

保守手順に適合する文書情報の編成法

野村 恵美子, 落水 浩一郎, 中村 圭吾, 小川 正明
静岡大学 大学院 静岡大学 工学部 住商コンピュータ
電子科学研究科 情報工学科 サービス株式会社

本論文では、事務処理システムにおいて、システムの機能に対する変更が要求されることにより発生する保守を対象として、変更対象プログラムの特定の際に必要となる作業を支援するための情報を記述する保守用情報モデルを提案する。本モデルは開発時文書を分析することにより設計情報を抽出し、保守者が変更箇所を特定する手順を分析することにより、変更箇所を特定する際に必要な情報を設計情報に付加して作成した。このモデルにしたがって実際のシステムからデータを作成し、実験的な保守活動に適用した結果、従来、保守者の経験に依存して進められてきた作業を明示化することができ、また、作業を効率化する効果が確認された。

An Documentation Schema Fit for a Maintenance Procedure

*EMIKO NOMURA, **KOICHIRO OCHIMIZU, ***KEIGO NAKAMURA ****MASAAKI OGAWA

*Graduate School of Electronic Science and Technology, Shizuoka University
**Department of Computer and Information Science, Faculty of Engineering, Shizuoka University.
***Sumisho Computer Service Corporation, LTD.

This paper describes an information model which can facilitate an adaptive maintenance process of a data processing system. To develop the model, design objects were identified from development documentation and monitoring data information structure of the maintenance process was attached to the objects. The result of experimental maintenance using a database designed on this model proved that our model can highly improve the efficiency of fixing the system components to be changed corresponding to end-user's change requirements and deciding the jobs to be re-run.

1. はじめに

ソフトウェアの保守にあたっては、以下のような問題点が存在する。

- (1) 保守に必要な調査が、保守対象システムに対する保守者の経験的な知識に基づいて行われるため、調査の十分性を客観的に評価しにくい。
- (2) 保守は、時間上の制約が大きいため、開発担当者のような、システムに対する知識が十分にある人以外は保守を行えない。

このような問題を改善するために開発時文書を利用することが考えられ、標準開発工程の文書をデータベース化することが試みられている^{1), 2), 3)}。しかし、保守時における文書の参照状況を調べると、多種類の開発時文書の内、ごく一部の文書のみが頻繁に参照される傾向がある。さらに、保守者自身のシステムに対する理解に依存して進められると思われる作業もあり、開発時文書の体系を直接データベース化することは、上記のような問題点の十分な改善にはつながらないと考えられる。すなわち、保守の効率化のためには、開発時文書から保守時に利用される正味の情報を抽出し、データベース化することが必要である。

本報告では、事務処理システムにおいてユーザからシステムの機能に対する変更が要求されることによって発生する保守を対象とし、変更対象プログラムの特定に関連して必要となる作業を支援するために必要な情報を抽出する目的で行った以下の実験の結果について述べる。

- ・開発時文書、および開発時文書を用いた保守手順を分析することにより、システムの構造を特徴づける情報と、変更要求を基にそれらに対応づけを行うために必要な情報を抽出し、構造化する。

- ・抽出した情報構造を実際に開発・運用されているシステムの保守に実験的に適用することにより、この情報構造の保守活動に対する有効性を確認する。

以下、2章で、事務処理システムの構造上の特徴と開発時文書の構成との関係を議論し、開発時文書を保守に利用する際の問題点を明らかにする。3章では、設計情報の抽出とモデル化、および変更箇所特定の観点から付加される情報スキーマの設計を行い、保守用情報モデルを構成する。4章では、このモデルを実験的な保守作業に適用した結果と考察を示す。

2. 設計情報の抽出および保守手順の分析

2.1 事務処理システムの構成

事務処理システムでは、一定のタイミングで一連のデータ処理が行われ、ユーザからのシステムへのデータの投入、計算・編集処理、ユーザへのデータの出力が実現される。このようなデータ処理の過程は、図1のように示すことができる。

