

PrototypeTracer: 使い捨て型プロトタイピング時の自己省察支援システム

北原壮[†] 中西泰人^{†,††}

PrototypeTracer: A System To Support Reflective Thinking on Throwaway Prototyping

SOH KITAHARA[†] and YASUTO NAKANISHI[†]

1. はじめに

近年のソフトウェア開発においてウォーターフォールモデルの問題点の解決策として実際に制作を始める前段階に簡易的なモデルやプログラム等を作成し、利用者、要求主からの評価を反映し機能改善を繰り返すプロトタイピングという手法がある。代表的な手法として、短期間でより簡易的なプロトタイプを作り評価し捨てる事を繰り返す使い捨て型プロトタイピングと、一つのプロトタイプを発展させ最終的な成果とする進化型プロトタイピングがある。一般的に、前者の手法の後に後者の手法を用いることが多い¹⁾

使い捨て型プロトタイピングの問題点として、一つのプロトタイプに固執してしまい、捨てずに機能を追加をし続け、使い捨て型として適切な評価を行っていない場合があるという点がある。これは、作成者が評価する捨てるという行動に対して、正確に認識が行えていない事が原因だと考えられる。

また一方で三輪らは創造活動において自己省察を行う事でメタ認知的な活動への重要性を理解、認識できるとしており²⁾、使い捨て型プロトタイピングにおいても自己省察を行う事は有効であると考えられる。また、中小路は情報創出の初期段階は創出される情報を常に最終形態として見ておらず、思考の外在化を行うことでどうつくるか、何を作るべきかの理解が進むと述べており、プロトタイプ作成時の思考を外在化する事は自己省察に有効であると考えられる³⁾。しかし自

己省察は繰り返し行われる必要があり²⁾、手間がかかるため、この手間を軽減する事で自己省察を繰り返し行いやすくする事が出来るだろう。そこで本研究では、プロトタイプ作成中の思考の外在化の手間を軽減する事で使い捨て型プロトタイピング時の自己省察を支援する PrototypeTracer を提案する。

2. PrototypeTracer

2.1 PrototypeTracerLog

プロトタイプ作成時の思考で外在化が行われるべきものは、こうしてみた、こうだどうだろうかという試行錯誤の情報等が考えられる。こうした意図の外在化が、次のアイデアの発展や後の気づきにつながる。

思考の外在化には利用者の負担が少ない必要があるが³⁾、本研究ではソースコードのコメントを利用することにした。コメントを利用する事で利用者はプログラム制作に用いるエディタ以外のソフトウェアを使う必要がない。

一般的なコメントと区別するために特定の形式(//@コメント@//)として、意図や、気づいた事、思考の揺れなどを書き込む。そうして記述されたソースコードを実行し、実行を終了すると、そのスクリーンショットをコメントを保存できるよう、Processing のクラスを拡張した。UNIXTIME をディレクトリ名としたディレクトリをソースコードと同階層に作成し、スクリーンショットをコメントが実行の度に自動的に保存される。ソースコードと同階層に保存の為のディレクトリを作成し、このディレクトリはそこへ集積される(図1)。

