

構造モデルによるシステムプログラム外部仕様の レビュ手法[†]

田代 勤^{††} 春名 宏一^{††}

プログラムの外部仕様の正しさは、そのプログラムを使用するユーザの目的を、矛盾なく十分達成し得るという視点から保証されねばならない。これに対し、現在、システムプログラム外部仕様の正しさの保証は、仕様書そのものを徹底的にレビューするという属人的かつ管理に依存した方法で行われている。仕様書そのものによるレビューでは、ユーザの目的とこれを達成する機能との関係、機能と入出力の関係等からなる設計項目相互の関連は Explicit にはておらず、関連の有無の検討、分析はレビューに大部分を依存しており、レビューの能力により、その精度が左右される。

本論文では、構造モデルを基本にした、レビュー手法を提案する。まず、ユーザの目的を含めて、仕様をレビューするという観点から、システムプログラムの各種設計項目を明らかにし、項目間の関連を構造的に記述する基本モデルを提案する。このモデルを基本に、項目間関連の不整合な点のチェック、項目意味にまで立入った上でのユーザ目的から見た仕様の妥当性チェックをそれぞれ目的に、項目間関連をレビューにビジュアルに認識させる2種類の図式表現を提案する。また、これらの表現を用いたレビューの具体手順を示す。

従来手法でレビューを終えている仕様書数件について、提案手法を適用し、従来手法では摘出されなかった新たな不良がかなり摘出されることを確認している。

1. まえがき

システムプログラムの欠陥は、他の多くのプログラムに影響を及ぼすため、その信頼性向上は重要な課題となっている。信頼性向上のためには、外部仕様設計、プログラム設計、コーディング、テスト・デバッグ、運用・保守といった各工程で不良を発生させないことが必要である。その中でも、外部仕様設計等の初期工程を正確に実行することが、後工程で不良が発見された場合の修正工数の大きさ等を考えた場合、特に重要である。

外部仕様の正しさは、そのプログラムを使用するユーザの目的を、矛盾なく十分達成し得るという視点から保証されねばならない。従来、外部仕様を誤りなく定義するための方法として PSL/PSA¹⁾、SREM²⁾、SADT³⁾ 等の要求仕様定義技術が提案されている。しかし、PSL/PSA、SREM では、開発すべきシステムそのものを記述し、仕様間の不整合を自動検証することに力点が置かれており、ユーザの目的と仕様との関係は取り扱わない。また、SADT では、ユーザの目的も記述はするが、ユーザの目的と仕様間の関係は、

明確には取り扱っていない。正確な外部仕様記述を目的に利用される、ディシジョンテーブル⁴⁾、原因結果グラフは⁵⁾、外部仕様を分りやすく表現するのに十分な記述力を有していないため、外部仕様がユーザの目的を達成し得るという正しさを保証する上で満足の行くものではない。

このような状況において、現在、システムプログラム外部仕様の正しさの保証は、仕様書そのものを徹底的にレビューするという属人的かつ管理に依存した方法で行われている。これに対しては、従来、レビューの構成、レビューの進行方法等、管理の面からレビューの効果を向上させる方法について提案がある⁶⁾。しかし、仕様書そのものによるレビューでは、ユーザの目的とこれを達成する機能との関係、機能と入出力の関係等からなる設計項目相互の関連が Explicit にはておらず、関連の有無の検討、分析はレビューに大部分を依存しており、レビューの能力により、その精度が左右される。

本論文では、構造モデル* を基本にしたレビュー手法を提案する。まず、ユーザの目的を含めて仕様をレビューするという観点から、システムプログラムの各種設計項目を明らかにし、項目間の関連を構造的に記述する基本モデル（以下仕様記述構造モデルと呼ぶ）を提

* 対象の特徴をいくつかの要素集合とその集合の要素間の関係によって捉えるモデル。

[†] A Review Method of System Program's Functional Specifications using Structural Model by TSUTOMU TASHIRO and KOICHI HARUNA (Systems Development Laboratory, Hitachi Ltd.).

