

システムの分類による impact analysis 導入の課題整理

長谷川 勇^{†1}

システムインテグレータやウェブ開発企業における開発の大半は、既存のシステムのエンハンス開発であるため、impact analysis に関する解析技術は重要である。しかし開発現場への導入には、責任範囲の問題、分析結果のデータ量爆発、ツールの不足といった、いくつかの課題があり、まだまだ進んでいない。分析対象のシステムを規模・構成言語・コンポーネント間結合度の観点で分類すると、複数言語で構成された結合度の高いシステムへの導入が困難であることが分かる。企業における開発ではこの領域のシステムが多いため、課題の解決が必要である。

Considering problems to apply impact analysis by classifying systems

ISAMU HASEGAWA^{†1}

The great portion of development in system integrators, and web application developers is enhancement development of the existing systems. Therefore, impact analysis technology is important. However the technology has not been applied yet, because of some problems like the responsibility range, data volume explosion of analysis, and insufficient tool. We classified target systems according to the viewpoint of scale, language, component coupling, and noticed that it is difficult to apply to the multi language, and highly coupled systems.

1. はじめに

システムインテグレータやウェブ開発企業におけるシステム開発の現場では、システム開発の大半は既存のシステムのエンハンス開発である。このため、企業におけるシステム開発において、impact analysis に関する解析技術の重要性は論を待たない。

一方、それらのシステム開発の現場において、impact analysis の研究成果はほとんど使われていない。

ここでは、開発現場に impact analysis の技術を適用する上での課題を述べ、次に解析対象のシステムの分類を行い、各分野への適用時の問題を考える。

2. impact analysis 適用時の問題

なぜシステム開発の現場では、impact analysis に関する解析技術の導入が進まないのだろうか。筆者らの現場では主に以下の3点の問題が原因となっていた。

- 責任範囲
- 分析結果のデータ量爆発
- ツールの不足

ここでは、これらの問題を述べる。

2.1 責任範囲

開発するシステムが巨大になるにつれ、システムを

構成する各コンポーネントを担当するチームの組織上の距離も遠くなっていく。例えば、比較的規模の小さいシステムであれば、コンポーネントを担当する各チームは同じ部署、課であることが多く、チーム間の交流も日頃から行われていることが多い。一方、システムが巨大であれば、担当する各チームは、別の部署あるいは別の会社、別の国であることもあり、チーム間の交流は減ってしまう。

このため、例えばチーム A の担当するコンポーネント A の変更による、チーム B の担当するコンポーネント B への影響調査を行う場合、チーム A は変更点をチーム B に伝えるのみで、コンポーネント B の調査を行うのはチーム B である。

このため、広範囲の影響調査を行うために、新しく技術やツールを導入するモチベーションは低く、導入が進まない原因となる。

2.2 分析結果のデータ量の爆発

システムに変更を行う際の影響調査においてツールなどを利用する場合、ユーザが期待するのは影響があるすべての箇所であり、分析結果に false negative があることは想定していない。このため、巨大なシステムで影響調査を行うと、分析結果のデータ量が大きくなりすぎて実用にならないことがある。

現場のツール利用者はボタン一発で「正しい結果」が出ることを想定するが、ある程度の false negative

^{†1} zigsow 株式会社
zigsow Inc.

表 1 分析対象システムの分類

システムの規模		小		大	
構成言語		一	多	一	多
コンポーネント間結合度	低	1	3	5	7
	高	2	4	6	8

を許容したり、分析を段階に分けて実施するなど、利用者側が使い方に経験やスキルを問われ、ツール導入の敷居が高いことがある。

2.3 ツールの不足

近年の開発において、そのシステムが単一の言語で作られることはほとんどない。さらにアプリケーションフレームワークでは設定ファイルにより呼び出すプログラムを制御するため、正確な分析結果を得るためには、これらの設定ファイルも解析対象とする必要がある。このため impact analysis の解析対象は、システムを構成する複数の言語、フレームワークの設定ファイル、RDBMS のスキーマなど多岐にわたる。

例えば COBOL の著名な解析ツールである、Micro Focus Revolve¹⁾ では、COBOL 言語だけでなく、IBM メインフレームの JCL も含めた解析を行う。この例からも、単一言語の解析では意味がないことが言えるであろう。

このため、impact analysis の適用を考えると、以下 2 点が問題となる。

- マイナーな言語・フレームワークを使う場合、解析できるツールがない。
- 特定の言語をサポートするツールを結合できる必要がある。

3. 課題の整理

前章で述べた問題は、必ずしも一般に適用される問題ではない。ここでは、impact analysis の分析対象のシステムを分類し、どの領域のシステムにどのような問題があるかを考える。

まずシステムを以下の観点で分類する (表 1)。

- システムの規模
- 構成言語: そのシステムが単一の言語で構成されるか、複数の言語で構成されるか (前述の通り、ここで言う言語にはフレームワークなどの要素も含まれる)。
- 結合度: コンポーネント間の結合度。

以下では、これらの各領域において、どのような問題や解決策があるかを考える。

3.1 システムの規模

規模の大きいシステム (表 1 の 1, 3, 5, 7) の場合、前述の責任範囲の問題が起きる。しかし、この問題は

技術的な解決方法よりは意識改革などが必要なため、ここでは考えない。

3.2 単一言語のシステム

単一言語によるシステム (表 1 の 1, 2, 5, 6) の場合、Eclipse や Microsoft Visual Studio をはじめとする統合開発環境や、前述の Revolve などの解析ツールによる impact analysis は、すでに開発現場において使われている。

コンポーネント間の結合度が高い場合 (表 1 の 2, 6)、前述の解析結果のデータ量爆発の問題が起きるが、今後研究が進みツールが改善されていくことで、自然と解決に向かうと考えられる。

3.3 複数言語だが結合度の低いシステム

複数言語だがコンポーネント間の結合度の低いシステム (表 1 の 3, 7) の場合、前述のツール不足の問題が起きる。しかし、単一のコンポーネントに絞ると単一の言語で構成されることが多い。このため、この領域は利用者側の工夫で、単一言語であり結合度の低いシステム (表 1 の 1, 5) として解析することができる。この領域にはあまり問題がないため、ツール導入の敷居も低いと考えられる。

ただし、利用者側の工夫を一般的なユーザに求めるのはまだまだ難しいため、利用者の教育や意識改革、あるいは専任の部隊による代行やコンサルティングは必要となる。

3.4 複数言語で結合度の高いシステム

複数言語であり、コンポーネント間の結合度の高いシステム (表 1 の 4, 8) の場合、前述の利用者側の工夫では単一言語のシステムとして解析することができない。このため、この領域では解析結果のデータ量爆発や、ツールの不足といった問題があり、impact analysis の導入は難しい。

さらに、企業におけるシステム開発の現場において、システムの複雑度の高いこの領域が、最も impact analysis の必要性が高く、問題解決の優先度が高いと考えている。

4. impact analysis 技術開発への期待

上記のとおり、著者らのグループでは、複数言語で結合度の高いシステムにおける、解析結果のデータ量爆発や、ツールの不足といった課題の解決が急務であると考えている。この辺りの解決策を議論したい。

参考文献

- 1) : Micro Focus Revolve. <http://www.microfocus.co.jp/support/techtips/revolve.asp>.