

ソフトウェア開発における潜在要求発見の支援手法

鶴貝康男[†] 高橋竜一[‡] 深澤良彰[‡]早稲田大学[‡]

1. 背景

ソフトウェア開発における要求獲得を, 本研究では, ステークホルダが抱える問題に対する「解決したい」という要求を定義する工程と定義する. 近年の要求獲得では, 同じドメインのソフトウェアとの競争に勝つために, 他のソフトウェアに存在しない新しい要求を獲得する事が求められる.

2. 問題

Maiden ら^[1]は, Web サービスを提示し, それらが何の要求を満たせるか, という観点から新しい要求を誘発させる手法を提案している. しかし, これらの候補となる機能は, Web サービスという既存のソフトウェアが持つ機能であり, そこから他のソフトウェアには存在しない新しい要求を誘発する事は困難である.

3. 提案

そこで本手法では, 複数の Web サービスを組み合わせて出来る他のソフトウェアに無い新しい機能を, 候補として提示する. この時 Web サービスの組み合わせ方は膨大な数になる場合があり, それらを元に新しい要求を見つける事は大変な時間が掛かる作業である. また候補の中からステークホルダが新しい要求を獲得するためには, それぞれの候補がどんな機能を表すのかが理解しやすい方が好ましい. そのため本手法では, Web サービスの全ての組み合わせから, 他のソフトウェアが持つ機能, 要求を誘発させる可能性が低い機能, そして同じ機能を表す組み合わせの中で, その機能を理解する事がより難しいと考えられる組み合わせをフィルタリングし, 候補の数を減らす. 更に残りの候補を, それぞれの候補がユーザに与える価値によってグループ化する事で, ステークホルダのそれぞれの機能への理解を支援する.

4. 手法

本手法は図1に示す3つのステップで構成される. まず Web サービスにおいてインプットとアウトプットの依存関係がある組み合わせを全て

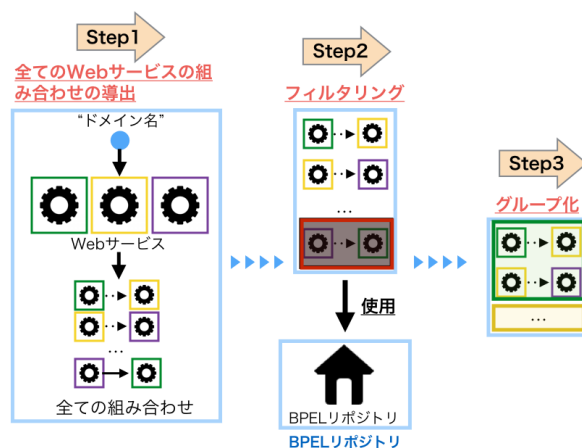


図1. オーバービュー

導出する. そしてそれらの組み合わせから, 他のソフトウェアによって実装済みの機能, ユーザがコストの高いデータを与えなければいけない機能, 無駄な処理を含む組み合わせをフィルタリングする. 最後に残りの候補を, それぞれの候補が導出するアウトプットとそれ以外に現実世界に与える効果によって, グループ化する.

4.1. Web サービスの組み合わせの導出

Web サービスにおいてインプットとアウトプットの依存関係がある組み合わせを全て導出する. それらの依存関係を表すグラフとして SDG^[2]と呼ばれる Web サービス同士の依存関係を表すグラフ使い, その全てのサブグラフを取得する.

4.2. フィルタリング

次に取得した全ての組み合わせから, 3つのルールによってフィルタリングを行う.

ルール1: 他のソフトウェアによって実装済みの機能の識別

本手法の目的の1つとして, 他のソフトウェアに無い機能を提示する事が挙げられる. そのため, その目的を満たさない機能をフィルタリングする. 本手法では, BPEL リポジトリに存在する BPEL が表す機能の集合を BR とした時に $Combination \in BR$ となる $Combination$ を, 他のソフトウェアが持つ機能として扱う.