^{††} (株)日立製作所システム開発研究所

案する。このモデルを基本に、項目間関連の不整合な点のチェック、および項目意味にまで立ち入った上のユーザ目的から見た仕様の妥当性チェックを目的に、項目間関連をレビュアにビジュアルに認識させる2種類の図式表現を提案する。また、これらの表現を用いたレビューの具体手順を示す。以上の、仕様記述構造モデルとその表現、手順によれば、仕様書を直接用い、各レビュアなりに項目間関連を頭の中だけで構築、検討する方法に比べ、レビュアの能力に依存せず、レビューの効率、精度を向上できる。

2. システムプログラム外部仕様レビューへの構造的アプローチ

ユーザがシステムプログラムを利用するための手段(ユーザマクロ、制御ブロック等のこと)で、以下、インターフェースと呼ぶ)の仕様は、非常に複雑化している。たとえば、制御ブロックは、マクロのオプション機能の指定やバッファアドレス、リターンコード等の入出力(マクロ、そのオペランドといったインターフェースと区別し、これら制御情報を入出力と呼ぶ)の指定、取得等に用いられる。この際、制御ブロックの個々のフィールドの働きとマクロの関係が1対1でない、一つのマクロに複数の制御ブロックが関係する等、マクロ、オプション機能、入出力、制御ブロックのフィールドといった、各種設計項目相互の関連が複雑化している。現在、仕様をレビューするという観点から、このような関連をどのように捉えるべきかということは、各レビュアに完全に依存している。また、仕様書を直接用いる現在のレビュー方法では、仕様書には設計項目相互の関連がExplicitにはでていないため、関連の有無の検討、分析の大部分がレビュアに依存している。このため、レビュアの経験、能力が、レビューの質を大きく左右する。

本論文では、まず、ユーザの立場から仕様をレビューするという観点と、ユーザ要求はユーザから直接見えるインターフェースの仕様に反映されるという原則に基づき、インターフェースを中心に設計項目相互の関連を構造モデルによって記述する。すなわち、ユーザマクロを中心に、その目的、機能、入出力、これらに付随するその他のパラメタといった設計項目の関連を正確に定義する仕様記述構造モデルを提案し、レビュア個人に依存しない、一つのレビューの枠組を与える。この枠組の上で、外部仕様のレビューを次の二つの目的を持って行う。

(1) 設計された仕様の不整合な点、すなわち、設計項目、項目間関係の抜け、誤り等の不良を抜けなく摘出する。

(2) さらにその上で、設計された仕様が妥当かどうか、すなわち、ユーザの目的から見て、機能、入出力、インターフェースといった設計項目に、欠落、矛盾等の不良がないかを総合的に判定する。

このために、提案した仕様記述構造モデルを基本とし、設計項目間の関係を、上記二つの分析の目的に応じ、レビュアにビジュアルに認識させる図式表現を導入する。(1)のレビューに対し、相互に関連する複数のマトリクスを用いた図式表現を導入する。これは、個別の設計項目間の関係をもれなく、正確にレビュアが把握、分析できることを狙ったものである。次に、(2)のレビューに対し、仕様記述構造モデルの樹木状表現を導入する。これは、ユーザの目的の体系を最上位に、以下、個々の設計項目間関係に従い、設計項目を下位に展開した図形である。本表現は、設計項目の全体系を、項目の内容を含め理解容易な形に視覚化し、項目の意味にまで立ち入った仕様の妥当性チェックの促進を狙うものである。以上の仕様記述構造モデルとその表現を利用することにより、外部仕様レビューの質を、レビュアの能力に依存せずに向上できる。

3. システムプログラム外部仕様の構造モデルによる記述(仕様記述構造モデル)

システムプログラムでは、機能、インターフェースが他プログラムに比べ、非常に多種、複雑化している。機能は、主に、ユーザマクロにて提供されるが、マクロは、ユーザのさまざまな目的に対し利用できるよう、多くのオプション機能を有する設計とされる。ここで、オプション機能は、マクロのオペランド値で指定する他に、ユーザプログラム実行時に実行機能を種種に変更したいという理由から、レジスタ値にて指定する方法と、別にいくつかの制御ブロックを設け、このフィールド値として指定する方法がある。同様に、ユーザとシステムプログラム間の入出力も、マクロのオペランドの他に、レジスタ、制御ブロック等を介して行われる。また、機能を提供するインターフェースには、マクロの他に、出口ルーチンといった形態のものも存在する。レビューの際には、これらのシステムプログラム外部仕様を構成する種々の項目を次の観点から把握せねばならない。

(1) ユーザマクロには何があり、どのような機能

表 1 仕様記述構造モデルの要素集合

Table 1 Element Sets of a structural model which describes system program's functional specifications.

