

暗黙的な知識の再利用を目的とした電子的な付箋の分類手法 A Classification Method for Electronic Sticky-Notes to Reuse Tacit Knowledge

三澤 勇太[†]
Yuta Misawa

山中 啓祐[‡]
Keisuke Yamanaka

高田 秀志[†]
Hideyuki Takada

1. はじめに

システム開発においては、ソースコードや要求定義書、また、システム設計書などの様々な成果物が作成される。これらの成果物は、開発から保守に至るまで保管され、組織内あるいはプロジェクト内で共有される。成果物の作成の過程においては、暗黙的な知識や一時的な情報が発生する。暗黙的な知識とは、個人が経験により獲得したノウハウやもの見方などを指し、一時的な情報とは、タスクに関するメモなどを指す。暗黙知や一時的な情報は、成果物の作成を補助したり、意味を補完したりするなどの役割がある。これらは成果物の作成における背景知識と言える。しかし、こうした知識は組織内で共有されにくいという性質がある。その理由として、暗黙的な知識は、個人の頭の中にある考えであるために形式化されることが少なく、また、一時的な情報は個人用途であり、かつ、一過性の情報であるため、蓄積されにくいことが挙げられる。したがって、このような知識を開発者が形式化し、また、共有することを可能にする仕組みが必要である。

開発中に発生した知識を共有するための手法として、我々は、統合開発環境上で電子的な付箋の添付が可能なインタフェースを開発している [1]。この付箋には成果物に関する知識が記述され、その内容は組織内で共有される。しかし、添付された付箋をそのまま共有すると、添付されている付箋の数が増えるほど、開発者が求める知識を抽出することが難しくなる。様々な情報が混在する中で、開発者が求める知識を抽出するには、まず、共有される知識を分類することが有効であると考えられる。

本稿では、付箋がどういった用途で用いられ、また、その知識が誰に向けられているものであるかによって、付箋を分類する手法を提案する。このような付箋の分類を行うことで、開発者が求める知識への到達容易性を向上させることができる。また、我々の開発している「開発者知識ネットワーク」[2]では、成果物や開発者の行動履歴などの情報が統合開発環境を通して収集され、組織化される。開発者知識ネットワークに本手法を適用することで、成果物同士の関連を考慮した知識提供が可能であると期待できる。

2. 知識の蓄積における問題

本節では、関連研究として、暗黙的な知識や一時的な情報を形式化し管理するものについて述べ、その利点と問題点、また、それらをふまえて、知識共有のときに考慮すべき点を挙げる。

2.1 タグ付けによる注釈の整理

近年の統合開発環境では、ブックマークや開発中に生じたタスクに関するメモ、ハイパーリンクなどといった、

ソフトウェア開発者が記述した注釈の管理を支援する仕組みが提供されている。注釈は役割によって2種類に分けることができる。ひとつは、ソースコードの特定箇所についての情報や意味合いなどを表す「Reminding」であり、もうひとつは、どのソースコードの、どの部分に、何があるかを表す「Refinding」である。これらの2つの役割を兼ね備えたものとして「TagSEA」[3]が提案されている。TagSEAでは、注釈に対し、自由なキーワードや注釈の作成者、作成日時、注釈が付けられた場所などの情報を設定できる。これらの情報を元に注釈が整理されるため、開発者は、誰がどのような考えに基づいて開発を行ってきたかを把握することができる。

2.2 知識の共有および分類の必要性

TagSEAは、特殊なコメントとしてソースコードにデータを直接埋め込む方法を採用しており、ソースコードのみを対象としている。そのため、システム開発において作成される要求定義書やシステム設計書など、ソースコード以外の成果物の作成において発生する知識の蓄積は想定されていない。しかし、ソースコード以外の成果物の作成において発生する知識も有用である。また、TagSEAでは自由なキーワード設定が可能のため、事前に語彙を覚える必要が無く、利用するときの敷居は低い。一方、キーワードの付け方は個人の考え方や習慣に依存するため、開発者ごとに差異が生じやすい。したがって、TagSEAは開発者同士で情報を共有することには不向きと言える。

組織内で個々の知識を広く共有し、効率よく活用することは組織の競争力や業績を挙げることに繋がる [4]。知識活用の効率を上げるためには、開発者が求める知識に容易に辿り着けることが重要である。そのため、個人の考え方や習慣に依存しない知識の分類を開発者に提示し、個人ごとの習慣の違いを吸収することが有効であると考えられる。

3. 付箋の分類

3.1 知識の分類手法

TagSEAの研究では、注釈の2つの働き Reminding と Refinding について着目している。これに対し、本稿ではさらに、開発者が書き残す知識が誰に向けられているかについて着目し、新たに2つの性質があると捉える。ひとつは、自分だけでなく他の開発者に向けても情報を発信する性質であり、この性質を「Public」とする。もうひとつは、個人用の覚え書きに特化した情報を発信する性質であり、この性質を「Personal」とする。そこで、本手法では、Reminding と Refinding という軸と、Public と Personal という軸の2つの軸を基に、知識を4つに分類する。それぞれの分類に対して、どのような知識が該当するのかについて、以下で説明する (図1参照)。

