

付箋機能を用いた情報共有によるソフトウェア開発支援システム

門田 樹[†]柴田 佳菜美[†]大西 雅宏[‡]高田 秀志[†]立命館大学情報理工学部[†] 立命館大学情報理工学部[†] 立命館大学大学院理工学研究科[‡] 立命館大学情報理工学部[†]

1. はじめに

ソフトウェア開発は複数の開発者で行われることが多いが、開発作業を効率よく行うためには、開発者が成果物を生み出すための情報を共有することが重要である。ソフトウェア開発を行うとき、仕様書やソースコードには書かれていない背景の知識を利用することがある。たとえば、ソースコードを作成するときに参考にしたプログラミングマニュアルの内容は、後で他の開発者がこのソースコードを見たときに処理の内容を理解するのに有用である。しかし、実際に仕様書やソースコードを作成する作業は、個人での活動になりがちであるため、開発者間でこのような情報の共有が十分に行われていない。その結果、勘違いや思い込みが起り、システム開発における非効率性に繋がっている。

ソフトウェア開発を対象とした情報共有の手法としては、メーリングリストに投稿される情報を収集し、階層的に表現された知識として利用者に提示するものがある [1]。しかし、この手法では、どの情報が自分にとって有用であるのかわからないため、開発者が共有している情報を提示する新しい仕組みが必要である。

本稿では、開発者が開発時の背景知識を記録し、それを他の開発者に提供するための仕組みとして、付箋機能を用いた情報共有手法を提案する。本手法では、開発者と成果物、成果物の一部である構成要素などの関係が開発者知識ネットワーク [2] により管理される。また、開発者間の関係を用いて、ソースコードなどに貼られた付箋を関連付け、各開発者にとって有用な情報を判断し、開発者に情報を提供する。

このような付箋機能を実装した環境で開発者に情報を提供することで、開発者は有益な情報を共有することが可能となり、ソフトウェア開発の非効率性を減少させることができる。

2. 付箋の利用

我々が普段使っている紙の付箋は、メモ書きを一時的に文書や書籍などに貼り付けることを目的として作られたものであり、気軽にメモを残すことに適している。仕様書にも付箋は付けられており、仕様書のあいまいな部分を補うために貼られたり、どのようにソースコードが作成されたか等の情報が書かれたりする。そこで、本手法では仕様書やソースコードにメモ書きを残すことに着目し、知識を共有するための仕組みとして、付箋機能を用いる。

付箋機能によりソフトウェアの開発を支援するものとしては、ソフトウェアブラウザというものがある [3]。この研究では、付箋を使ったコメントをソースプログラム上に貼り、開発者間で共有することで、コミュニケーションを支援し、効果的にプログラム開発を進められるようにしている。しかし、付箋は一時的なものとして扱われており、時間の経過により削除されている。そのため、短い期間しか情報を共有することができず、共有された

情報が有益であるのかという判断ができない。これに対して、本研究では、付箋に書かれている情報を削除せずに保持しておき、ユーザの参照履歴によって付箋の有益さを判定する仕組みを導入する。これにより、付箋の有益さに基づいて適切な情報を開発者に提供することが可能になる。

3. 付箋の推薦

付箋には作成者、作成日時、コンテンツ、キーワード、リンク、評価値の要素を持たせる。付箋が作成されたときに、作成者と作成日時は自動的に決定される。コンテンツ、キーワードは作成者によって入力される。コンテンツには、実際に付箋に表示される内容を書く。リンクには、その付箋に関連するウェブページや、関連のファイルなどを登録することができる。評価値には、キーワードを元に計算された数値が初期値として登録される。

新たに登録された付箋は、他の付箋との関連度を表す評価値がキーワードの一致度に基づいて計算され、ある閾値を越えるものを付箋間のリンクとして扱う。登録された付箋には、図 1 に示すように、他の付箋との関連性に基づいてリンクが貼られる。開発者が成果物に貼られた付箋を見たとき、その付箋に関連するほかの付箋が推薦される。どの付箋が推薦されるかは、付箋間のリンクの評価値に、付箋を貼った開発者との関連性を加味して決定される。

