

複数ビューに基づくシステム仕様化過程の分析

3H-1

西村一彦 春木和仁
 (株)東芝 研究開発センター
 システム・ソフトウェア生産技術研究所

1 はじめに

システムの多様化に対応して、柔軟で効率的な仕様化作業を支援する技術の開発を行っている [1]. 今回、支援システムを試作し、実際に仕様書(ソフトウェアの特許明細書)の作成に適用し、評価実験を行った. 本論文では、仕様化作業における設計者の視点およびその視点の変化についての実験結果を述べる.

2 仕様化作業プロセスの特徴

仕様化作業は仕様獲得と文書化の2つの作業からなる. 仕様獲得は曖昧な顧客要求を様々な視点から詳細に定義し、要求を実現するシステム概念を決定する作業である. この作業を支援するために多くの方法論が提案され、作業の具体的な内容や手順が定義されている [2]. 文書化は決定されたシステムの正確かつ詳細な振る舞いを一定の形式に従って記述し、成果物として作成する作業である.

これらの2つの作業は相互に依存して進行する. すなわち、ある視点で得られたシステム概念は仕様書の複数箇所に表れる. また、現在の視点と作業状態により次の視点が決まる. ゆえに、仕様化作業の支援は、これらの相互に依存し合う2つの作業を支援することである.

3 仕様化作業プロセスにおける複数ビュー

方法論を実際に活用するためには専門知識やノウハウが必要となる. 特に、仕様獲得作業の過程で、設計者の視点とその変化を明確にすることが求められる [3]. ここで、視点(ビュー)とは対象システムの持つ様々な諸概念を定義する枠組みである. 対象システムをソフトウェアに制限すると、基本的なビューとして機能面、動作面、構造面に関連したものがある. 今回は図1に示すビューを用いる. 矢印はビューの変化可能な関係を示す.

機能的ビューは対象システムの処理と処理間のデータの流れに関連する. データ変換処理(SP), 変換処理に関連するデータ(DF), データストア(DB), データの発生源と受理者(TM)がある.

動作的ビューは対象システムの振舞いに関連する. システム処理(PR), 条件判定(IF), 処理の流れ(FL), そして、条件つき制御の流れ(FC)がある.

構造的ビューは対象システム内のデータの属性やデータ間の関係に関連する. システムにおけるデータ(EN), データの属性(AT), データ間の関係(RL)がある.

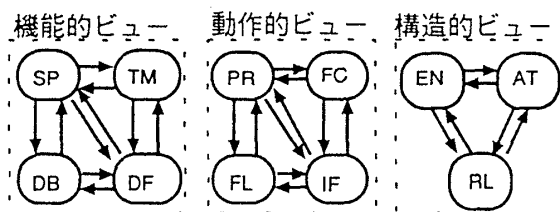


図1: ソフトウェアシステムに関連するビュー

4 文書化の役割

文書化の役割は、1)ドキュメントの自動生成(仕様獲得の成果(図面)を正確に表す自然言語の文書を作成する),そして、2)仕様のチェック(仕様獲得の成果自身に抜けや矛盾がないことを確認する)[3]にある.

本稿では、2)の支援方法として、図面内容を自然言語表現で言い換える機能を提供する. 評価実験では、この機能がどのように利用されているかを分析する.

[SP]は、[S]より、[DF1]を[動詞].
 そして、その値に基づいて[T]に[DF2]を[動詞].
 動詞群: 読み込む、計算する、伝送する、出力する
 S・T: SP、TM、DBで明かになる要素名
 DFx: DFで明かになる要素名

図2: 説明文のテンプレート

説明文はテンプレートに図面情報を埋め込むことによって作成される. 図2はデータ変換処理(SP)のテンプレートである. []で囲まれた部分は、図面の基本構成要素の名称である. 基本構成要素間の関係に応じて、動詞が選択される. 例えば、図2では、入力先(DF1)がデータストアの場合、動詞「読み込む」が選択される.

5 仕様化作業の分析実験

実験システムを、実際の仕様化作業に適用し、仕様化作業におけるビューに関して、1)ビューを意識する順序、

2) 設計経験の違いとビューの関係, 3) ビューを変更する理由, を分析する.

実験システムの機能構成を図3に示す. 図面エディタは, ビューを図形部品として提供する. 設計者の意図したビューは選択した部品によって明らかになる. ビューの変化は図形部品に対する編集操作で示す. すなわち, 部品の追加, 削除, 名称変更で, ビューの変化が明らかになる. 文書エディタは仕様書の項目の論理構造に従って文書編集を支援する. 文生成部は, 図面説明文をテンプレートによって生成する. 図面作成過程記録部は設計者の図面作成作業の過程を記録する.