	記号	集 合 名	定 義	備 考
目的	A	目的項目集合	プログラムの果たすべき役割を表わす項目の集合	通信セッションの管理、データフローの管理等
機能	B	機能項目集合	個々の機能を表わす項目の集合	機能内容の詳細さの程度はユーザインターフェースのパラメタを説明するために必要な程度
入出力情報	C	入力情報項目集合	ユーザから入力する情報項目の集合	ユーザ指定バッファアドレス、送信メッセージ長等(下記Fにはない)
	D	出力情報項目集合	ユーザに対し出力する情報項目の集合	マクロのリターン情報、受信メッセージ長等
機能インサブ	E	機能利用インターフェース項目集合	ユーザが機能を利用するきっかけとなるインターフェース項目の集合	ユーザマクロ、JCL、出口ルーチン等
	F	利用機能詳細指定パラメタ項目集合	ユーザが上記Eにて、機能を利用する際、利用する機能の詳細を指定するパラメタ項目の集合	ユーザマクロのオペランド値、制御ブロックのフィールド値等
入出力インターフェース	G	入力情報インターフェース項目集合	ユーザが入力情報(C)を与えるためのインターフェース項目の集合	ユーザマクロのオペランド、制御ブロックのフィールド等
	H	出力情報インターフェース項目集合	ユーザが出力情報(D)を得るためのインターフェース項目の集合	制御ブロックのフィールド、レジスタ等

を持ち、どのようなユーザの目的に対し利用されるのか。

(2) ユーザマクロはどのように利用するのか(オプション機能の指定方法、入出力の指定、取得方法)。

(3) システムプログラムとユーザ間の入出力には何があり、どのような機能と関係するのか。

(4) ユーザマクロ以外にどのようなインターフェースがあり、どのような機能を持ち、どのようなユーザの目的に対し利用できるのか。

(5) ユーザマクロ以外のインターフェースはどのように利用するのか。

以上の分析を満足させる、設計項目相互の関連を構造的に記述する仕様記述構造モデルを以下のように提案する。

モデルを構成する要素は、表1に定義する集合に区分、整理できる。ユーザがシステムプログラムの種々の機能を利用する各目的に対応させ目的項目集合(A)を、各機能に対応させ機能項目集合(B)を定義する。さらに、入出力に対応させ、それぞれ、入力情報項目集合(C)、出力情報項目集合(D)を定義する。また、マクロ、出口ルーチン等ユーザに機能を提供するインターフェースに対し機能利用インターフェース項目集合(E)を、そのオプション機能の指定方法に対し利用機能詳細指定パラメタ項目集合(F)を定義する。入出力インターフェースに対しては、入力情報インターフェース項目集合(G)、出力情報インターフェース項目集合(H)を定義する。

モデルを構成する要素間の関係は、以下に定義する

7つの関係($R_1 \sim R_7$)に整理できる。

(1) 目的どうしの上下関係

$R_1: A \times A, R_1 \ni (a_1, a_2) \leftrightarrow$ 目的 a_1 のサブ目的は目的 a_2 である。

(2) 目的と機能利用インターフェース、その機能の関係

$R_2: A \times E \times B, R_2 \ni (a, e, b) \leftrightarrow$ 目的 a のために機能利用インターフェース e により機能 b を利用する。

(3) 機能と入力の関係

$R_3: B \times C, R_3 \ni (b, c) \leftrightarrow$ 機能 b 利用の際に入力情報 c を与える。

(4) 機能と出力の関係

$R_4: B \times D, R_4 \ni (b, d) \leftrightarrow$ 利用者は機能 b により出力情報 d を得る。

(5) オプション機能とその指定方法の関係

$R_5: F \times B, R_5 \ni (f, b) \leftrightarrow$ 利用機能詳細指定パラメタ f により機能 b の実行を指定する。

(6) 入力の指定方法を表わす関係

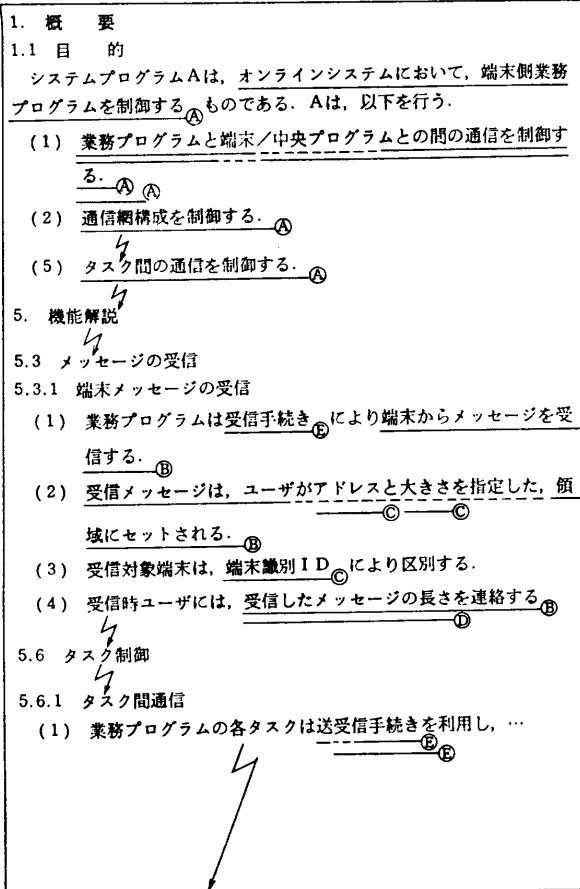
$R_6: G \times C, R_6 \ni (g, c) \leftrightarrow$ 入力情報インターフェース g により入力情報 c を与える。