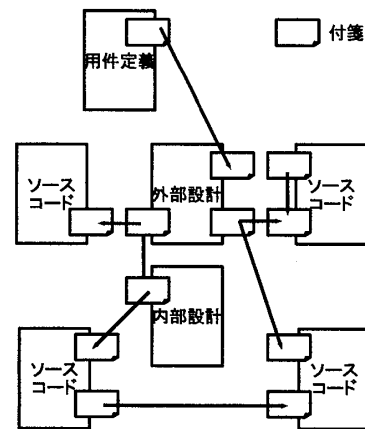


図 1: 付箋間のリンク

3.1 付箋間の評価値

付箋間の関連は、各付箋が他の付箋に対して持つ関連の強さであり、評価値として表わされる。図 2 に、付箋間の関連と、その評価値の例を示す。付箋 A, B, C, D を開発者 a が作成し、付箋 E は開発者 c が作成しているとする。このとき、 $E_{D \rightarrow A}$ は、付箋 D から見た付箋 A に対する評価値を意味し、図の例では 0.5 としている。付箋間の関連の評価値の初期値はキーワードの共起率によって計算される。例えば付箋 A から見た付箋 B への評価値は、

$$\frac{\text{付箋 A のキーワードのうち、付箋 B のキーワードに含まれている数}}{\text{付箋 A のキーワード数}}$$

A Software Development Support System by Information Sharing Using Annotations

[†] Tatsuru Kadota · Ritsumeikan University

[†] Kanami Shibata · Ritsumeikan University

[‡] Masahiro Ohnishi · Ritsumeikan University

[†] Hideyuki Takada · Ritsumeikan University

によって求められる。さらに、この評価値は、実際に開発者によって付箋が見られたかどうかによって、値が更新される。例えば、付箋Aに対して付箋BとCが推薦され、開発者が、付箋Bのみを見たすると、付箋Bの評価値が上昇し、付箋Cの評価値は減少する。

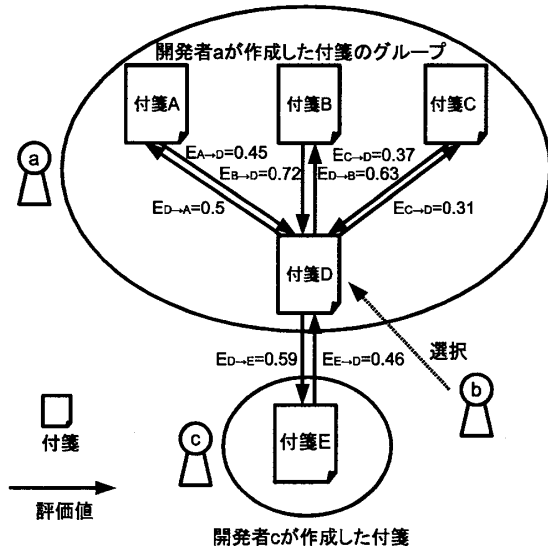


図2: 付箋間の評価値の図

3.2 開発者間の関連度

開発者間の関連度は、開発者間の関連の強さで表わされる。開発者bから見た開発者aへの関連度は、

$$\frac{\text{開発者 a から作成した付箋のうち、開発者 b が見た付箋の数}}{\text{開発者 a が作成した付箋の数}}$$

で求められる。例えば、開発者aが作成した50個の付箋のうち、開発者bが30個見ていれば、この関連度は0.6となる。

3.3 付箋推薦の重要度

付箋は、付箋間の関連の評価値に、開発者間の関連度を加味した重要度に基づいて、推薦される。付箋Aから付箋Bへの重要度は、『付箋Aから見た付箋Bへの評価値 + 付箋Aを作成した開発者から見た付箋Bを作成した開発者への関連度 $\times \alpha$ 』により求められる。ここで、 α はユーザーが任意に決定することができ、 α が0に近いと付箋間の評価値を重視し、 α が大きいと開発者間の関連度を重視する。この重要度に基づいて、付箋が推薦される。さらに、開発者は推薦される付箋の重要度を制限することができる。

