

ソフトウェア開発のためのユーザ要求機能分析に関する一考察

加藤直孝

石川県工業試験場

國藤 進

北陸先端科学技術大学院大学

ソフトウェアの開発は、上流の設計工程が肝要といわれる。この工程では、創造的思考作業が主体であり、ユーザと開発者がお互いの要求を出し合い、両者の要求を歩み寄せながら設計を進めていく。しかし実際は、両者間にしばしば生じるコミュニケーション・ギャップが原因となり、ソフトウェアの設計品質の悪化あるいは生産性の低下を招いている。

そこで我々は、ユーザと開発者それぞれの要求獲得、両者間の合意形成およびユーザの要求を反映したソフトウェアの機能分析をコンピュータ支援する方法を提案する。本方法の特徴は、品質管理方法論の一つである品質展開アプローチ(QDA: Quality Deployment Approach)の考え方をヒントに、ユーザの要求をソフトウェア機能に直接反映させると同時に、開発者の要求もユーザにフィードバックさせる点にある。また両者の要求間に存在するトレードオフ分析にも検討を加えた。

A Study of Function Analysis Support Reflecting User's Requirement for Software Development

Naotaka Kato

Industrial Research
Institute of Ishikawa

Susumu Kunifuji

Japan Advanced Institute of
Science and Technology, Hokuriku

The design process of upper reaches is essential in software development. In this process, creative thinking is main work, user and developer show their own requirement each other and design work goes forward by their mutual concession. But actually, communication gaps often occurs during both sides, and it is a cause, design quality of software may become worsen or productivity of software may become lower.

We propose a computer support method which acquires requirement of both user and developer, makes their consensus formation and analyzes desirable software functions reflecting user's requirement.

The essential characteristic of our method is QDA(Quality Deployment Approach) that is one of quality control methodologies, and user's requirement is reflected directly in software functions and developer's requirement can give back to user simultaneously. Furthermore, we research analysis of trade-off problem which exists between user's requirement and developer's one.

1. はじめに

知的生産活動の例としてソフトウェア開発がよく取り上げられる。その課題として、生産性の向上と品質の追求が挙げられる。プログラミング工程のように従来から自動化が進められている下流工程に比べ、企画、要求仕様の設計などの上流工程は、人間の思考や表現の定式化が困難なうえ、複数人によるグループ協調作業が主体である。また、品質とはユーザがどの程度総合的に満足しているかを表す指標であり[1]、品質管理方法論では、開発の企画設計段階から「品質を作り込む」考え方が重要視されている[2][3]。このようなソフトウェア開発の上流工程における問題解決の場面には、意思決定および合意形成のプロセスが共通して存在する。そしてそれらは生産性や品質の向上を図る上で極めて重要な役割を持つ。

我々は、この意思決定および合意形成のプロセスをコンピュータ支援するべく、創造技法、システム工学、人工知能、グループウェア等の分野における要素技術を統合した支援ツールの研究開発を目指している[4-7, 20]。

本稿では、ソフトウェア開発を対象にユーザと開発者という立場や価値観が異なる者の要求獲得、両者間の合意形成およびユーザの要求を反映したソフトウェアの機能分析をコンピュータ支援する一方法を提案する。本方法は、ユーザ参画型の機能分析を基本思想としている点が、従来の開発者主導型のユーザ要求分析に対して大きく異なる。

2. 基本構想

我々は、ユーザと開発者のコミュニケーションギャップを埋めながら、両者の合意形成を支援するためには、システムティックな手法を取り入れたグループ協調作業支援環境が必要と考える。その場合、第一に考慮しなければならないことは、ユーザおよび開発者の要求が、個々の価値観に基づき、しかも個々の立場により異なる点である。

そこでまず、価値観の違いあるいは双方の歩み

寄りの度合いを表すための基本的な尺度として、重要度の考え方を導入する。すなわち、ユーザおよび開発者それぞれの要求を、要求に対して構造化された重要度木として取り扱うものとする。以下単に要求とはこの重要度木を意味する。

次にユーザの価値観での要求を開発者の価値観での要求へ変換することを考える。さらに開発者の価値観での要求をユーザの価値観での要求へ逆変換することを考える。こうすれば、お互いの価値観のすり合わせが行え、相手方の価値観での要求を自分が解釈可能な価値観での要求としてとらえ、これを互いに提示し合うことで合意形成の支援が可能と考える。この変換・逆変換プロセスに品質管理の方法論として知られる品質展開アプローチ (QDA; Quality Deployment Approach) [8][9]をヒントにして生まれた関連度行列[6][7]を利用する。