(7) 出力の取得方法を表わす関係

$R_7: H \times D, R_7 \ni (h, d) \leftrightarrow$ 出力情報インターフェース h により出力情報 d を得る。

以上定義した、仕様記述構造モデル要素の実際の仕様書に記述された例を図1に示す(集合の記号を付加してある)。

機能解説書



マクロ文法書

1. 記述規則
6. READ マクロ

(1) 機能概要
メッセージの受信を行う。
⑪

(2) 書き方

ラベル	命 令	オペラ ンド
【記号】	READ ⑫	RQCB={リクエストブロック名}, MTYPE={U T}

(3) オペラ ンド

(a) RQCB=<記号名称>… 要求の詳細を指定する RQCB の名称を指定する。

(b) MTYPE=U… 端末からのメッセージ受信を指定, T… タスクからのメッセージ受信を指定
⑬ ⑭

(4) マクロインタフェース

レジスタ	リターン情報
GRO ⑮	以下の受信結果が通知される。 ⑯ 正常終了 “0000”, 异常終了 “0010” ⑰

16. RQCB の形式
送受信要求時ユーザは以下の形式の RQCB を用意せねばならない。

フィールド名	位置	長さ	内 容
RQSSID ⑯	0(バイト)	4(バイト)	相手先識別 ID ⑰
RQBFAREA ⑱	4	4	バッファアドレス ⑲
	⑳	㉑	㉒
RQINFLLEN ㉓	62	4	受信メッセージ長 ㉔
RQBFFLEN ㉕	66	4	受信領域サイズ ㉖

図 1 仕様記述構造モデルの要素例

Fig. 1 Examples of elements for a structural model which describes system program's functional specifications.

4. 仕様記述構造モデルを中心としたシステムプログラム外部仕様の図式表現

導入した仕様記述構造モデルを基本に、レビュー者がビジュアルに設計項目間の関係を認識するために用いる、2種類の図式表現を提案する。

4.1 関連マトリクス

外部仕様のレビューでは、まず、設計された仕様の不整合な点を完全に摘出することが必要である。すなわち、仕様記述構造モデルで定義した、設計項目、項目間関係に、抜け、誤り等の不良がないかを、全項目に渡り網羅的に追求できることが必要である。この目的のために、相互に関連する複数のマトリクス（以下、関連マトリクスと呼ぶ）を用いて、モデルを表現する。マトリクスの持つ、(1)網羅性、(2)相互関係表現の正確性から、項目、項目間関係の抜け、誤り等のチェックを確実に行うことができる。

関連マトリクスを図2に示す。マトリクスの各軸は、各々の設計項目集合を表わしており、交点は、項目間関係の有無を表わしている。なお、目的項目、機能利用インターフェース項目、機能項目間の3項関係を表わすために、目的項目集合(A)と機能利用インターフェース項目集合(E)の要素を対にして並べた軸を用いている。目的項目どうしの関係(R_1)は、4.2の図式表現上で扱う。

関連マトリクスを用い、仕様を次のように整理する。

- (1) 仕様として、項目間に関係のあることが明確になっている・・・交点に○をつける。
 - (2) 仕様として、項目間に関係のないことが明確になっている・・・交点に×をつける。
 - (3) 他はブランク
- この整理結果を用いて、次の点に焦点を合わせてチェックを行う。

