

FS (Flow Sketch) 法による要求定義技術

松本良治 谷昌之 山本昌彦

日立ソフトウェアエンジニアリング(株)

1. はじめに

ソフトウェアの大規模化に伴い、品質の低下、開発/保守費の増大といった問題が生じソフトウェア危機と呼ばれる状況が生まれ、この問題に対処するために、ソフトウェア工学¹⁾の研究が盛んになってきた。

ソフトウェア工学の研究成果は、技法としては、構造化プログラミングや、トップダウン設計等が発表されており、効果のあることが既に確認されている。また、プログラミングツールとして、エディタ等が発表され、効果を上げているが、これらの技法やツールは単独に、かつ開発工程別に利用されているのが現状であり、ソフトウェア開発の全サイクルの合理化、効率化を考えると、有機的に結合された総合的なシステムとする必要がある。現実にソフトウェアの製造工程においては、構造化プログラミングとそれに適した言語処理系や、テキスト・エディタ等が存在し、作業環境もバッチシステムからTSSへと発展している。さらに、これらをソフトウェア開発支援システムとして位置付け、有機的に結合する努力もなされ、実現しつつある。

しかし、ソフトウェアの開発過程は、要求分析、要求定義、設計、製造、テスト、保守等の作業とそれに伴う文書化が複雑にからみ合っており、製造工程はその一部にすぎない。真に総合的なソフトウェア開発支援システムを構築するには、システム設計、テスト、保守等の工程に対する要求工学的な方法論が必要である。要求工学²⁾は、比較的歴史の浅い分野であるが、ソフトウェア開発の初期段階で作り込んだ抜けや誤解の修正に多大な費用

がかかることが認識されるにつれ、ソフトウェア工学の重要な位置を占めるようになった。

要求工学の目的は、つぎのことである。

- (1) 要求を正確に記述できる手段を与える。
- (2) 要求の一貫性、完全性、整合性を検証できる手段を与える。

この目的のために具体的な方法で解決する技法の研究を行い、フロースケッチ (Flow Sketch ; FS) 法という技法を開発した。

要求定義は、要求元からの機能、ハードウェア、性能のシステム構築に対する要求を定義付けるものであり、フロースケッチ法は、要求定義の中の機能要求を定義付ける単純で効果的な方法論である。

フロースケッチ法は、特定のカードを使い、ソフトウェアシステムに対する要求を、要求元が必要とするデータという形で書き出すことから始まる。次に、そのデータをもとにして、誰が何時、何のためにそのデータが必要かといったデータの特性を分類しながら、各データ間の流れを描くことにより、ソフトウェアシステムで実現する機能を定義する要求モデルを完成していく技法である。

本報告では、フロースケッチ法の特長とその適用方法について述べる。

2. フロースケッチ法の問題点

2.1 要求定義の問題点

要求定義は、要求元のシステムに対する機能要求、ハードウェア要求、性能要求を分析して、実現方法を明確化するものであり、これをまとめたのが要求定義書である。この要

求定義において、ハードウェア要求や性能要求は、目標値を定めることで終了し、その実現性は、機能設計によって検証され、設計変更が比較的可能である。しかし、機能要求に関しては、要求元に提供すべきソフトウェアシステムの機能の抽出はできるが、その実現性においては、ソフトウェアシステム構築過程において早期に検証する手段がなく、大部分の誤りは計算機で実働化した段階で表面化するのが現状であり、そのための設計変更に要する費用も多大である。

2.2 フロースケッチ法の考え方

従来の機能要求は、機能を中心にデータの入力および出力の定義付けを行っていたが、機能中心の手法は、データの流れが不明確となる点や、データの欠落があっても気付かない問題があった。

そこで近年は、データを中心に機能の抽出を行う研究が盛んになり、SREM³⁾、ISDOS⁴⁾等の具体的な技法も開発され、試行例も報告されている。

今回開発したフロースケッチ法は、SREM、ISDOS等と同様にデータを中心に機能要求を定義付ける技法であるが、SREM、ISDOSは、計算機を用いて要求を整理する手段をもっているが、フロースケッチ法では、要求定義は計算機でできる単純作業とは考えず、むしろ整理分類の過程こそ人間の思考をめぐらす重要な頭脳作業と考え、計算機処理は敢えて導入せず、要求モデルを完成させることを考えている。したがってフロースケッチ法は、要求元のデータの抽出とデータ間の関係を明確にすることが、機能の誤解釈を防ぐために重要なことと判断し、データの抽出およびデータ間の関係付けこそ思考錯誤を繰返して、かつその過程を着実に残しながら行うべきであるとの思想にたっている。さらにフロースケッチ法は、曖昧になりがちな文章

表現ではなく、図式表現でデータの関係を関連付けることが重要であると考えている。

2.2 フロースケッチ法の特長

要求定義におけるデータ中心のアプローチは、つぎの理由による。

- (1) 要求は機能ではなく、機能を使用した結果得られるデータである。
- (2) 機能の記述には、データの記述が不可欠である。
- (3) データは要求元が直接求めているものであり、明確化しやすい。

フロースケッチ法は、上記のことを考慮して、計算機を用いずに要求定義を具体的な方法論として確立するために、データの流れを着実に残していく手段を検討し、データとその流れ、および仕事の流れを一体化しながら機能を明確化する方法を採用した。

フロースケッチ法の特長は、つぎのとおりである。

(1) カードの採用

機能を構築する要素はデータである。このデータを、データの要求とその供給の関係を明確にしながらか抽象化していくことにより機能を抽出できる。一般にデータの記述は一覧表等に項目を記述することで行っていたが、この方法は、データの項目の記述は可能だが、データの要求と供給の関係は明確にできない。この問題に対して、図1に示す形式のカードを採用した。本カードは仕事を行う者が必要とするデータを要求として記述し、仕事をする者が渡すデータを供給として記述する。データを抽象化する過程で、各データの流れを定義する。本カードの採用により、データの要求と供給の関係を明確にしながらか、方法論的にデータの抽象化が行え、機能の抽出が実現できる。なお、本カードをFS (Flow Sketch) カードという。