推薦される付箋がどのように決定されるかを図2の例を用いて説明する。今、開発者bが付箋Dを選択したとする。このとき、付箋Dからリンクが貼られている付箋A,B,C,Eに対する評価値は $E_{D \rightarrow A} = 0.5$, $E_{D \rightarrow B} = 0.63$, $E_{D \rightarrow C} = 0.37$, $E_{D \rightarrow E} = 0.59$ となっているとする。これを数値が高い順番に並べると、 $E_{D \rightarrow B}$, $E_{D \rightarrow E}$, $E_{D \rightarrow A}$, $E_{D \rightarrow C}$ となる。この数値に、付箋A,B,Cを作成した開発者aに対する関連度と付箋Eを作成した開発者cに対する関連度を前項の手法で求め、付箋の評価値に加える。開発者bから見た開発者aの関連度と開発者cの関連度が同じ場合は、推薦される付箋の順序は、B,E,A,Cである。しかし、開発者bは開発者aとの関連が弱く、開発者cとの関連が強かった場合、付箋Eが1番最初に

推薦され、E,B,A,Cの順番に提供されることになる。たとえば、開発者aとの関連が0で、cとの関連が0.6の場合、開発者bが α を0.1に設定していると仮定すると、付箋Eの重要度 = $0.6 + 0.6 \times 0.1 = 0.66$ となり、付箋Eの重要度が最上位に推薦される。

4. システムの実装

本稿で提案している付箋機能は、統合開発環境の機能として開発者に提供される。開発者は、統合開発環境上でソフトウェア開発を行い、成果物に付箋を貼っていく。貼られた付箋は、開発者知識ネットワークに蓄積されていく。この蓄積された情報を用いて、付箋の重要度が計算され、開発者に提供される。

開発者は、自分へのメモ、ノウハウ等を付箋として作成し、成果物に貼り付けていく。貼りつけた付箋は、開発者知識ネットワークに蓄積されていく。付箋を作成する時に、開発者はキーワードを登録する。このキーワードに基づいて、既に存在している他の付箋に対する評価値が求められる。

開発者がある付箋を指定したとき、指定された付箋から見て重要度の高い他の付箋を推薦する。ある付箋がクリックされた時に、その付箋からリンクされている他の付箋への評価値に開発者間の関連度に加えられ、数値の高いものから表示されていく。開発者がシステムに推薦された付箋を閲覧すると、閲覧された付箋の評価値が上昇する。また、システムに推薦されても開発者に選択されない場合、開発者が設定した期間ごとに、評価値が減少する。さらに、付箋を辿っていくことで、同じ付箋が何度も表示される可能性があるために、同じ付箋は1度しか表示されないようにする。

開発者が有用と判断した付箋は、クリップボードに貼り付けておくことでいつでも見ることができる。これにより、目的の付箋を見るために、わざわざ付箋が貼ってある成果物を開いてから見るという作業をする必要がなくなる。

5. おわりに

本稿では、ソフトウェア開発における付箋機能を用いた情報共有手法を提案した。本手法は、成果物に付けられている付箋を共有し、開発者が必要とする情報を提供することを可能にする。

今後は、本稿で提案したシステムが有効であることを検証するために、本システムを用いて実際にアプリケーションの開発を行う予定である。また、この検証を通じて得た意見を参考に、GUIやシステムの改善を行っていく予定である。

参考文献

- [1] 濱崎雅弘, 武田英明, 河野恭之, 木戸出正継. 『知識の階層構造を用いた協調的情報推薦ネットワークの提案』. 第10回マルチエージェントと協調計算ワークショップ, pp.69-70, 2001.
- [2] 柴田佳菜美, 大西雅宏, 高田秀志. 『開発者の状況に基づく情報共有のための開発者知識ネットワークの構築』. 情報処理全国大会 4X-4, 2009.
- [3] 濱崎雅弘, 武田英明, 河野恭之, 木戸出正継. 『知識の階層構造を用いた協調的情報推薦ネットワークの提案』. 第10回マルチエージェントと協調計算ワークショップ, pp.69-70, 2001.