図1において、一連のデータ処理を引き起こすタイミングには、「月次」、「期末」のような定期的な繰り返しと、「振替情報の入力」のようなシステム外部からの非定期的なデータの到着とが含まれる。あるタイミングで引き起こされるデータ処理の過程は、ファイル、データベースをインタフェースとして、システム外部からの入力データまたはインタフェースデータをもとに、次のインタフェースデータまたはシステム外部への出力データを作成するようなデータ接続の連続的な実行としてとらえられる(図1の各円内)。このようなファイルレベルで把握されるデータ処理の流れをファイルレベルのデータの接続関係とよぶことにする。

同じタイミングで実行されるファイルレベルのデータの接続関係をジョブフローとする。ジョブフロー間のインタフェースとなるファイル、データベースをジョブフローのインタフェースデータとよぶ。図1では、データベース2が振替情報の入力と月次のジョブフロー間のインタフェースデータであり、データベース3が月次と期末のジョブフロー間のインタフェースデータである。

1つのジョブフロー内におけるデータの接続点の最小単位は、実行単位のプログラムから構成される。バッチプログラムは、それぞれのプログラムが最小単位のデータの接続点となり、オンラインプログラムは、1つまたは複数のプログラムで最小単位のデータの接続点を構成する。オンラインプログラムにより構成される最小単位のデータの接続点をオンラインジョブとよぶことにする。連続的に実行されるいくつかのバッチプログラムは、オペレータの起動単位であるバッチジョブに編成される。バッチジョブが起動されると、ジョブ内のプログラムが一定の順序で起動され、データの接続が行われる。同一のタイミングで起動されるジョブ(バッチ、オンラインを含む)間には、一定の

起動順序が与えられ、それに従ってジョブを起動することによりジョブフローが実現される。ジョブ間でインタフェースとなるファイル、データベースをジョブのインタフェースデータとする。

パッチプログラムとオンラインジョブでは、入力ファイル、入力票からレコード単位でデータが入力され、レコードを構成する項目の値を使って一定の手順で計算が行われる。計算結果は出力ファイル、帳表のレコードに編集され、レコードの生成、更新が行われる。このような入力項目値からの出力項目値の生成（項目レベルのデータの接続関係）がデータ接続の基本単位となる。

項目レベルのデータの接続関係は、図1に示すような計算のパスとして把握できる。図1の例では、a, c, eによってfi1の値からfo1の値を生成するパス、a, b, c, d, fによってfi1, fi2の値からfo2の

値を生成するパス、gによってfi2の値からfo3の値を生成するパスの3本である。このような計算のパスをデータの接続パスとする。データの接続パスにより、項目レベルのデータの接続関係が構成される。

2.2 開発時文書の編成と問題点

前節で述べたシステムの構成は、図2に示すような体系で作成、文書化される。図中で□は文書を示し、□内に各文書の記述内容を説明する。

文書の作成過程は、3段階にわけられる。レベル1では、開発者がユーザ要求の把握を行う。ユーザ要求は、業務という単位で大きく分類され、各々の業務は、減価償却計算、固定資産台帳作成のような、データに対応した作業（以下、システム機能とよぶ）から構成される。開発者は、まず、システム化する業務、およびユーザとシステムとの間のインタフェースとなるデ

