Feature Location 技術の開発現場での活用に向けて

岡田 譲二[†1]　鹿島 悠[†2]　坂田 祐司[†1]

多くの Feature Location ツールが提案されているにもかかわらず，システムインテグレータによる開発の現場では活用されていない．本稿では，なぜ開発現場で Feature Location ツールが導入されないのかを損失回避の考えに基づいて述べ，開発現場で活用できる Feature Location ツールの方向性を述べる．

Towards Use of Feature Location at the Actual Development

Joji Okada[†1]　Yu Kashima[†2]　Yuji Sakata[†1]

Many people have proposed many kinds of Feature Location tool/method. However, at the actual development of System Integrators, Feature Location tool/method is not utilized yet. This paper describes why Feature Location tool/method is not utilized at the actual development.

1. はじめに

近年システム開発は新規開発より既存システムの保守や追加開発，システム更改が主流となっている．システムの保守や追加開発においては，変更要求に対応する既存のソースコードを特定する作業が重要である．これらを支援する技術として，機能に対応するコード片を特定する Feature Location (FL) という考え[1]があり、今までに多くの手法やツールが提案されている[2]．一方で，システムインテグレータによる開発現場では提案されてきた手法やツールが活用されているとは言えない状況である．

本稿では，まず著者らが携わった開発現場における FL の現状と，提案されている既存の FL ツールを開発現場に導入する際の障壁について述べる．次に，開発現場で活用できる FL ツールの方向性を述べる．

2. 開発現場における FL の現状と障壁

開発現場における FL の現状と障壁の一例として，著者らが携わった開発現場における FL の現状と，FL ツールの適用の過程を述べる．

2.1. 開発現場における FL の現状

この開発現場では，システムの変更要求から修正箇所を特定する FL を以下の 3 手順で実施していた．

1. キーワード選定:
   有識者が，顧客から受領した変更要求を入力として，変更要求に関連する変数名やコード値をキーワードとしてリストアップする．
2. キーワード検索:
   キーワード毎に割り振られた作業者が，Unix のコマンド grep を用いて全ソースを対象にキーワード検索を行う．
3. 検索結果ふるい分け:
   作業者が，検索結果を修正対象に必要か不要か(例:コメント，定義部での出現など)を判断し、修正対象箇所と不要箇所にふるい分ける．

この開発現場では，手順 2.のキーワード検索の結果が 10 万箇所程度出力されるため，手順 3.の検索結果ふるい分けに 3 人月程度の工数がかかっていた．

2.2. 開発現場への FL 技術導入の障壁

この開発現場に対して，grep の代わりに著者らが開発した FL ツール [1)]を導入し，ツールの評価を行った．

評価方法は，この開発現場で過去に実施された FL の結果を正解とした適合率と再現率，および工数削減効果である．評価結果を表 1 に示す．

表 1 提案手法の評価結果

	既存手法	提案手法
適合率 [2)]	0.7%	100.0%
再現率 [3)]	100.0%	81.9%
実施工数	60 人日	1 人日

†1 株式会社 NTT データ
NTT DATA Corporation
†2 大阪大学
Osaka University

1) Eclipse の Java 検索に類似したプログラムの意味を考慮した検索を行えるツール．現在社内で開発中のツールのため詳細は割愛する．

本ツールにより，FL の実施工数は従来比の 1/60 程度まで削減できることが分かった．また，grep の検索結果にはないが実際には必要な修正対象箇所を指摘できた．一方で，従来手法では検索できたが，本ツールでは検索できない修正対象箇所も存在した．

上記の結果を元に開発現場の PM1 名，リーダ 1 名，担当者 2 名の計 4 名に「本ツールを導入できるか」というヒアリングを実施した．その結果を，以下に示す．

- 既存手法で検索できた修正対象箇所が検索できなくなり，調査漏れが発生するため，既存手法の代替としては導入できず，工数削減には寄与しない．
- 今まで見つからなかった修正対象箇所を指摘できるため，既存手法との併用をすることで品質向上には寄与する．その場合は 1 人日の追加の工数が発生することを受け入れる必要がある．

この回答を踏まえ，同じ 4 名に以下のようなヒアリングを実施した．

- 改善策：「適合率は 30%程度になるが，再現率が 98%程度に向上し，既存手法では検索出来なかった修正対象箇所を数多く指摘できる」を実施した場合は既存手法の代替として導入できるか

この質問に対する回答は，「再現率が向上しても 100%にならない限り，既存手法の代替として利用することはできない．改善前と同じく品質向上としての目的であれば利用できる」といったものであった．

この回答は一見，非合理的である．なぜならば，合理的に考えれば，「本ツールでのみ指摘できる修正対象箇所と，既存に比べて漏れる修正箇所の多寡によって採用するかどうかを判断する」はずであるが，ヒアリングの結果は，「本ツールでのみ指摘できる修正対象箇所数とは無関係に，既存に比べて漏れがあるかどうかだけで評価している」ためである．

こういった一見非合理的な判断が行われる理由として，損失回避が挙げられる．損失回避とは，「人間は同じ量の利得と損失であれば，得られる利得よりも失う損失の方を大きく評価する」という経済学における仮説の一つである [3].

この損失回避が FL ツールを開発現場に導入する際の障壁となっていると考えられる．今回の現場の判断もこの損失回避が働いたものと考えられるし，この損失回避は人間の一般的な性質であるため，同様のことが他の開発現場における FL ツールの導入時にも起こっていると考えられる．

このことから，FL ツールを開発現場に導入するためには，「損失回避されない」技術や評価方法を検討する必要があると考える．

3. 現場で活用できる FL の方向性

筆者が考えている現場で活用できる FL ツールの方向性をいくつか挙げる：

- キーワード選定のサポート： 変更要求を入力として，有識者が思いつかない変数名やコード値を挙げる．
- 検索結果のサポート： 影響が波及する箇所も検索結果として出力する．
- 結果絞り込みのサポート： 結果の絞り込みをインタラクティブに実施する．

4. おわりに

本稿では，FL を現場で活用するかどうかの判断に大きく影響する要因として，「既存手法で検索できる修正対象箇所は提案手法でも漏らさないこと」を挙げ，その理由を損失回避の観点から論じた．また，損失回避を踏まえた現場で活用できる FL ツールの方向性を挙げた．

ワークショップでは，その他に考えられる FL ツールの方向性や，損失回避を踏まえた FL ツールの評価方法について議論したい．

参考文献

[1] V. Rajlich and N. Wilde: The Role of Concepts in Program Comprehension, IEEE International Workshop on Program Comprehension, 2002

[2] B. Dit, M. Revelle, M. Gethers and D. Poshyvanyk :Feature Location in Source Code: A Taxonomy and Survey, Software Maintenance and Evolution: Research and Practice, 2011

[3] A. Tversky and D. Kahneman: Loss Aversion in Riskless Choice: A Reference-Dependent Model, The Quarterly Journal of Economics, Vol. 106, No 4, pp. 1039-1061, 1991

2)提案手法の適合率が 100%となっているが，これは過去の実施結果で適合率が 100%となるようにツールを作成したためである．別の案件では 100%を下回ると考えられる．

3)既存手法の再現率が 100%となっているが，過去の実施結果を正解としているためである．
「真の正解」に対しては過去の実施結果自体にも漏れがあるため，「真の正解」に対して 100%であるとはいえない．