Public型のRemindingに当たるのは、開発中に生じ

[†]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

	Reminding	Refinding
Public	疑問・解決策	参考情報
Personal	変更情報	タスク情報

図 1: 2つの軸と分類の対応

た疑問やバグと、それらに対する解決策に関する知識を記述する「疑問・解決策」であり、Public型 Refindingに当たるのは、開発中に参考になった知識自体、あるいは、情報源へのリンクを記述する「参考情報」である。また、Personal型の Remindingに当たるのは、成果物に変更を加えるとき、その変更理由と変更前の内容を記述する「変更情報」であり、Personal型 Refindingに当たるのは、開発中に生じたタスクについての情報を記述する「タスク情報」である。それぞれの分類に対する、注釈の具体例を表1に示す。

表 1: 各分類の具体的な例

分類	具体例
疑問・解決策	(疑問) ここでの処理が異様に遅いが、原因が不明
	(解決策) メソッド A の代わりに B を用いて解決
参考情報	このサーバの仕様はこのサイト (http://.com/) が参考になった
変更情報	バグの原因究明のために、メソッド C をソースコードからコメントアウトします
タスク情報	スキーマの設計が変更になったので、後でこのテーブルの設計を見直す

3.2 目的別に分類可能な付箋インタフェース

本手法に基づいた、知識の分類が可能な付箋インタフェースは、統合開発環境の機能として提供される。開発者は、統合開発環境上で表示されている成果物の任意の場所に電子的な付箋を成果物に添付することができる。開発者は、付箋をどのような目的で成果物に添付するのかを、図1に示した分類から選択し、その後知識を付箋に記述することができる。

また、付箋は成果物とは分離された情報として添付されるため、付箋の内容が成果物に直接埋め込まれず、成果物に影響を及ぼさない。そのため、メモ書きやソースコードにおけるコメントとは違い、付箋に記述されている知識を長期的に蓄積しておくことが可能である。

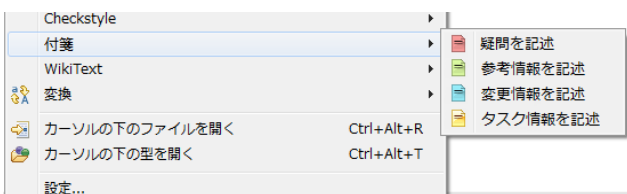


図 2: 付箋の分類を選択するインタフェース

4. 適用例

4.1 開発者知識ネットワークへの適用

開発者知識ネットワークが付箋の収集および組織化を行い、成果物と付箋の関連を抽出することで、関連のある知識を含めた知識提供が可能になる。例えば「変更情報」の付箋が添付される前後に参照あるいは編集されていた成果物には、成果物の変更に至った原因や、変更後に影響を受けた成果物があると考えられる。

4.2 プロジェクトへの適用シーン

本節では、本手法を2つのWebアプリケーション開発プロジェクトに適用したことを想定し、付箋がどのように分類されるかを示す。

例えば、あるプロジェクトでは、Web上のカレンダーサービスのAPIを用いて、組織のメンバーのスケジュール管理ツールを開発したとする。また、開発中に生じた様々な知識は開発者によって付箋に記述され、その内容が蓄積されているとする。蓄積されている知識として、例えば、組織内でのネットワークの設定項目やAPIの利用方法、同じAPIを用いた開発プロジェクトについて述べているWebページへのURLは、「参考情報」の付箋となる。また、APIを利用するときに生じやすい誤解から発生したバグや、その解決策は「疑問・解決策」の付箋となる。一方で、このプロジェクトと同じカレンダーサービスのAPIを用いて、組織で行われるイベントの出席管理ツールを開発している別のプロジェクトがあるとする。

2つのプロジェクトは、共通したAPIを用いているため、成果物の作成過程において、共通する作業があると考えられる。したがって、過去のプロジェクトで蓄積された付箋は、開発中のプロジェクトにおいて、有用な情報源となる。例えば、開発中のプロジェクトにおいて、ネットワーク設定に困ったときは、過去のプロジェクトにおける「参考情報」の付箋を参照することで、目的の知識へ辿り着くことが可能である。

5. おわりに

本稿では、システム開発において発生する知識を蓄積および共有することの重要性について述べ、目的の知識への到達容易性を向上させるための付箋の分類手法を提案した。

今後は、目的別に分類可能な付箋インタフェースをEclipseのプラグインとして実装し、我々の研究室で行われている、実際のソフトウェア開発で使用してもらい、分類の妥当性を評価する予定である。

参考文献

- 門田 樹, 柴田 佳菜美, 大西 雅宏, 高田 秀志. 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. 情報処理学会第71回全国大会 4X-4(2009).
- 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. 情報処理学会第71回全国大会 4X-3(2009).
- Margaret-Anne Storey, Jody Ryall, Janice Singer, Del Myers, Li-Te Cheng, and Michael Muller. How Software Developers Use Tagging to Support Reminding and Refinding. IEEE Transactions on Software Engineering, Vol.35, No.4 (2009).
- 青山 浩二, 鶴飼 孝典, 小幡 明彦, 原田 裕明. 知識共有を動機付する手法. 人工知能学会, 第3回知能流通ネットワーク研究会 SIG-KSN-003-01 (2008).