエンドユーザである。

エンドユーザが、システム設計に積極的に参加

するためには、参加した結果をまとめたシステム設計書をエンドユーザが理解できなければならない。

SDDによるシステム設計書は、「エンドユーザが理解できるシステム設計書」であると筆者らは考えている。

SDDによるシステム設計書があれば、エンドユーザは、SEとシステムについて対等に議論ができる。この議論を通じ、「必要な機能が欠けている」「使い勝手が悪い」などが少なくなることが期待できる。

筆者らは、SDDを用いた

- ・ システム設計書の作り方
- と、SDDによるシステム設計書を用いた
- ・ ソフトウェア設計書の作り方
  - ・ システム検査仕様書の作り方
  - ・ システムの操作マニュアルの作り方
  - ・ システム製造の進捗管理法
- などについて、研究、教育、普及を進めている。

## 参考文献

- 1) 日経コンピュータ編：動かないコンピュータ - 情報システムに見る失敗の研究、日経BP社(2002)
- 2) Edward Yourdon(黒田純一郎、渡部研一訳)：構造化手法によるソフトウェア開発、日経マグローヒル社(1987)
- 3) 吉田裕之、山本里枝子、上原忠弘、田中達雄：UMLによるオブジェクト指向開発実践ガイド、技術評論社(1999)
- 4) 増永良文：リレーショナルデータベース入門：サイエンス社(1991)
- 5) 二村良彦：プログラミング技法 - PADによる構造化プログラミング、オーム社(1984)
- 6) 林田熙、福田民生、飯倉道雄：図面による情報システム設計書の作成法、教育システム情報学会27回大会、no. T5-1、pp173-174、Aug. 2002
- 7) 林田熙、飯倉道雄：エンドユーザにも理解できるシステムの表し方、教育システム情報学会研究報告、Vol.19, No.1, pp.63-70(2004)