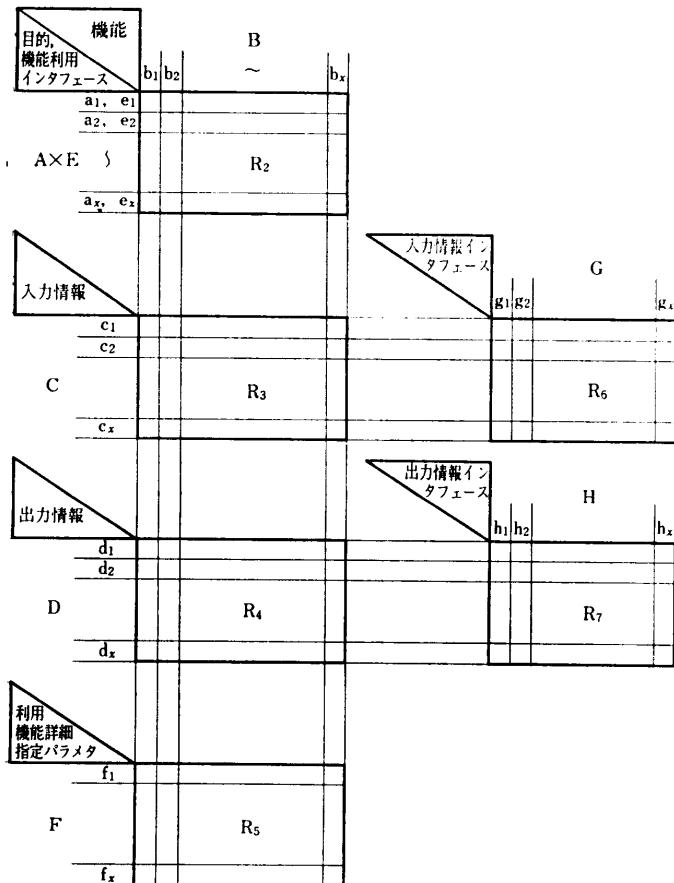


図 2 関連マトリクス
Fig. 2 Related matrix.

(1) 交点に○が全くつかない項目について、(a)その項目と関係すべき項目が抜けていないか、(b)他の項目と関係あることが仕様として不明確ではないかをチェックする。

(2) 交点に○かつ×がついた関係について、関係の有無を示す仕様に矛盾がないかをチェックする。

関連マトリクスでは、これらのチェックを多元的に行えるため、チェックの信頼性を向上できる。たとえば、入力情報に関しては、機能から見た場合、入力情報インターフェースから見た場合と2重のチェックを行える。

4.2 機能説明樹木

外部仕様の妥当性、すなわち、ユーザの目的から見て、機能、入出力、インターフェースといった設計項目に欠落、矛盾がないか等の総合的判定を行うには、ユーザの目的及びこれに対し設定された設計項目との対応の体系を十分把握せねばならない。

ここでは、モデルを樹木状に表現する機能説明樹木

を提案する。これは、樹木図の持つ、(1)一つの項目がどのように他の項目に展開されているのかを表現するのに適する、(2)枝をたどることにより、個々の項目が、他の項目とどのように関係しているのかを容易に把握できるという特徴を利用し、仕様全体の体系的把握を促進するものである。機能説明樹木の全体構造を図3に示す。さらに、図1の仕様に対応する機能説明樹木の例を図4に示す。設計項目は、樹木のノードとして表現する。項目間関係は、ノードを結ぶアーチとして表現する。機能説明樹木では、まず、構造モデルで定義した目的項目間の関係(R_1)を最上位に展開し、ユーザの目的の体系を明確にしている。さらに、これに、機能利用インターフェース項目、機能項目、入出力情報項目の順で、3章にて定義した関係、 R_2, R_3, R_4 を結合し表現することにより、個々の目的に対応する設計項目の体系を明らかにする。また、利用機能詳細指定パラメタ、入出力情報インターフェースを、3章にて定義した関係、 R_5, R_6, R_7 に従い、それぞれ、機能項目、入出力項目の上に配置し、目的達成のためにどのようなインターフェースが用意されているのかを明らかにしている。なお、図4に示すように、機能項目をいくつかまとめた内容の項目を適宜追加することにより、機能体系をより理解しやすいものとしている。

機能説明樹木を利用することにより、(1)外部仕様全体を見通した総合的判断が可能となり、さらに、(2)ユーザの目的ごとの議論の範囲が明確になり、議論の発散、横道へのそれを防止することができる。このことより、設計項目を目的と対応させ総合的に吟味

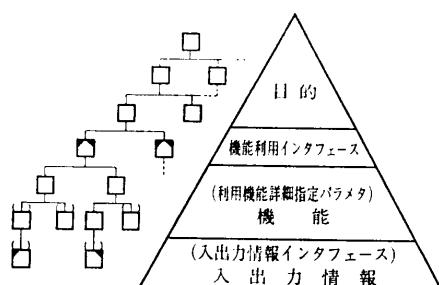


図 3 機能説明樹木の構造
Fig. 3 Structure of functions descriptive tree.