またユーザおよび開発者の要求には、多種多様な構造があり、「こちらたてればあちらたたず」式の競合が存在する。ある目的を改善するために、他の目的をどの程度犠牲にしなければならないかという価値判断をトレードオフという。合意形成プロセスでは、両者間に潜むトレードオフ[10]の解消を効率的に行うことが重要となる。そこでユーザおよび開発者の要求を獲得する段階および両者の合意形成を支援する段階の2段階でトレードオフ分析機構を持たせる。ここでトレードオフ分析とは、トレードオフ解消のための感度分析と定義する。以上の概念を図1に示す。

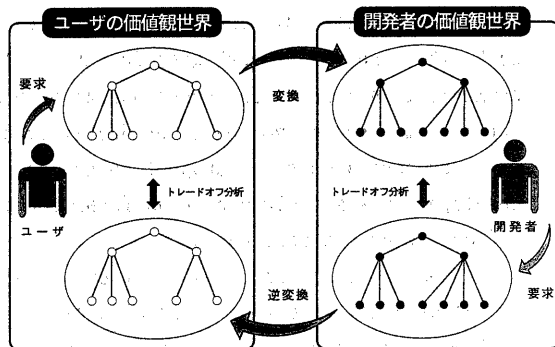


図1 基本構想の概念図

3. ユーザおよび開発者の要求獲得

要求獲得とは、要求構造を作成し、要求項目間の重要度を算出して要求に対する重要度木を形成するステップをいう。

3.1 要求構造の作成

要求構造を簡潔な言語情報の集合体としてボトムアップ的に形作るため、KJ法[11]を用いる。

まずユーザの顕在的、潜在的なソフトウェアに対する要求をありのままの原始情報（入力しやすい、見やすいなど）の形で提示してもらう。次に簡潔な表現の言語情報にラベル化、グルーピングを行ってKJ法A型図解に準じた階層的な要求構造としてまとめる。ラベル化のステップでは、過去の事例で収集したラベル群からのピックアップを可能にすることで作業能率の向上を図る。また開発者の要求についても、ユーザの要求を満足するために必要な機能について同様の手順を踏む。なお、原則的にKJ法主体で要求構造を作成するが、開発者においては問題意識や対象知識に対する知見レベル、あるいはKJ法の習熟レベルを考慮してISM (Interpretive Structural Modeling) 法による要求の構造化も行えるようにする。

3.2 重要度の算出

一般に要求の各項目は、尺度がそれぞれ異なり、かつ主観的特性をもつ。このような項目間の重要度算出に適した方法として階層化意思決定法 (AHP: Analytic Hierarchy Process) [12] を利用する。これは項目間の一対比較を行うことで相対的重要度を整合性のとれた比尺度値として求める方法である。しかし、AHPの問題点として次の2点が指摘されている。

①比較項目数が多くなるにつれて一対比較作業が煩雑となり作業者の精神的負担が増す。

②要求項目間には、AHPの適用条件である相互独立性がすべてあるとは限らず、従属性が存在し得る。この従属性を完全に取り払うことは一般に困難と考えられる。

そこで、①の対策に不完全一対比較法[13][14]を、②の対策に非加法的ウェイトを用いたAHP[15]をそれぞれ適用する。

(1) 不完全一対比較法

一対比較に自信がない、あるいはわからない部分を整合性があるように補完し、全体として整合性のある完全な一対比較行列を構成する技法である。一対比較の回数を減らす効果があり、比較項目数の増加に伴う一対比較作業の負担を減らすことができる。

(2) 非加法的ウェイトを用いたAHP

項目間に従属性がある場合や、新たに評価基準が加わった場合に、本来の重要度が影響を受けて変化し、順位の逆転現象が起こりうる。この理由は重要度の和が1になるように基準化された加法的測度であるためと考えられている。

そこで重要度の最大値が1になるように基準化する「非加法的ウェイトを用いたAHP」を適用して順位の逆転現象の発生を解消する。具体的には、通常のAHPで重要度を求めた後、重要度の最大値が1になるような基準化を行う。

