

Title	OS教材を事例とした, アスペクト指向によるソースコードとドキュメントの関連づけ
Author(s)	大場, 勝
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1783
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 修士

OS 入門用の教材を事例とした, アスペクト指向によるソースコードとドキュメントの関連づけ

大場 勝 (210012)

北陸先端科学技術大学院大学 情報科学研究科

2004 年 2 月 13 日

キーワード: アスペクト指向, 関連, ドキュメント整理, 追跡性.

我々はソフトウェアに関係する多種のドキュメントについて, アスペクト指向を用いて整理する手法を提案する. 一般にソフトウェアには多種多様なドキュメント (例えば, 仕様書やマニュアル, 開発メモ) が混在する. 開発者はここから, ドキュメントを必要に応じて検索し利用している [1]. これは, ソースコードを理解する上でドキュメントが必要不可欠だからである.

しかし, 多種のドキュメントが混在する中から必要なドキュメントを検索することは煩雑である. 従って検索し易くするためにはドキュメントの整理が必要である. 整理とは, 読み手の関心事ごとに, 必要なドキュメント断片の集合を 1 つにまとめる作業をいう. 多種のドキュメントを整理するには以下に挙げる問題点がある.

- ソースコードに読むべきドキュメントの記録がない.
- ドキュメントの分類は, 読み手の知りたい事 (関心事) によって異なるため整理が容易ではない.
- 既存のドキュメンテーションツールはこれらの点に関しては十分でない.

本研究の目的はこれらの問題点をふまえ, ソフトウェアに関係するドキュメントを効率よく検索するためのドキュメント整理手法の提案と, その有効性を実験により評価することが目的である. 本研究の提案するドキュメント整理法は, 関連するソースコードとドキュメントの追跡性を向上させ, 効率のよいプログラム理解を実現する. 追跡性とは, 読み手の関心事に応じて必要なドキュメントを取得できることをいう.

本研究は, ソースコード中にプログラム理解に必要なドキュメントへの参照を関連として形式的に記述し, アスペクト指向を用いてその関連を横断する関心事に基づいて整理する. 関連とは, ソースコードとそのコードの理解に必要なドキュメントへの参照をいう.

アスペクト指向の問題意識は, 1 つの視点でまとめた構造にはそれを横断する関心事が存在するということである. アスペクト指向は, この横断する関心事を 1 つのモジュール

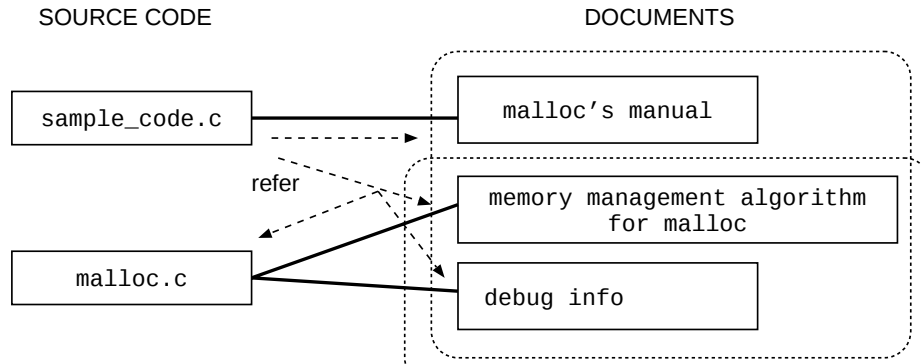


図 1: アスペクトの例

にする技術をいう。本手法を適用することによって、追跡性のあるドキュメントやソースコードを容易に取得することが可能になり、効率のよいプログラム理解が可能になると考えた。

本研究では、関連を記録することによって、ソースコードからドキュメントへの追跡性を向上させる。さらに、開発者が関連を横断する関心事を1つのアスペクトにすることで、関連の追跡性を向上させる。アスペクトは、プログラムの開発者自身が関連を関心事別に整理する。これによって、読み手がソースコード中に埋め込んだ関連以外にも追跡性の高い関連を取得することが可能になる。

図1は、「malloc 関数のサンプルコード」と「malloc 関数の実装」が持つ関連を実線で表し、アスペクトによって分類されたドキュメントを点線四角で囲んだ図である。「malloc 関数のサンプルコード」から直接得られる関連は「malloc 関数のマニュアル」のみである。図1のように、「malloc 関数のサンプルコード」の関連と「malloc 関数の実装」の関連を1つのアスペクトにまとめておけば、「malloc のサンプルコード」からより詳細なドキュメント「メモリ管理」や malloc の実装を取得することが可能になる。さらに「ページングの実装」などの関連をこのアスペクトにまとめておけば、メモリ管理に関するより詳細な情報を追跡することが可能になる。

例えば、既存のドキュメンテーションツール Javadoc[3] では、このような追跡性の高い関連の取得を行うことは難しい。Javadoc はソースコード中に埋め込んだコメントから API リファレンスを生成するツールで、コメント中に URL のリンクを埋め込むこともできる。Javadoc をドキュメント整理ツールとしてみた場合、ソースコード中に直接リンクを埋め込んだドキュメントは取得できるが、アスペクトのように追跡性のある他の関連の取得はできない。また、ソースコードとの対応をとることもできない。

本研究では、アスペクト指向によるドキュメントの整理法を適用したツール ADIOS[2] を実験的に実装した。ADIOS を使用して独自開発中の OS 教材 udos[4] を事例とし、ペー

ジングに関してドキュメントの整理を行った。OS教材を事例としたのは、多くのドキュメントをソースコードと照合しながら理解する必要がある、ドキュメント整理の効果を明確に評価できると考えたからである。

実験の結果、ソースコードから直接埋め込んだ関連以外に、追跡性のある他の関連を取得することができた。この実験に限って、我々の方法が複雑な関係をもつドキュメントをアスペクト整理法で、上手く整理できたことを意味している。また、本研究の問題点として、読み手が関連を選ぶための説明文が難しいことがわかった。我々がドキュメント整理を適用した事例が小規模だったため十分な評価が得られなかった。このため、本手法のより大規模な適用を行い問題点の洗い出しをしなければならない。

参考文献

- [1] Andrew Forward and Timothy C. Lethbridge. The relevance of software documentation, tools and technologies: a survey. In *Proceedings of the 2002 ACM symposium on Document engineering*, pp. 26–33. ACM Press, 2002.
- [2] Masaru OHBA. Home page for Aspect oriented Documents Inspection System(ADIOS).
<http://www.jaist.ac.jp/~m-ohba/adios/>.
- [3] Sun Microsystems. Javadoc for generating API documentation.
<http://java.sun.com/j2se/javadoc/>.
- [4] udos project. Home page for micro-disk-operating-system(udos).
<http://www.jaist.ac.jp/~m-ohba/udos/>.