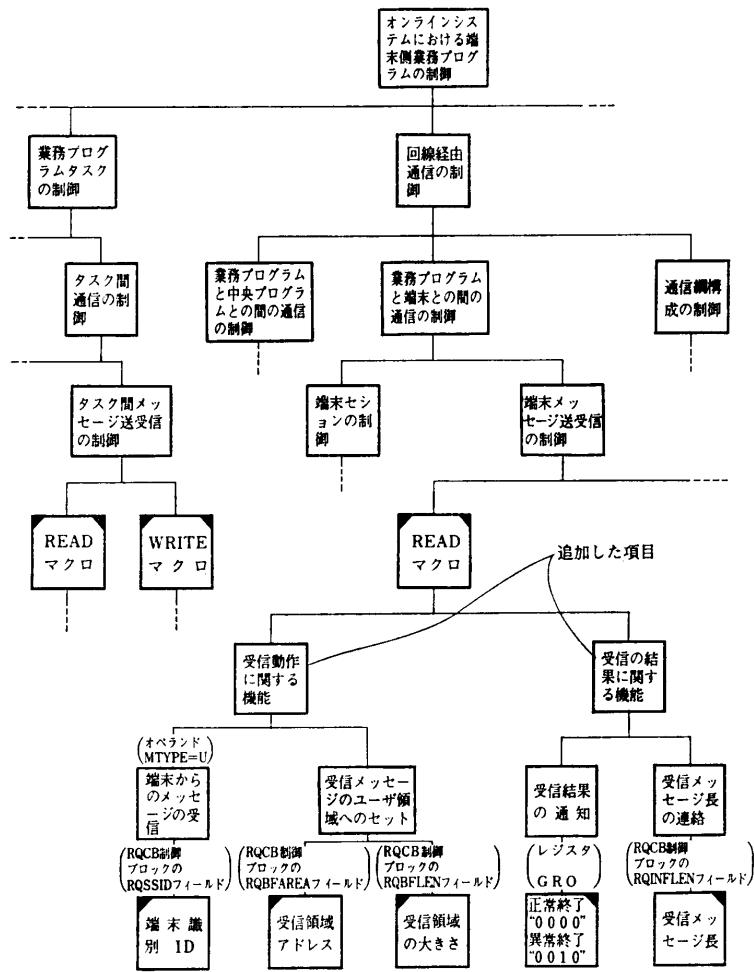


図 4 機能説明樹木例

Fig. 4 An example of functions descriptive tree.

することを、効率的かつ確実に行なうことが可能となる。

5. システムプログラム外部仕様のレビュー

5.1 レビュ手順

前章の2種類の図式表現を用いる、システムプログラム外部仕様のレビュ手順を提案する。提案手順にて、チェックするチェック項目を図5に示す(以下、チェック項目に対して本図の番号を用いる)。ここにあげた項目は、外部仕様レビュにおいて最も基本的かつ重要なものである。レビュ手順を図6に示す。以下、これに沿って各作業の内容を説明する。

(1) 設計項目の抽出、分類

システムプログラムの外部仕様を記述する資料(たとえば、機能解説書、マクロ文法書等の仕様書)から、設計項目を抽出し、設計項目集合ごとに分類す

る。この際、チェック項目④、⑥をチェックする。

(2) ユーザ目的の明確化

機能説明樹木の最上位の構造を明らかにする。すなわち、ユーザの目的体系及び最下位目的に対し用意されている機能利用インターフェース(3章で定義した関係 R_2 の $A \times E$ の部分)を明らかにする。外部仕様レビュのためには、ユーザの目的が明確でなければならないが、現実は、目的が十分仕様として記述されていることは少ない。設計者自身も明確に意識しないまま設計が進んでしまうことが多い。本段階で、ユーザの立場から、できればユーザも交じて討議を行い、目的体系を十分明らかにしておく必要がある。次の作業からなる。

(a) 目的項目を、3章で定義した関係 R_1 に従い樹木状に並べる。

(b) いくつかの項目をまとめた内容の上位目的項目、一つの項目を詳細に展開した下位目的項目を追加し、目的体系を明確にする。

(c) 樹木の最下位の目的項目に対応する機能利用インターフェース項目を明らかにする。

本段階のチェック項目は⑩である。

(3) 関連マトリクスによる項目間関係の整理

分類した各設計項目間の関係を関連マトリクスを用いて整理する。この際、4.1に述べた方法で、チェック項目①、②、⑦、⑪、⑫、⑯をチェックする。なお、これらの項目以外は、必ずしも対応相手の存在する必要のない設計項目で(入力情報を必要としない機能もある)対応相手がないという点での不良チェックはできない。次の段階での総合的判断が必要である。

