

設計履歴とドキュメントを用いたソフトウェア変更支援

2U-4

村瀬 康人 深澤 良彰 門倉 敏夫

早稲田大学 理工学部

1 はじめに

近年、ソフトウェアの増大に伴ってソフトウェア変更支援に対する要求が高まっている。変更時におけるソフトウェアの理解や変更解析などを行う際にドキュメントの参照は有効な手段である。変更時に、変更担当者は各種ドキュメントの対応をとりながらソフトウェアの理解をしている。例えば、要求仕様書に書かれている要求に対応する機能は、機能仕様書のどこに書かれているか、またそれはソースコードのどこで実現されているかというような対応がとられている。この対応を基に、変更の際に設計の判断間の依存関係を調べ、それを基に波及解析を行ったり、変更の方針を決めたりする。

また、ソフトウェアの設計判断過程を履歴として残し、それを後に用いる事によりソフトウェア開発や保守の支援を行う研究が行われている。この設計履歴を用いる事により、ソフトウェアの有効な変更支援が期待できる。

我々は、ソフトウェア変更時におけるドキュメントの対応づけとその参照、及び設計の判断間における関係把握と確認を容易化するシステムを考案した。これにより、ソフトウェア変更担当者はドキュメント検索や、設計履歴間の依存関係確認のための設計トレースの様な単純な機械的な作業から解放され、ソフトウェア及び変更理解や変更方針決定に専念することができる。

2 基本構想

本システムは、設計フェーズ、ドキュメント対応フェーズ、変更フェーズにおいて用いられる。

設計フェーズでは、開発者は設計履歴とドキュメントを記録し、それらの利用関係を入力しながら設計を進める。この設計履歴とは、どのドキュメントのどの部分に対応する問題であるか、またそれに対する解決候補案、候補案選択の際に考慮に入れなければならない属性、その候補案に課せられた制約条件を集めたものである。また、ドキュメントは設計作業の生産物である。

ドキュメント対応フェーズでは、例えば機能仕様書で指定された変更対応箇所を要求仕様書やソースコードに対応させる。これにより変更担当者は、各種ドキュメント及び設計履歴から必要な箇所を容易に探し出す事ができる。

変更フェーズでは、ドキュメント対応フェーズでドキュメントを参照し、実際の変更方針を決めた後、実際に変更すべき設計履歴の位置を入力する。本システムは、指

示された設計履歴に対応するドキュメントと、その変更により設計中の判断に影響を与えられている設計履歴、及びその影響理由を提示する。また、この影響により利用できなくなった候補案に対して代替案の利用可能性を調べ、利用可能なものを提示する。

3 システム詳細

3.1 システム構成

本システムの構成を図1に示す。本システムは大きく分けてユーザインターフェース部、設計履歴取得部、変更支援部から構成される。

ユーザインターフェース部はユーザからの設計履歴の入力や、解析結果の提示などを行う。設計履歴取得部は本システムの定める設計モデルに従って設計履歴を入力、蓄積する部分であり、設計フェーズで用いられる。変更支援部は変更箇所の特定とその変更の波及解析を行なう部分であり、ドキュメント対応フェーズ、変更フェーズを支援する。以下でそれぞれをフェーズに従って説明する。

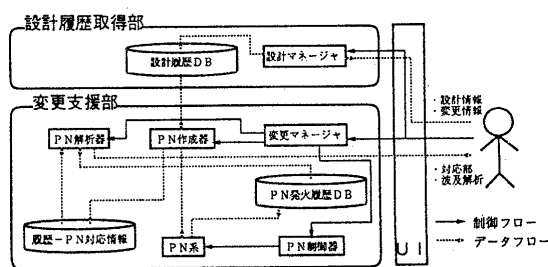


図1 システム構成

3.2 設計フェーズ

設計履歴を設計マネージャによりユーザと対話しながら獲得する。設計過程は、問題提起、問題の解決作業の連鎖と捉える事ができる。そこで本システムでは設計履歴として、問題、問題解決の候補案、候補案選択の際に課せられた制約、MCDM法 (Multiple Criteria Decision Method)[1] による解決策選択、ドキュメントを記録する。MCDM法は複数の候補案からの有効な代替案選択法として知られている。この方法では、選択の際に考慮に入れなければならない属性を洗い出し、各属性に対して評価基準を設定し、各々の属性に対して重みをつけ、その重みを考慮しながら最も評価の高い候補案を選択する。