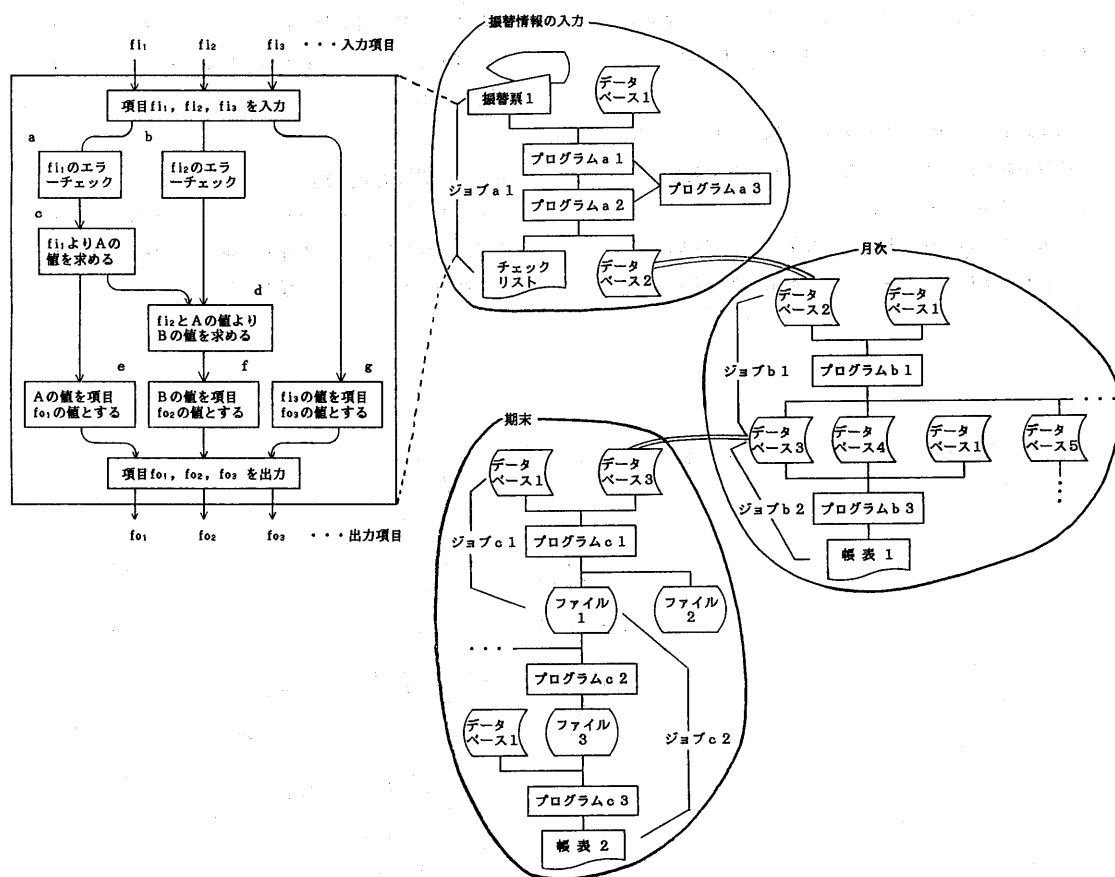


図1 事務処理システムの構造

ータ（入力票、帳表）を洗い出す。さらに、業務からシステム機能を抽出し、システム機能の実施時期、および扱うデータを調査する（文書①～⑤）。その結果、ファイル、データベースのようなシステム内のデータが導入され（文書④）、各業務におけるシステム機能とシステム内のデータ間の入出力の関係を定め、さらに入力票や帳表の入出力を行うシステム機能を定める（文書⑥）。

レベル2では、システム機能の実施時期に基づきタイミングを設定し、システム機能の入出力データよりタイミングごとにジョブフローのインタフェースデータ、入力票、帳表を定義する（文書⑦、⑧、⑨）。システム機能に対してさらに詳細な調査を行うことにより、ジョブフロー内のファイルレベルのデータの接続関係と、各接続点における接続パス上の計算を定義する（文書⑩、⑪）。

レベル3では、ファイルレベルのデータの接続関係をジョブに分解し（文書⑫）、計算を手続き化する（文書⑬）。

開発時文書を利用して保守を行う場合、次のような手順で開発過程に沿って文書を追跡し、変更箇所を特定することが考えられる。

①変更の対象となる機能を含むタイミングを特定することにより、そのタイミングで実施されるジョブ

フローを抽出する。

②ジョブフローの中から変更対象機能を実現している最小単位のデータの接続点を特定することにより、変更対象プログラムを抽出する。

③接続点におけるデータの接続パスの中から変更対象機能に対応するものを特定する。

このような変更箇所特定の方法においては、文書を追跡するにしたがって変更要求をより詳細なシステムの構成要素に対応づけていく必要があるが、一般に、ユーザはシステムの構成については十分な理解がないため、このような対応づけに関する情報をユーザから得ることはむずかしい。開発時文書の中でシステムの構成に基づく編成をとらないものは、要求の分析・定義の際に作成される文書（図2文書①～⑤）であるが、これらの文書は、業務名、システム機能名、およびその概要の説明が記述されているのみで、計算方法等の具体的な内容は含まないため、文書の追跡を始める前に具体的なレベルで保守要求を確定させておくことも困難である。