(4) 機能説明樹木上のレビュ

機能説明樹木を用い、総合的判定を行う。次の作業からなる。

(a) 作成した目的体系((2))の下に、関連マトリクスの整理結果を図3の構造に従い結合し、樹木図化する。

(b) 機能項目をいくつかまとめた内容の項目を適宜追加し、機能体系を明らかにする。

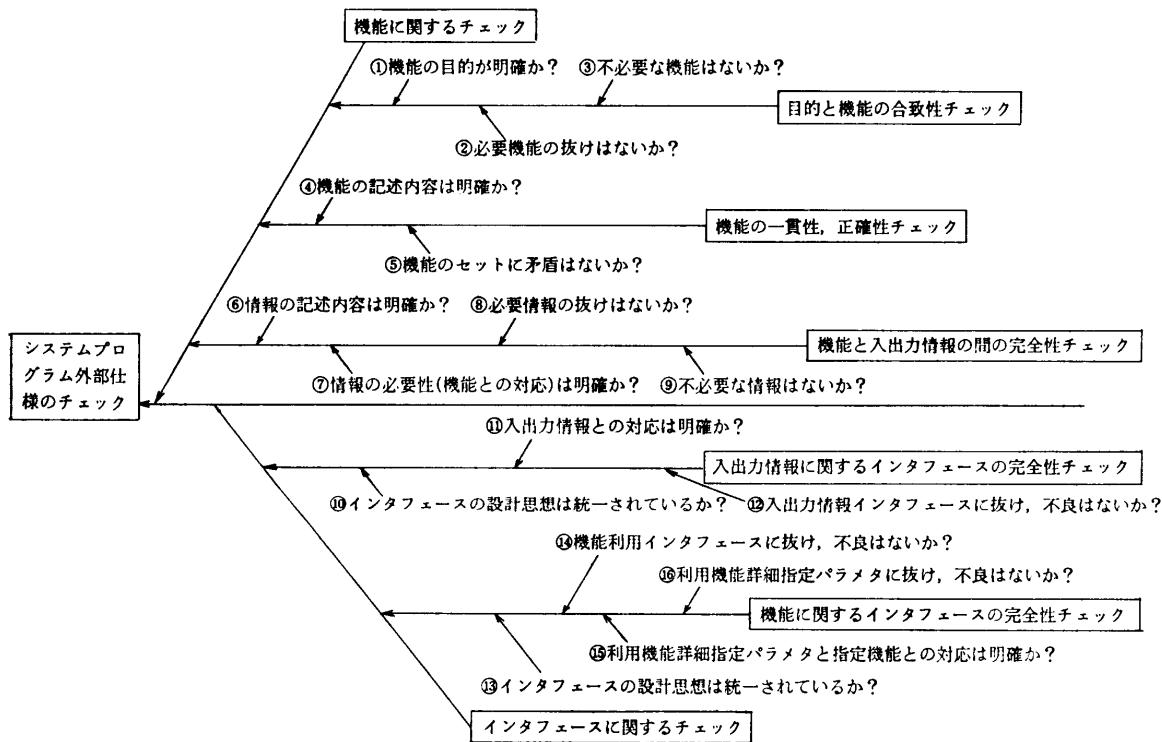


図 5 システムプログラム外部仕様のチェック項目

Fig. 5 Check list for system program's functional specifications.

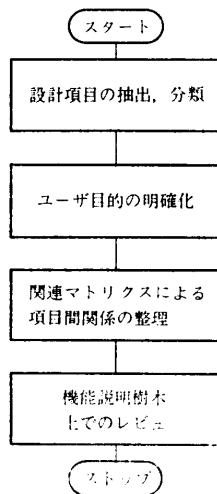


図 6 レビュ手順

Fig. 6 Review process.

樹木上にて、ユーザの目的までを総合して判断し、チェック項目②、③、⑤、⑧、⑨、⑩、⑪、⑫、⑬、⑯をチェックする。

5.2 試行例

本手法の試行結果を示しておく。対象は、新規開発システムプログラム(予想開発総ステップ数 32Ksteps)

で、従来手法の外部仕様レビュを終え、プログラム設計、一部コーディングに入っていた。これらの工程で、外部仕様不良が新たに指摘、修正されていた。試行は、これらの不良指摘以前の、もとの外部仕様書を用い、新たに指摘されていた不良内容を知らない第三者によって実施した。抽出した外部仕様不良を表2に示す。これらは、プログラム設計、コーディング段階で新たに指摘されていた不良をすべて含んでおり、本手法を用いることにより、従来手法では後工程にもれ

表 2 試行にて新たに抽出された外部仕様不良

Table 2 New errors of functional specifications detected by proposed method.