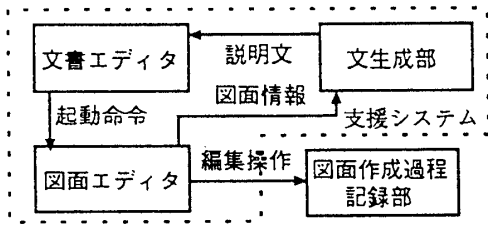


図3: 実験システムの構成

被験者は, 対象システムの詳細(特許明細書の実施例に相当する部分)を実験システムを用いて作成する. そして, 被験者の図面作成履歴をもとにして, 被験者にインタビューを行った. なお, 今回はビューを機能面と動作面に関連するものに限定した. 本実験の目的の一つである設計経験とビューの関連を分析するために, 被験者は2つのグループからなる. グループ1(G1)は大規模なソフトウェア設計の経験が豊富であり, 過去に10回以上の明細書作成経験がある. グループ2(G2)は小規模のソフトウェア設計の経験はあるが, 明細書の作成経験が乏しい. 表1は作業過程の記録結果である.

表1: 仕様化過程の比較

	グループ1	グループ2
追加	201	183
削除(対追加)	72(0.36)	99(0.54)
変更(対追加)	80(0.40)	55(0.30)
総ステップ数	353	337
ビューの変化	機能→動作→機能	機能→動作

6 分析結果と考察

1) 視点を意識する順序 視点の変更順序は対象システムの特徴に依存する. G1とG2は各々数式処理とデータ処理システムの仕様化を行なった. G1は機能面に注目して作業を始めたが, すぐに動作面に視点を変えた. これは機能より先に動作を意識したことを示す. 理由は, 対象システムが数式によって表現されており, 仕様化のポイントがその処理順序にあったからである. 一方, G2は機能面に注目した作業をすべて終えてから, 動作面を

注目していた. これは, データ処理システムの仕様化のポイントが機能面にあったことによるものと思われる.

2) 補助的な視点 仕様化作業の効率アップに貢献する補助的な視点がある. G2は特定のビューを連続的に注目していた. 例えば, 機能面では, まずデータストアだけに注目し, ついでデータ変換処理だけに注目する. その結果は冗長なデータ変換処理を含むデータフロー図をもたらし, その後に頻繁な削除が行なわれた. 一方, G1は基本的なビュー(各構成要素)だけでなく, それらの関係にも注目する. これは, G1が補助的な視点をもっていたことを示す. 例えば, データ変換処理とデータストアというビューをサブシステムという視点で意識している. ある要素を編集するとその要素が属しているサブシステムの他の要素にもその影響が及ぶ. この結果, G1は冗長や矛盾のない図面を効率良く描くことができた.

3) 視点を変更する理由 第1の理由は, 他のビューの影響による. G1は名称変更を頻繁に行っている(変更/追加=0.40). G1の場合, 名称変更はそのほとんどが動作面に関連したビューでの作業結果の影響を受けている. G1は動作面から本格的な作業を始め, 機能的なビューに移っている. ゆえに, 機能的ビューから得られたデータフロー図は動作的ビューの影響を多くの点で受けている.

第2に対象システムにおける要素の役割という補助的なビューを考慮することによる. システムにおける役割が類似した要素はその名称も似ている. G1の場合, ある要素を編集すると類似の他の要素も編集する. 例えば, 「〇〇修正手段」を「〇〇計算手段」に変更すると, 「××修正手段」も「××計算手段」に変更している.

第3に説明文の読みやすさによる. 曖昧な図面から生成される文は非常に難解である. 例えば, 複数の入出力データをもつデータ変換処理において, 入出力データ間の依存関係が不明な場合, 作成される説明文は非常に難解である. この結果, 設計者は意味不明なデータ変換処理に視点を移し, データ変換処理を分割するか, あるいは入出力データに注目してそれらの依存関係を定義する.

7 おわりに

ソフトウェアシステムの仕様化過程を設計者の視点という概念に基づいて明かにした. 今回の実験では, 熟練者は基本的なビューの組合せ(例えば, サブシステムや機能的な役割)を意識しており, 効率の良い作業を行なっている. 今後は, このような補助的なビューをベースとした編集機能を支援システムに組み込みたいと考えている.

参考文献

[1] 西村他. プラン生成に基づく仕様獲得支援. プログラミングシンポジウム'91, A-34, 1991.
 [2] A. Finkelstein, et al. Viewpoints: A Framework for Integrating Multiple Perspectives in System Development. *Int. J. SE. KE.*, 2(1), 31-57, 1992.
 [3] C. Niskier, et al. A Pluralistic Knowledge-based approach to Software Specification Acquisition. *Proc. of ESEC'89*, 411-423, 1989.