3.3 要求項目間のトレードオフ分析

ユーザあるいは開発者が、以上の手順で求めた重要度に対し、予想との食い違いあるいは不満を抱いた場合、満足するような重要度に改善しなければならない。また後述の合意形成支援段階では、お互いの要求の歩み寄りを進めるため、重要度の再配分を行わなければならない。そのために要求項目間のトレードオフ分析が必要となる。

AHPでは、一対比較行列 $A = [a_{ij}]$ (一対比較値 a_{ij} , $a_{ji} = 1/a_{ij}$, $1 \leq i \leq n$, $1 \leq j \leq n$) の最大固有値 $\lambda_{\max}$ に対する固有ベクトル $\omega =$

$(\omega_1, \omega_2, \dots, \omega_n)$ を重要度として取り扱っている。ここで整合性が完全である場合の a_{ij} と ω の関係は式(1)で表される[16]。

$$a_{ij} = \omega_i / \omega_j \quad (1)$$

これより、例えば k 番目 ($1 \leq k \leq n$) の重要度の値

ω_k を上下した場合、整合性を保つためには、これに伴って a_{kj} ($=\omega_k/\omega_j$) および a_{ik} ($=\omega_i/\omega_k$) の値が変更の対象候補となることがわかる。この従属関係は、一対比較行列の独立な要素の組を選んで一対比較数を低減させるために用いられる場合[17]もあるが、ここではこの従属関係をトレードオフの操作対象となる a_{ij} の選択に用いる。そこで、これら a_{ij} の候補群をシステム側から作業者に提示し、その中でトレードオフとして現在最も注目している一対比較値 a_{ij} の値を作業者が変更し、再び重要度の計算と整合度のチェックを行う。ここでのトレードオフ分析はこのような手順を繰り返す感度分析で行う。

4. ソフトウェアの要求機能分析

4. 1 ユーザ要求の開発者側への変換

ここでは、ユーザの価値観での要求を開発者の価値観での要求へ変換する方法について述べる。

一般にユーザの価値観は主観的で定性的な感性指向の世界にあり、一方開発側の価値観は、客観的で定量的な機能指向の世界にあると考えられる。例えば、ユーザの要求とは、画面が見やすい、操作しやすい、処理速度が早いなどであり、開発者の要求とは、メニュー操作機能、マウス操作機能、高速演算処理機能など機能的な要求である。

そこで、ユーザの要求項目を縦軸、開発者の要求項目を横軸として両者を対応させ、その対応の強さを表す関連度表を作成する。関連度には、関連の強さに応じて◎ (大きい)、○ (中くらい)、△ (小さい) のような記号で順序付けを行う。さらに◎: 5点、○: 3点、△: 1点、記号無し: 0点のような点数付けを行うことで関連度行列を作成する。図2に関連度表の一例を示す。

この関連度行列を利用することにより、ユーザの要求が開発者の要求の次元に変換される。すなわち、前述の3. 2節で求めたユーザの要求の重要度ベクトルを u とすると、開発者側に変換された要求重要度ベクトル v' は式(2)で表される。

$$v' = W u \quad (2)$$

ここで、 W は関連度行列の転置行列である。

従って式(2)により、開発者はユーザの要求を自分側の要求の世界へ変換し、ユーザが望んでいる機能とその重点配分を知ることが可能となる。

式(2)からもわかるように関連度の与え方が直接結果に影響を及ぼすことから、関連度の値の設定は慎重に行わなければならない。

そこで、ユーザの要求を極力反映させるため、関連度の決め方について次のルールを適用する。

- ①関連度表の横軸に位置する開発者要求(ソフトウェア機能)の用語の意味について誤解を生じないように、まず開発者からユーザに対してこれらの用語について意味とその範囲を説明する。
- ②ユーザが主体的に関連の有る無し、有る場合にはその関連度を決定し、関連度行列を作成する。不明な箇所は開発者の助言を得た上で決定する。
- ③開発者がその後チェックし、矛盾点や改良点があればユーザに提言し、ユーザが修正する。

Correlation Matrix説明											
強い対応	◎	5									
対応	○	3									
対応が予想	△	1									
			誤操作チェック機能	障害迅速対策機能	バックアップ機能	アンドウ機能	ファイルロック機能	パスワード機能	ロギング機能	ファイル変換機能	データリンク機能
通常操作でトラブルがない	◎		○	○	○						
誤操作時に警告が表示される	◎	◎									
誤操作によるデータ破壊がない	◎		○	◎	○	△					
障害原因を突き止め易い			△						◎		
データの復旧が容易である			△	◎	○				△		
データが不正利用されない						◎	◎	○			
他ソフトとデータが連携できる										○	◎