2.3 保守手順の分析

そこで、以下のような方法で、現状では保守時にどのような情報に基づいて変更対象プログラムを特定しているのかということを調査した。

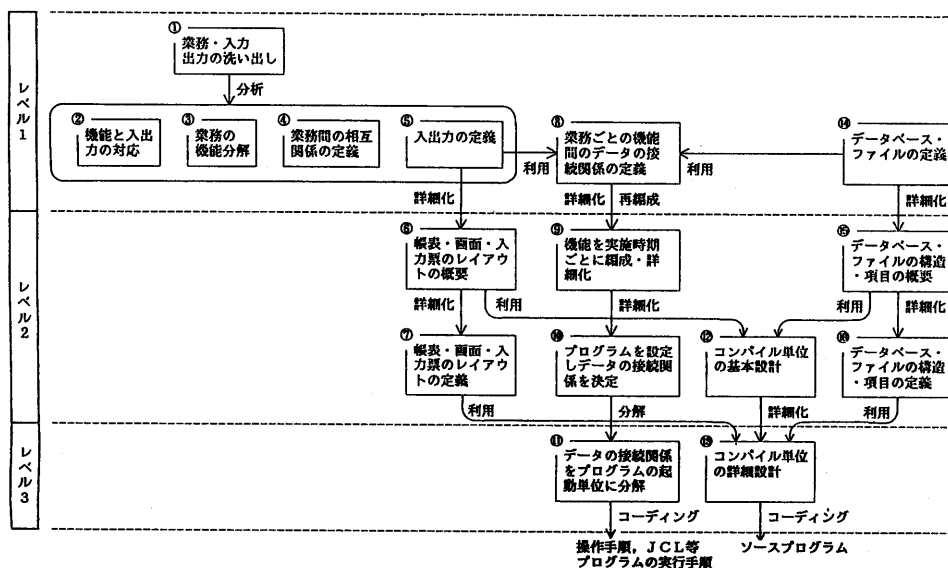


図2 事務処理システムの文書体系

NO	調査内容
1	業務説明書から、計算式の名前、計算式を含む機能の名前、計算式に含まれるデータ名、計算の実施時期等、変更が指示された機能に関する情報を集める。
2	1で得た計算の実施時期に対応する図1文書⑩を参照し、プログラム名、データの接続関係を調べ、変更対象となるプログラムを特定する。
3	特定したプログラムに対応する文書⑩を参照し、その内容に基づいて変更内容を確認する。

表1 変更対象プログラムを特定する調査手順

- 1) 設計情報の調査を行ったシステムに対し、開発時にユーザがシステム機能の説明のために作成した文書（業務説明書）を用いて保守要求を設定する。
- 2) これをユーザからの保守要求として、システムの開発者に変更対象プログラムを特定してもらう。
- 3) この過程でどのような情報に基づいてどのような判断を行ったか、その情報を図2の文書から得る場合、どの文書を参照しているか等をインタビューする。

調査の結果を表1に示す。この結果より、保守要求の確定は、文書⑩の記述に基づいて行われることがわかる。保守者は、ユーザが述べる保守要求に含まれる情報に基づいて、保守対象となる計算を含むプログラムを直接に抽出し（表1ステップ2）、このプログラムに対応する文書⑩を調査することにより、保守対象となる計算に関する記述を見つけ、保守要求の確認を行っている。しかし、保守要求に含まれる計算と、その計算を実現しているプログラムとの対応関係を記述した文書がないため、文書⑩に含まれるプログラム名、ファイル等のデータ名、プログラムとファイル等のデータの接続関係より、プログラムに含まれるデータ項目の接続関係を思い出す、または推測することにより、変更対象プログラムを特定している。

3. 情報構造の抽出

2. 3節でのモニタリングの結果より、以下の2点が結論される。

- ①保守要求は、システム機能に対する計算式レベルの具体的な記述に基づいて確認される。
- ②システム機能をこのような具体的な単位でとらえた場合、プログラムおよび項目レベルのデータの接