この設計履歴を、親子リンクによりその問題の入力、及び出力となったドキュメントと関係づけ、制約リンクにより候補案と外部から求められる制約を関係づける。

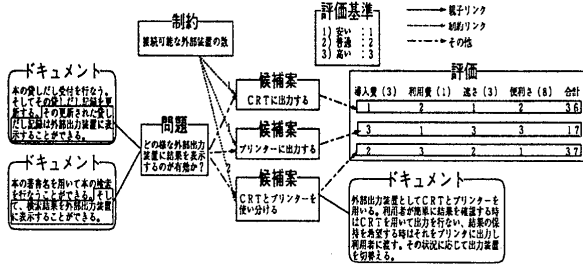


図2 設計履歴

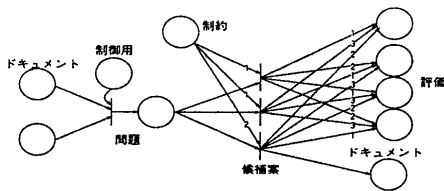


図3 ペトリネット

制約は MCDM 法で洗い出された属性または制約用で作成された属性から作られる。図2に設計履歴の例を示す。

本システムでは以後の解析を行なうために設計履歴とドキュメントの関係をペトリネット (以下PNと略す) を用いてモデル化する。PNは次のように表現できる。

$$N = \langle P, T, F, W, M \rangle$$

ここでPはプレース集合、Tはトランジション集合、Fはアーク集合、Wはアークの重み、Mは初期トークンの配置を表している。PNの各構成要素に、本システムの情報を次のように割り当てる。

- P: ドキュメント (D)、属性 (A)
- T: 問題 (B)、候補案 (H)
- F: 親子関係 (O)、制約 (C)、評価 (E)
- W: 制約、評価値 (その他は1)    M: 随時決定

ただし、親子関係は問題とその候補案との関係を含む。この他に、ある問題を発火禁止にしたり、発火回数を高々1回に制限するための制御用プレース (G) がある。図3に図2の設計履歴をPNに対応させた例を示す。

### 3.3 ドキュメント対応フェーズ

あるドキュメントで指定された変更対応箇所を、そのドキュメント作成の基となった上位ドキュメントとそのドキュメントを基に作られた下位ドキュメントに親子アークを辿ることにより、他のドキュメントと対応をとる。

このためにPN作成器において、設計履歴DBの内容をもとに次の様なPNを作成する。

$$N_1 = \langle DUAUG, BUH, O_{rev}UCUE, W, M_1 \rangle$$

$O_{rev}$  は親子アークの向きを反転させたアーク集合である。また、 $M_1$  は、下位ドキュメントの存在しないドキュメント (以下最終ドキュメント) に対応するプレースと、対応箇所と指定された箇所以外の制御用プレースと属性プ

レースに充分数トークンを配置したものである。トークンの充分数とは、そのプレースからの出力アークの重みの和である。このPNをPN系において発火させ、発火しなかったトランジションと、必要数トークンが溜らなかつたプレースに対応する上位ドキュメントの存在しないドキュメント (以下初期ドキュメント) が対応箇所である。トークンの必要数とは、そのプレースへの入力アークの重みの和である。この時トランジションの発火順はPN制御器により制御される。

次に、その初期ドキュメントを基に作成したドキュメントを辿るために、以下の様なPNを作成する。

$$N_2 = \langle DUAUG, BUH, OUCUE, W, M_2 \rangle$$

ここで $M_2$ は、 $N_1$ を発火させた結果、初期ドキュメントに辿りつくことができたプレースにトークンを充分数配置したものである。 $N_2$ を発火させた結果、発火できなかったトランジションが対応箇所である。この結果は、PN解析器により設計PN対応情報を用いて設計履歴に対応を取り、変更担当者に提示する。

### 3.4 変更フェーズ

ドキュメント対応フェーズで絞り込まれたドキュメントと設計履歴を用いて実際の変更箇所を特定し、それに対する波及解析を行なう。

実際には、この変更箇所に対応する設計履歴を発火禁止とし、初期ドキュメントに対応するプレースにトークンを充分数配置する。そこで $N_3$ の様なPNをPN作成器により作成しPN系で発火させる。これにより発火できなかったトランジションが変更影響箇所である。

$$N_3 = \langle DUAUG, BUH, OUCUE, W, M_3 \rangle$$

次に、設計判断の依存により影響を受けている箇所について、その影響理由を調べる。このためには、制約を満たしていないプレースに対して、充分数のトークンを配置する。そしてその発火系列に制約をマークしていき、最終ドキュメントに波及させる。この情報はPN解析器により変更担当者に提示される。

また、制約が満たされなくなっている候補案に対して他の代替案の利用可能性を調べ、もし利用可能であれば、その中で最も評価値の高い候補案を提示することにより、設計変更に対する代替案選択を容易化する。

## 4 おわりに

設計履歴とドキュメントを用いたソフトウェアの変更支援手法について報告した。これにより、ソフトウェア変更時に起こるドキュメント間の対応づけや変更の波及解析などの作業を支援する事ができ、変更担当者を機械的な単純な作業から開放することができる。

現在本システムを製作中であり、本システムの有効性について評価していきたいと思う。

### 参考文献

[1] 渡部和雄, Clyde W. Holsapple, Andrew B. Whinston: グループ MCDM 法に基づく日本の意思決定支援方式, 情処学情報システム研報 No.25(1989)