という。

要求	仕事	供給
(1)内容	(1)インタフェイス	(1)内容
(2)何時	(2)内容	(2)何時
(3)インタフェイス		(3)インタフェイス

図1 FSカード

(2) 作業員からのデータトレース

機能要求において重要なことは、システムにとっての入出力データ、およびデータを要求する作業員を明確にすることである。従来、機能の抽出においてはシステムが要求するデータの抽出は可能であったが、作業員が要求するデータが正しく得られているかの検証手段がなく、計算機によるデバッグ段階において誤解釈等が判明する等の問題があった。

FSカードは、データの要求と供給の関係と同時に、そのデータを使用する作業員および時点を記述し、データの整合性を絶えず検証可能としてある。なお、作業員をインタフェイス(Interface)と呼ぶ。

(3) ネットワーク図の採用

ソフトウェアシステム構築において重要な要因は、業務の流れとデータの流れを明確にして、機能の実現性を検証することである。従来は機能中心で、データの流れからの実現性を表現することが難しい。

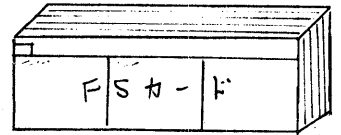
フロースケッチ法では、インタフェイスとデータの流れ、および作業の流れを表現したネットワーク図を作成することにより、具体的に機能の実現性の検証を可能とした。この図をFSチャートといい、インタフェイスとデータとの連関図をインタフェイスネットワーク(Interface network)

3. 要求定義手順

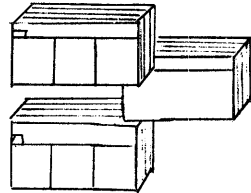
フロースケッチ法による要求定義は、FSカードの作成、要求データの類型化、データフローの決定、要求定義モデルの作成の過程を経て完了する。

図2は要求定義手順を示している。

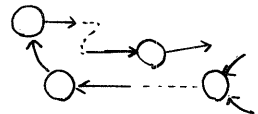
(1) FSカードの作成



(2) 要求データの類型化



(3) データフローの決定



(4) 要求定義モデルの作成

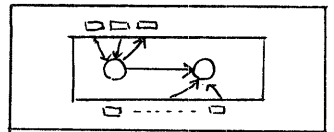


図2 要求定義手順

(1) FSカードの作成

FSカードに仕事の内容、インタフェイス、要求データの内容、要求時点を記入する。他の項目は、明確な場合に限り記入す

る。個々のインタフェイスが、何時、どういう目的でデータを要求するかを抽出すればよく、データ間の関連やソフトウェアシステムとしての実現性の考慮は一切する必要はない。図3はFSカードの記述例である。

要求	仕事	供給
(1) 納税者氏名	(1) 登録者	(1) 納税者氏名
(2) 納税者登録時	(2) 納税者登録	(2) 納税者登録時
(3)		(3)

図3 FSカードの記述例

(2) 要求データの類型化

FSカードを、インタフェイス、仕事の内容、データの要求時点をキーにして類別する。類別したデータ群は、要求時点、要求者、要求目的が一致した意味的にまとまったデータの集合体となり機能の抽出につながる。この段階で、各データ群毎に必要なデータの欠落、不要データの混入をチェックし、データが十分かを検討する。類型化の結果は、代表データカードとして、類型化した各データ群毎に一枚作成する。

(3) データフローの決定

データの流れに基づいてインタフェイス間を関連付ける。各インタフェイス毎に、要求データと供給データの整合性を検証し、要求データと仕事から供給データを生成できることを確認する。

(4) 要求モデルの作成

データフローをネットワーク化し、FSチャートとして図示する。FSチャートをもとに、システム機能を抽出した要求モデルを定義する。要求モデルは、インタフェイスからみたシステム機能を、データの要

求、供給に対応して与えるものである。図4に要求定義モデルの例を示す。

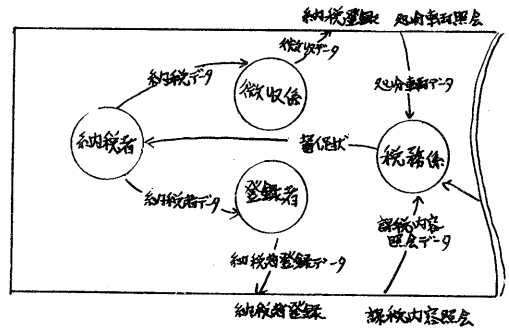


図4 要求定義モデル

4. おわりに

本報告では、フロッスケッチ法の特長とその適用方法について主に述べた。

フロッスケッチ法を業務プログラムに適用した結果、ユーザの要求が明確に記述でき、実現性の検証が容易にできることが判明した。今後は、機能設計へとつなげる方法論を研究する所存である。

5. 参考文献

- 1) 国井利泰, 他; 要求定義技術, 共立出版, 1978
- 2) 松崎稔; 進展著しいソフトウェアの要求仕様技術を見る(上)(下) 日経エレクトロニクス, 1980.1 PP.65-84, 1980.2 PP.96-110
- 3) Alford M.W., 'A Requirement Engineering Methodology for Real Time Processing Requirements, IEEE Trans-SE vol SE-3, No1, pp60-68, 1977
- 4) Teichrow, D and Hershey, E. A., 'PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems, IEEE Trans-SE, vol SE-3, No1, pp 41-48, 1977