続関係と直接対応が付き、変更箇所特定にはこの情報が有効である。

本章では、前章での考察結果をふまえて、保守用情報スキーマを設計する。

3.1 設計情報のモデル化

ファイルレベルのデータの接続関係を記述する情報として、ジョブフロー、ジョブ、プログラムそれぞれを接続点とするデータの接続関係、ジョブフローとジョブ、ジョブとプログラム間の階層関係、およびジョブフローの実施タイミングを抽出した（図3①、②、③）。データの接続関係は、接続点となるジョブフロー、ジョブ、プログラム、および接続点間のインタフェースデータ（ファイル、データベース）、ユーザへの出力データ（帳表）、ユーザからの入力データ（入力票）を対象として、接続点とデータの間の入力、出力の関連で接続点におけるデータの接続を表現し、ある接続点の出力が他の接続点の入力となることによるデータ接続の連続性を、前者の出力データが後者の入力データと一致することで実現するものとした。

項目レベルのデータの接続関係は、最小単位の接続点から接続パスを抽出し、接続パスと接続パスで設定、参照されるファイル、データベース、入力票、帳表の項目との間に設定、参照を表す関連ををつけることで表現する。接続パスとそれを抽出した接続点との関係を接続点を構成するプログラムとの間の関連「プログラム機能」で保持する（図3④）。

3.2 変更箇所特定のための情報スキーマ

システム機能の計算式レベルでの記述とプログラムおよび項目レベルのデータの接続関係との対応をモデル化する際の問題は、計算式レベルの記述をまとめる単位である。開発時文書の中で計算に関する記述を含む文書は、図2文書⑩、⑪であるが、文書⑩における記述のまとめ方は文書作成者によりかなり任意性があり、文書⑪はソースプログラムのステートメントと対応するレベルで記述されるため、意味的なまとまりがとらえにくくなっている。そこで、以下のような方法で計算に関する記述をまとめる単位を抽出した（図3④、⑤）。

4.1 関係スキーマの設計

図3のモデルを関係データベースのスキーマに編成する際に、以下のような方針をとった。

- 1) 関係が実験者にとってなじみやすい概念に対応すること。
- 2) 関係の結合操作を行わずに、できるだけ多くの情報をみることができるように、関係の数はなるべく少なくすること。
- 3) 関係の実現値に重複するデータが多く存在することになっても、現時点ではよいとする。

その結果、図3の点線に示すようにモデルを分解し、それぞれの部分に対応して関係を設計した。

4.2 実験の概要

データ作成の対象としたシステムは、2章、3章の調査で利用したシステムで、期末、上期期首、下期期首、月次、いくつかの特定月と、随時（入出力発生時）に実施されるジョブフローから構成されるが、この中から、ユーザ入力から帳表出力までのデータの流れが存在する比較的小さな部分として、随時実施される4つのジョブと、月次に実施される3つのジョブを選び、図3から設計した関係スキーマにしたがってデータを作成した。実験設備として、IBM5540上のinformixを用いた。

保守要求としては、次の2つの型のものをいくつか想定し、変更対象となるプログラムの特定に関して、どのような情報検索が行われるかを追跡した。

- 1) 「計算内容」の定義の変更を直接要求する。
- 2) 帳表の項目を指定して、その項目の計算方法を変更したいと要求する。または、入力表の項目を指定して、その項目のチェック方法を変更したいと要求する。

想定した保守要求から変更対象の候補となる「機能単位」がいくつか抽出された時点で、適当なものを選択し、保守要求を確定させるものとした。

実験者は、システムの開発、保守には十分な経験があるが、保守対象としたシステムに対しては、概要についての知識があるのみで、データ処理の構造など、詳細については、知識はなかった。

4.3 実験結果の考察

想定した保守要求すべてに対し、変更対象プログラムを特定する過程で、次のような手順の情報検索が行われた。

①変更対象機能の確定

保守要求中のデータ名を使って、「機能名」、「計算内容」を検索し、保守要求と照合することにより、これらに関連「名前づけ」、「内容定義」で対応する「機能単位」を変更対象として確定させる。