図2 関連度表(一部分)の例

4. 2 開発者要求のユーザ側への逆変換

次に開発者の要求をユーザへフィードバックさせるための逆変換を行う方法について述べる。

式(2)の関連度転置行列 W は、一般に長方形列であり、一般逆行列の理論によると、 W がMoore-Penrose一般逆行列の条件[18]を満たす場合、 W の逆行列が一意に導かれる。すなわち、開発者の要求重要度ベクトルを v とすると、ユーザ側へ変

換された要求重要度ベクトル u' は式(3)で表される。

$$u' = W^{-1} v \quad (3)$$

4. 3 トレードオフ分析による要求機能分析

合意形成のプロセスに入る前の段階では、式(2)の u および v' はユーザの初期重要度ベクトル、また式(3)の v および u' は開発者の初期重要度ベクトルを表している。この段階では一般にベクトル u と u' （あるいはベクトル v と v' ）は異なるベクトル方向を持つ。

そこでお互いが合意形成に向けて歩み寄るべく、開発者が Δv およびユーザが Δu だけ重要度を変化させたとする。この Δv および Δu の変更量は、少ない方が望まれる。以上のパラメータを用いて次の指標関数を定義する。

ユーザに提示する指標関数

$$S(\Delta u, \Delta u') = (u - \Delta u)^t (u' - \Delta u') \quad (4)$$

$$Ru(\Delta u) = u^t \Delta u \quad (5)$$

$$Rd(\Delta u') = u'^t \Delta u' \quad (6)$$

$$C = S / (\alpha Ru + \beta Rd) \quad (7)$$

$$(\alpha + \beta = 1, \alpha > 0, \beta > 0)$$

開発者に提示する指標関数

$$S(\Delta v, \Delta v') = (v - \Delta v)^t (v' - \Delta v') \quad (8)$$

$$Ru(\Delta v') = v'^t \Delta v' \quad (9)$$

$$Rd(\Delta v) = v^t \Delta v \quad (10)$$

$$C = S / (\alpha Ru + \beta Rd) \quad (11)$$

ここで、 S はユーザと開発者の価値観の接近度、 Rd は開発者の変更度、 Ru はユーザの変更度、 C は両者の合意形成度を表す。 α および β は、両者が置かれた状況に応じた適当な加重係数である。

そして、式(4)～(7)の値をユーザに、式(8)～(11)の値を開発者にグラフィカルな画面構成で同時に提示する。これを見ながら両者が合意形成に向けて要求項目間の重要度をインタラクティブに変更する。この変更は直ちに式(4)～(11)で与え

た合意形成の指標値に反映される。このような感度分析を繰り返すことで両者の合意形成が支援されると同時に、開発者にとってユーザの要求を反映したソフトウェアの要求機能分析が可能となる。

5. システム構成

図3にシステム構成を示す。本システムは、要求獲得モジュールと要求機能分析モジュールの2つのモジュールから構成される。要求獲得モジュールはさらに次の3つのサブモジュールから構成される。①要求抽出②要求構造化③AHPによる重要度計算。ここで①と②にはKJ法をコンピュータ上でツール化したI SOP[19]を利用する。また②にはISMのツール[20]も利用できる。

一方、要求機能分析モジュールはさらに次の3つのサブモジュールから構成される。①関連度行列の決定②ユーザと開発者間の要求重要度の変換・逆変換③要求機能分析。

各モジュールは、ウインドウシステム上で必要に応じて適時起動することができ、対話型のシステム操作環境を持つ。例えば操作を進めるにつれて、新しい要求項目の存在や、逆に無視できる要求項目の存在に気づいた場合、適宜追加または修正が行える。図4に画面例を示す。