不 良 内 容	件数
・何に使用されるのか不明な機能がある	1
・設計された機能のみでは目的が達せられない	2
・不必要的機能がある	2
・機能のセットに矛盾がある	2
・必要な入力情報が抜けている	1
・必要な出力情報が抜けている	1
・制御ブロックのフィールドにセットすべき情報が不明である	3
・入力情報の指定方法に矛盾がある	1
・出力情報の出力場所がない	1
・制御ブロックの出力フィールドに出力される情報の説明がない	3
・機能利用インターフェースの設定に矛盾がある	1

てしまっていた不良の早期摘出が可能なことを示している。また、本表に示される不良は、仕様書の複数箇所の内容を総合して判定せねばならないものばかりであり、このような不良摘出に効果的であるという本手法の特徴を示している。

6. むすび

システムプログラム外部仕様のレビュー手法を提案した。現状レビューの問題は、仕様書そのものを直接レビューする点にある。すなわち、仕様書では、ユーザの目的とそれを達成する機能との関係、機能と入出力の関係等からなる設計項目相互の関連が Explicit にはておらず、レビュー時チェックする必要のある関係の有無の検討、分析の大部分がレビューに依存しており、レビューの能力によりその効果が左右される。本論文の提案は、設計項目相互の関連を Explicit に表現するモデルを基本とするレビュー手法である。

提案したモデルは、ユーザの目的を表わす目的項目集合、ユーザに提供する機能を表わす機能項目集合、入出力を表わす入力情報項目集合、出力情報項目集合、ユーザとのインターフェースを表わす機能利用インターフェース項目集合、利用機能詳細指定パラメタ項目集合、入力情報インターフェース項目集合、出力情報インターフェース項目集合からなるシステムプログラム外部仕様の設計項目の集合及びこれらの項目間の関係からなる。次に、提案モデルを基本に、レビューに項目相互の関連をビジュアルに認識させる、関連マトリクス、機能説明樹木を提案した。前者は、相互に関連する複数のマトリクスを用いモデルを表現するもので設計項目、項目間関係の抜け、誤り等の不良を網羅的にチェックできる。後者は、ユーザの目的の体系を最上位に、以下、個々の設計項目間関係に従い、設計項目を下位に展開した図形であり、設計項目全体を総合的に見た不良の判定を精度良く行える。最後に、これらの表現を用いる、具体的レビュー手順を提案した。従来

手法でレビューを終えている仕様書数件について、提案手法を適用し、従来手法では摘出されなかった新たな不良が摘出されることを確認している。

謝辞 本研究の機会を与えて頂いた、システム開発研究所所長三浦武雄博士および当社旭工場の篠宗泰孝部長、また、レビュー手法の共同検討をしていただいた、旭工場の下島賢二氏、松浦信彦氏および当社ソフトウェア工場の松下健彦氏に深謝いたします。

参考文献

- 1) Teichrow, D. and Hershey, E. A. III : PSL/PSA A Computer-Aided Technique for Structural Documentation and Analysis of Information Processing Systems : IEEE Trans. on Software Engineering, Vol. SE-3, No. 1, pp. 16-33 (Jan. 1977).
- 2) Alford, M. : A requirements engineering methodology for real-time processing requirements : IEEE Trans. on Software Engineering, Vol. SE-3, No. 1, pp. 66-69 (Jan. 1977).
- 3) Douglas, T., Ross, Kenneth E. Schoman Jr. : Structured Analysis for Requirements Definition : IEEE Trans. on Software Engineering, Vol. SE-3, No. 1, pp. 6-15 (Jan. 1977).
- 4) Pollack et al. : Decision Table : Theory and Practice, John Wiley & Sons (1971).
- 5) William R. Elmendorf : Cause-Effect Graphs In Functional Testing : IBM Technical Report (1973-11).
- 6) 菅野文友：ソフトウェア・エンジニアリング：日科技連（1979）。
- 7) 田代、春名：システムプログラム機能の階層表現法：情報処理学会第21回全国大会予稿集, pp. 399-400 (昭55-5).
- 8) 田代、春名：システムプログラム外部機能仕様のレビュー方法：情報処理学会第22回全国大会予稿集, pp. 321-322 (昭56-3).

(昭和56年5月21日受付)

(昭和56年11月18日採録)