2.2 PrototypeTracerMap

PrototypeTracerLog によって保存された情報を表示する PrototypeTracerMap を、Adobe Integrated

[†] 慶応義塾大学環境情報学部

Faculty of Environment and Information Studies, Keio University

^{††} 科学技術振興機構 さきがけ

JST PRESTO

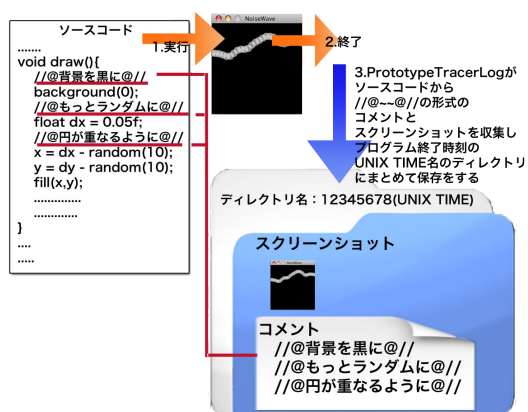


図 1 PrototypeTracerLog
Fig. 1 PrototypeTracerLog

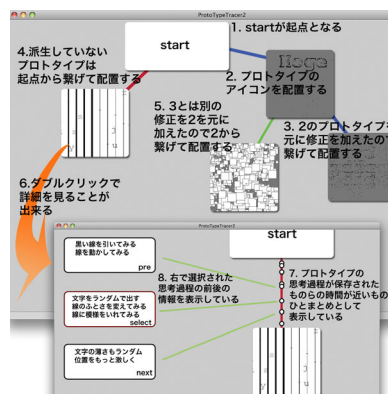


図 2 PrototypeTracerMap
Fig. 2 PrototypeTracerMap

Runtime(AIR) によって作成した．起動時にプロトタイプのプログラムが実行されるディレクトリを指定することで，そのディレクトリを探索し PrototypeTracerLog が保存した情報の取得を行う．収集後，プロトタイプが完成した際のスクリーンショットがアイコンとして PrototypeTracerMap に表示される．利用者は PrototypeTracerLog の情報を元にそのプロトタイプアイコンを選択し，かつてのプロトタイプから派生した物か，新しいプロトタイプかを判断し，PrototypeTracerMap 内でアイコンの配置を手動で行ってゆく．派生元となったプロトタイプアイコンをクリックで選択し，マウスをドラッグすると，派生関係を表示する線によってアイコンが結ばれる．これを繰り返す事で全てのプロトタイプ同士がどう派生しているかを確認する事が出来る．またプロトタイプアイコンをダブルクリックする事で自分が書いたコメントを見る事が出来る．プロトタイプが実行され終了する度に PrototypeTracerLog がコメントとスクリーンショットの保存を行うため，一つのプロトタイプが完成するまでにどの程度時間がかかった，どのような思考過程を経たかを見る事が出来る (図 2) ．

2.3 自己省察の支援

使い捨て型プロトタイプは制作，評価，捨てるというサイクルを繰り返すが，一つのプロトタイプを完成させ評価を行う際に，自己省察を行う．行われるべき自己省察は，どのようなプロトタイプを作ったかのような完成後への評価と，どのようなアイデアを思いついた，どのように作ったか等の思考過程の評価などがある．プロトタイプ同士の派生関係，プロトタイプが作成されるまでの思考過程を可視化する事で利用者はどう作っているか，どう考えているかを深く理解が出来るだろう．このような自己省察を踏まえた上でプロトタイプへの評価を PrototypeTracerMap へ書き

込む．この書き込みも他のプロトタイプの評価を行うときに役に立つだろう．

新しくプロトタイプを作る際に以前のプロトタイプを修正，組み合わせる場合があるが，PrototypeTracerMap からソースコードのコピーが行えるようにした．これにより再利用性が高まりプロトタイピングのサイクルをより繰り返しやすいくなる．また，制作されているプロトタイプの種類が少ない，派生も余り行われていない場合は思考が詰まってしまっているという気づきを与え，新しいプロトタイプの生成をユーザに促すことにつながると考える

3. 終わりに

本稿ではソフトウェアにおける使い捨て型プロトタイプ時の思考の外在化の手間を軽減する事で自己省察を支援するシステムの提案を行った．プロトタイプ同士の派生関係を明らかにし，プロトタイプ時の思考の情報をコメントという形で利用者の負担を軽減し外在化することで，自己省察を行う事が有効であると考えている．しかし，複数人でのプログラミングやプログラミング以外の環境での実行は行っていない．今後はこれらの調査，検討が必要がある．

参考文献

- 1) Fabrice Kordon, Luqi: An Introduction to Rapid System Prototyping, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol.28, No. 9(2002)
- 2) 三輪和久・石井成郎: プロセスの自己省察を軸とした創造性教育, 電子情報通信学会技術研究報告, Vol. 102, No. 509, pp. 37-42(2002)
- 3) 中小路久美代・山本恭裕: 創造的情報創出のためのナレッジインタラクシオンデザイン, Vol. 19, No. 2, pp. 235-246(2004)