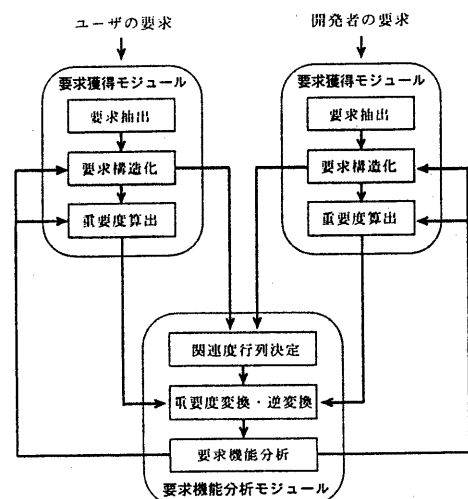


図3 システム構成

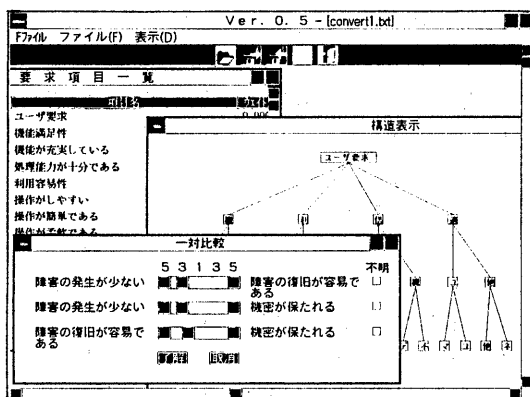


図4 対話型ウィンドウの画面例

6. おわりに

ソフトウェア開発を対象にユーザと開発者それぞれの要求獲得、両者の合意形成およびユーザの要求を反映したソフトウェアの機能分析を支援する方法について述べた。特徴として、ユーザの要求と開発者の要求をお互いが解釈可能な次元にそれぞれ変換して提示し合う環境を提供し、さらに両者間のトレードオフ分析機構を持つ。

現在パーソナルコンピュータ上のMS-Windows環境でプロトタイプシステムの開発を進めている。今後プロトタイプの完成に伴い、適用事例に基づく評価を行う予定である。

参考文献

- [1] 情報処理振興事業協会技術センター編：品質機能展開による高品質ソフトウェアの開発手法（解説編），コンピュータ・エージ社(1989)
- [2] 赤尾洋二：品質展開入門，日科技連出版(1990)
- [3] D. C. Gause, G. M. Weinberg：要求仕様の探検学，共立出版社(1993)
- [4] 國藤進，加藤直孝，田村賢之：関連度行列に基づく合意形成支援システムの研究，計測自動制御学会第17回システム工学部会研究会(1995. 1)
- [5] 田村賢之，加藤直孝，長田英希，國藤進：ユーザの要求を反映する合意形成支援システムの研究開発構想，計測自動制御学会第15回システム工

学部会研究会，pp. 9-16(1994)

[6] 加藤直孝，國藤進，上田芳弘：ソフトウェア開発における要求機能設計支援システムの研究開発構想，計測自動制御学会合同シンポジウム，p. 277-282(1993. 11. 11)

[7] 長田英希，國藤進：主観的評価－客観的評価の変換機能を持つ意思決定支援システム，第7回人工知能学会全国大会，pp. 699-702(1993)

[8] 大森晃：ソフトウェア品質展開の枠組み設計法，情報処理学会論文誌，Vol. 33, No. 12(1992)

[9] 大森晃：ソフトウェア品質展開アプローチ，システム／制御／情報，Vol. 37, No. 5，pp. 269-276(1993)

[10] G. M. Weinberg：システムづくりの人間学，共立出版社(1986)

[11] 川喜田二郎：発想法，中央公論社(1967)

[12] 刀根薫：ゲーム感覚意思決定法，日科技連出版社(1986)

[13] Harker, P. T. : "Alternative Modes of Questioning in the Analytic Hierarchy Process", Math. Modeling, 9, 353-360(1987)

[14] 竹田英二：不完全一対比較行列におけるAHPウェイトの計算法，オペレーションズ・リサーチ，Vol. 34, No. 4, pp. 169-172(1989)

[15] 市橋秀友：最大値を1とする重要度の基準化について，第5回ファジィシステムシンポジウム講演論文集，PP. 307-32(1989)

[16] 木下栄蔵：意思決定論入門，啓学出版(1992)

[17] 宮城隼夫，平良直之，山下勝己：加法形の一対比較行列を用いた多目的意思決定の一手法，計測自動制御学会合同シンポジウム，pp. 119-122(1994. 10. 7)

[18] 岡本良夫（武者利光監修）：逆問題とその解き方，オーム社(1992)

[19] ITEC：コンピュータ支援による智的問題解決技法ISO P，國藤研究室主催発想支援ツールシンポジウム資料集，三菱総合研究所(1992. 12. 12)

[20] 佐藤雅之，國藤進：KT法支援システムの実現に向けて，國藤研究室主催発想支援ツールシンポジウム資料集，三菱総合研究所(1992. 12. 13)