ソフトウェア工学 131－12
（2001．6．1）

ウィンターワークショップ・イン・金沢 報告
－計測と解析－

松本 健一

奈良先端科学技術大学院大学
情報科学研究科
〒630-0101 奈良県生駒市高山町 8916-5
matumoto@is.aist-nara.ac.jp

鯵坂 恒夫

和歌山大学
システム工学部
〒640-8510 和歌山市栄谷 930
ajisaka@sys.wakayama-u.ac.jp

あらまし 本稿では，2001 年 1 月に行われた「ウィンターワークショップ・イン・金沢」での「計測と解析」グループにおける議論の結果に基づき，ソフトウェアを対象とした計測と解析の研究・技術課題について述べる．特に，「オープンソース」を対象とした計測と解析では，ソースコードの公開が計測・解析技術の研究開発をより効率よく，系統的なものとすること，また，計測や解析の結果をソースコードに付加して公開することで，ソフトウェアの開発や利用に関する知識の共有が可能となるかもしれないことを述べる．また，「レガシーソフトウェア」を対象とした計測と解析では，「ソースコードの理解」の支援が一つの目的であり，保守作業者の知識を再利用可能とすることも重要であることを述べる．

Report on the discussion in IPSJ Winter Workshop in Kanazawa
- Software metrics -

Kenichi Matsumoto

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0101
matumoto@is.aist-nara.ac.jp

Tuneo Ajisaka

Faculty of Systems Engineering
Wakayama University
930 Sakaedani, Wakayama 640-8510
ajisaka@sys.wakayama-u.ac.jp

Abstract This paper describes the current two challenges on software metrics, based on the summary of discussion in "IPSJ Winter Workshop 2001 in Kanazawa." (1) The concept of "open source" makes it more efficient and systematic to develop software metrics. We claim that the share of the knowledge about development and use of software may be attained by adding and opening the result of measurement and analysis to a source code. (2) One of the main objectives to measure "legacy software" is to help maintainers to understand the source code to be revised. If we develop a model of software maintenance process, based on the result of measurement and analysis to the legacy software, we can share the knowledge about software maintenance.

1. はじめに

2001年1月に行われた「ウィンターワークショップ・イン・金沢」では，「計測と解析」グループが設けられ，ソフトウェアの精密かつ科学的な解析に向けた，メトリクスの抜本的な改善（何をどう測るか），についての議論を行った．本稿では，「計測と解析」グループにおける議論の結果に基づき，ソフトウェアを対象とした計測と解析（ソフトウェアメトリクス）の研究・技術課題について述べる．なお，グループでの議論の対象や範囲を表すものとして，グループメンバーがそれぞれのポジションペーパーで取り上げたトピックスを図1に示す．図1では，3つの軸：(1)フェーズ（保守，開発，利用），(2)計測対象（プロダクト，プロセス），(3)計測による改善対象（プロダクト，プロセス）を設け，トピックスの位置付けを大まかに表している[1][4][7].

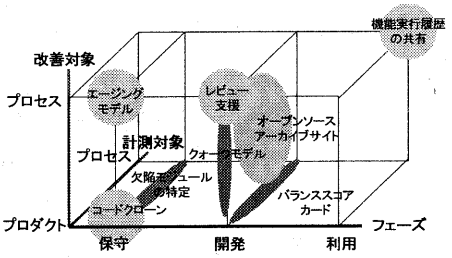

図1 ポジションペーパーのトピックス

2. 研究・技術課題

(1)オープンソース

オープンソースの考えに基づき公開されたソースコードをうまく利用すれば，ソフトウェアメトリクスの研究開発をより効率よく，系統的に行うことが出来る．例えば，公開されているソースコードの中には，実用規模で，かつ，多くのユーザによる利用実績のあるものがある．そうしたソースコードにメトリクスを適用して得られた結果は，いわゆるトイ・プログラムに適用して得られた結果と比べ，信頼性が高く，メトリクスの適用可能範囲も明確になる．

但し，ソースコード以外の情報，例えば，公開前の開発プロセスや公開後の改版プロセスに関する情報，をどのように収集し，表現するのかは今後の課題である．より多様な計測や解析を実現するためには，公開されていないソースコードに対する計測や解析の支援も引き続き検討していく必要がある[10].

また，計測や解析の結果をソースコードに付加して公開する技術の研究開発も重要である．公開されているソースコードであれば，その計測や解析の結果も公開可能である場合が多い．公開によってソフトウェアメトリクスのオープン化が進むと共に，オープンソースの利用者にとっては，ソースコードの特性に関するより多面的な情報が提供されることとなる．ソースコードそのものだけでなく，その開発や利用に関する知識の公開や共有へと発展する可能性もある[6][8].