または、保守要求中で指示された「帳表」、「入力票」、「項目」から、関連「項目構成」、「機能定義」を使って、保守対象となる「機能単位」を抽出し、これに対応する「機能名」および「計算内容」の記述を確認することにより、変更対象となる「機能単位」を確定させる。

②変更の影響を受ける帳表の確認

関連「機能定義」を使って、変更対象である「機能単位」が関係している帳表の項目を調べ、その項目が変更の影響を受けることをユーザに確認する。

③変更対象プログラムの特定

関連「実現」、「プログラム機能」を使って、変更対象として確定した「機能単位」から変更対象となる「プログラム」を抽出する。

④テストジョブの抽出

関連「ジョブプログラム構成」、「ジョブ入力」、「ジョブ出力」を使って、次のようなジョブをテストジョブとして抽出する。すなわち、③で特定された変更対象プログラムを含むジョブから②で確認された帳表を出力するジョブまでのファイル単位のデータの接続関係を形成するジョブ。

このような情報検索の手順は、実験者の経験と比較して、従来、変更対象プログラムの特定に関して行われてきた調査をほぼ十分に手順化していると判断してよいという評価をえた。しかも、②～④の検索において、帳表、プログラム、ジョブは、①で変更対象として確定された「機能単位」に対して、十分に、または正しく抽出されていた。

以上の実験結果、および結果に対する評価から、図3のモデルに従ってシステムに関する情報を整理する

ことにより、保守者が図3のモデルとその関係スキーマによる実現方法を理解していれば、保守対象システムについて十分な知識がなくても、変更する「機能単位」が確定した後、②～④の手順に従って変更対象プログラムの特定に必要な調査が行えると判断される。調査の正確さ、十分性については、作成されたシステムに関するデータの正確さ、十分性の問題となり、データ作成における技術的な問題として考えることができる。

②と③におけるデータベースに対する検索が、

1) 検索の対象となる関連を実現しているデータベースの関係を選択し、

2) 対象「機能単位」を指定し、関連「機能定義」を使って対象「項目」を得る(②)、または関連「実現」と「プログラム機能」を使って対象「プログラム」を得る(③)

という機械的な手順で容易に行えたのに対し、④の条件を満たすジョブを検索する作業は煩雑であった。②と③の結果に基づいてこの条件を満たすようなテストジョブを抽出するツールの作成が望まれる。

①の前半のような方法で変更内容の確認を行うための「機能単位」を抽出するには、ユーザから提示される保守要求から「機能名」、または「計算内容」で使われるデータ名を取り出さなければならない。しかし、一般に、ユーザがデータや計算方法の指示に使う名前には、同義語、多義語が多く、この作業にはかなりの試行錯誤が必要となることが予想される。変更対象となる「機能単位」の確定を効率的に行うためには、現在、文章として扱っている対象「計算内容」に含まれる情報の構造を分析することが必要であると考えられる。

また、今回の実験で検索の対象とされなかった対象、および関連については、さらに多様な保守要求に対してこのモデルを適用してみることで、役割を検討する必要がある。

5. おわりに

保守においては、最終的には、ソースリストに対して、詳細な調査を行い、変更を行わなければならない。しかし、大規模な事務処理システムでは、まず、このような調査の対象となるプログラムを特定することが

必要である。本モデルは、従来、保守者の経験に依存して進められてきたこのレベルの作業の内容と手順を明示化し、作業を効率化する効果があった。

今回の実験では、すでに開発されたシステムを対象に、開発時に作成された文書、またはソースプログラム群から、手作業でデータを作成したが、この作業には、かなりの期間を要した。本来このようなデータは開発時に作成すべきであり、今後、本モデルに従ったデータを開発時に投入する方法を開発することが必要である。

参考文献

- (1) E. Horowitz and R. Williamson: "SODOS-A Software Documentation Support environment: Its Use", Proceedings 8th ICSE, 1985.
- (2) M. H. Penedo, E. Don Stuckle: "PMDB-A Project Master DATABASE For Software Engineering Environments", Proceedings 8th ICSE, 1985.
- (3) E. H. Bersoff, V. D. Henderson, G. Siegel: "Software Configuration Management: A Tutorial", Computer, Vol. 12 No. 1, January, 1979, pp. 6-14.