ルール2: ユーザがコストの高いデータを与え

Supporting to elicit "Creative" requirement for software development

[†]Yasuo Tsurugai, Ryuichi Takahashi and Yoshiaki Fukazawa[‡]Waseda University

なければいけない機能の識別

ユーザがインプットのコストが高い機能を使いたいと望む事は少なく、その機能はニーズが無い可能性が高い機能と言える。本手法の目的の1つとして、新しい要求を誘発させる機能を提示する事が挙げられる。ニーズが無い機能は、その機能が満たす要求が無い事を指すため、要求を誘発させる可能性が低いと言える。よって、ニーズが無い可能性が高いと言えるユーザのインプットのコストが高い機能をフィルタリングする。ここで、Web サービスのインプットの中には、他の Web サービスが機械的に作成する事を想定したデータが存在する。機械的に作成される事を想定したデータをユーザが与える事は、大変なコストが掛かる作業である。本手法では、BPEL リポジトリに存在する全ての BPEL において Web サービスが与えているデータを WSD とした時に $Data \in WSD$ となる Data を、Web サービスが機械的に作成する事を想定したデータとし、ユーザが与えるコストが高いインプットとして扱う。

ルール 3: 無駄な処理を含む組み合わせの識別

Web サービスの組み合わせには、データの循環を含むものが存在する。図 2 にデータの循環を含む組み合わせの例を示す。

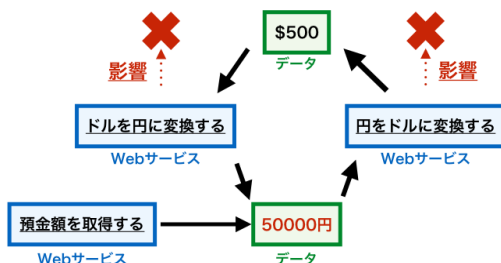


図 2. データの循環を含む組み合わせの例
その循環の元となるデータが常に同じ値を持ち、データの循環を引き起こす個々の Web サービスがアウトプットの導出以外に現実世界に与える効果が無い時、データの循環に含まれる処理は無駄な処理であると言える。本手法の目的の1つとして、同じ機能を表す組み合わせの中では、より理解し易いものを提示する事が挙げられる。ここでデータの循環を含む組み合わせと、そこから無駄な処理を除いた組み合わせは同じ機能を有す、よってそれらの中で、より長い組み合わせとなり、理解し難いと考えられる無駄な処理を含む組み合わせをフィルタリングする。また循環の元となるデータが常に同じ値を持つかを確かめるために、データの循環に含まれる Web サービスを実行し、循環の元となるデータが持つ値を観測する。

しかし Web サービスの実行は様々なコストが掛かるため、実行する Web サービスの数を出来る限り減らす事が必要である。そのため本手法では、あるデータの循環を含む組み合わせの実行結果を、同じデータの循環を含む別の組み合わせに再利用する事によって、実行する Web サービスを最小限に抑える事を実現する。

4.3. グループ化

最後に、残りの組み合わせから、ユーザに与える価値が同じ組み合わせを1つのグループとする事によって、ステークホルダがそれぞれの機能を理解する事を支援する。ユーザに与える価値は、アウトプットとそれ以外に現実世界に与える効果で表現される。

5. 評価

5.1. 実験方法

本手法によってフィルタリングした機能とその結果残った機能について、情報学専攻の大学院生3人に対して、それぞれの機能が「必要か?」という質問に5段階評価(1が、「必要だと思う」を表す)で回答して貰う。平均して3以下の評価値を持つ機能をニーズがある機能として扱い、本手法がどれだけそれらの機能をフィルタリングしてしまったのか、また獲得出来たのかを Precision と Recall によって評価する。

5.2. 実験結果

実験の結果、Precision が 0.63, Recall が 0.8 と共に高い値を表した。また本手法によって、7385 個の候補を 51 個まで削減する事が出来た。

6. まとめ

本稿では、Web サービスを組み合わせで出来る機能を候補として提示する事による新しい要求の獲得の支援手法を提案した。この時、Web サービスの組み合わせ方は膨大な数になる場合があるため、3つのルールによって候補をフィルタリングした。その結果、本手法によって7385個から51個まで候補の数を削減する事が出来た。今後はフィルタリングのルールを追加し、ニーズの無い機能をより多くフィルタリングする事が求められる。

7. 参考文献

- [1]K. Zachos, N. Maiden, Inventing Requirements from Software- An Empirical Investigation with Web Services, RE, 2008
- [2]Q.A.Liang, S.Y.W.Su, AND/OR Graph and Search Algorithm for Discovering Composite Web Services, IJWSR, 2005