(2)レガシーソフトウェア

稼動中の大規模ソフトウェアを全く新しいソフトウェアで置き換えるコストとリスクは依然として大きい．分野によるばらつきはあるが，遺産的なソフトウェア（レガシーソフトウェア）はまだしばらくは存在しつづけると考えられる[9].

レガシーソフトウェアの計測や解析における課題の一つは，保守作業（不具合の修正や機能の追加・変更）において不可欠な「ソースコードの理解」をいかに支援するかである．レガシーソフトウェアの多くは大規模である．しかも，度重なる改版でソースコードの構造は必要以上に複雑となっている．設計ドキュメントがもはや存在しない，あるいは，存在しても記述内容が現在のソースコードと一致しない，という場合もある．開発者自身が保守作業を行う場合であっても，ソースコードを正しく理解するためにはある程度の工数が必要となる．

そこで，例えば，ソースコードのみを計測や解析の対象とし，しかも，数百万行のソースコードを効率よく処理し，全体と部分を対比させながら，ソースコード理解に有益な情報を提供することの出来るツールの開発が望まれる[2]．また，計測や解析の結果に基づくリファクタリング（定量的リファ

クタリング）により，ソースコードの理解と共に構造の簡単化を支援するアプローチも考えられる．

レガシーソフトウェアの計測や解析では，保守作業者の知識を再利用可能とすることも重要である[5]．保守作業者の知識の多くは，いわゆる暗黙知である．例えば，レガシーソフトウェア中のあるモジュールについて，再構築（リエンジニアリング）すべきであるか，今後も継続して使用し機能追加していくべきであるか，その判断基準を明示できる保守作業者は多くない．計測や解析に基づいて保守プロセスをモデル化し，保守作業者に示すことができれば，暗黙知を形式知とし保守作業者間で共有することも可能となる．

3. おわりに

プロジェクト管理者による計測や解析は，開発者との間で「計測（解析）する側とされる側」という構図を生みやすい．この問題を避けるため，Humphrey が提案する Personal Software Process (PSP) では，開発者が自らのスキルを知り，能力を高めるために計測や解析を行う[2]．但し，計測や解析の結果を客観的に解釈するためには，開発者間での計測（解析）結果の共有が必要となる．ソフトウェアを対象とした計測と解析（ソフトウェアメトリクス）の今後の研究・技術課題の一つは，この「共有」にあると言える．

参考文献

[1] 鯵坂恒夫："ソフトウェアクォークモデルに基づくミクロソフトウェア工学の方法"，ソフトウェア工学の基礎 VI（日本ソフトウェア科学会 FOSE"99），レクチャーノート/ ソフトウェア学 22, pp.108-115, 近代科学社 (1999)．

[2] W. Humphrey, " Using a defined and measured personal software process, *IEEE Software*, Vol.13, No.3, pp.77-88 (1996).

[3] 神谷，楠本，井上："コードクローン検出における新手法の提案及び評価実験"，電子情報通信学会技術研究報告，ソフトウェアサイエンス研究会，Vol.100，No.570，pp.41-48 (2001-1)．

[4] R. S. Kaplan and D. Norton: *Balanced Scorecard: Translating Strategy into Action*, Harvard Business School Press (1996). 吉川武男（訳）：バランススコアカード－新しい経営指標による企業変革，生産性出版 (1997)．

[5] A. Monden, S. Sato, K. Matsumoto and K. Inoue, "Modeling and analysis of software aging process," *Proc. of International Conference on Product Focused Software Process Improvement* (2000), F. Bomarius and M. Oivo (Eds), *Lecture Notes in Computer Science*, Vol.1840, pp.140-153, Springer-Verlag (June, 2000).

[6] 森崎，門田，松本，井上，鳥居："機能実行履歴を用いたソフトウェア利用知識の共有"，情報処理学会論文誌，Vol.41，No.10，pp.2770-2781 (2000-10)．

[7] 中島，直田，堀田："C++ を対象とした再利用に関するメトリクスの測定"，オブジェクト指向最前線'98, pp.45-48 (1998)．

[8] 大橋，山本，阿草："ハイパーテキストに基づいたソースプログラム・レビュー支援ツール"，電子情報通信学会技術研究報告，ソフトウェアサイエンス研究会，Vol.98，No.28，pp.15-22.

[9] S. Sato, T. Hayama, A. Monden and K. Matsumoto, "Classifying defect-prone modules based on intra-module complexity metrics from a large scale system," *Proc. of 2nd World Congress for Software Quality*, pp.607-612, Yokohama, Japan (Sept., 2000).

[10] 吉田敦："ソースプログラム変更ツールのための内部表現形式の提案"，日本ソフトウェア科学会第 17 回全国大会講演論文集，D7-3，(2000-9)．