

発表概要

エンドユーザがプログラム可能な Wiki エンジン

横山 哲郎^{1,a)}

2012年3月15日発表

Wiki は、誰でも自由にウェブブラウザ上でウェブサイトを書き換えることができる手軽さから広く活用されてきている。Wiki 文書は、HTTP サーバ上では Wiki 記法で記述されており、ウェブブラウザなどからの要求があるとハイパーテキストに変換されて、エンドユーザから閲覧される。さらに、エンドユーザは閲覧時にフォームなどから Wiki 記法で Wiki 文書を自由に編集することができる。しかし、Wiki エンジンが稼働するサーバ上で実行可能であるソースコードは、通常、Wiki 文書には記述できない。この理由の1つは、管理者の意図しない悪意のある動作をするソースコードをサーバ上で実行できないようにするというセキュリティ上の制約である。我々は潜在的に危険をとまなう IO アクションを分離して管理することにした。プログラミング言語 Haskell を Wiki エンジンの実装言語として選択して、IO モナド内でエンドユーザの記述した任意のコードを実行させないことで、管理者の意図しない悪意のある動作を防ぐというアプローチをとった。プログラマブルな Wiki ページは、プログラミング言語レベルのモジュールに対応して、互いに参照し合うことができる。技術上の関心から極力 Wiki エンジンの深い部分まで、その Wiki 上でプログラミングができるように設計を試みた。プログラムの入力、通常の Wiki 上のフォームから可能であり、エンドユーザはクライアントに特別な環境を準備しなくても、その Wiki エンジンの開発にリアルタイムに協調的に携わることが可能である。Wiki 文書において記述された副作用のある任意のアクションが実行されないことは Haskell のクラスを用いて静的型検査によって保証される。

Programming Wiki in Wiki

TETSUO YOKOYAMA^{1,a)}

Presented: March 15, 2012

A wiki, a web application for modifying a website by web browsers easily and cooperatively, is now managed by many companies and organizations and many wiki clones are publicly available. Wiki documents are described under Wiki notation. Ordinary wikis are not able to execute part of Wiki documents, which are in part prevented by a potential security risk of the execution of arbitrary malicious codes. We eliminate such a risk by imposing restriction on all the actions under IO monads in end-users' programs. Programmable wiki pages are literate Haskell programs that are language-level modules and can refer to one another. We design the architecture of the proposed wiki engine to be programmable by end users as much as possible. The development of a wiki engine on itself does not require the developing environment in the client computers. Type checking ensures the absence of arbitrary actions with side effects in end-users' programs using Haskell's type class and static type checking.

¹ 南山大学情報理工学部
Faculty of Information Sciences and Technology, Nanzan
University, Seto, Aichi 489-0863, Japan

^{a)} tyokoyama@acm.org