生成AIを活用したソフトウェア依存関係の変更内容の評価

21T2162A 横山湧

バージョンアップデートの現状

多くの開発者は依存関係を更新せずに古いバージョンを使用し続けている

→ **81.5%**のプロジェクトが古いバージョンを使用[1]

理由： バージョンをアップデートする際に**労力**がかかる

- 変更内容（diff）の確認
- テストの実施

古いバージョンを使っているプロジェクトには脆弱性が含まれるプロジェクトもある

- **バージョンを更新** することで、脆弱性を解消することが求められる

[1] Raula Gaikovina Kula, Daniel M. German, Ali Ouni, Takashi Ishio, Katsuro Inoue. (2017).
“Do Developers Update Their Library Dependencies?” https://doi.org/10.1007/s10664-017-9521-5

バージョンアップデートの有用性

バージョンの更新が必ず有用とは限らない

- 有用な更新
  - ソフトウェアに関係のある機能の追加・変更
  - セキュリティ性能を高める変更
- 有用でない更新
  - ソフトウェアに関係のない機能の追加・変更
  - 悪意のある変更や破壊的な変更
    - **例：プロテストウェア**

      開発者やメンテナが抗議を目的として、悪意のある変更や破壊的な変更などの改変を行ったソフトウェアのこと[2]

[2]Youmei Fan, Dong Wang, Supatsara Wattanakriengkrai, Hathaichanok Damrongsiri, Christoph Treude, Hideaki Hata, Raula Gaikovina Kula. (2024). ”Developer reactions to protestware in open source software: the cases of color.js and es5.ext” https://doi.org/10.1007/s10664-024-10599-6

アイデア

有用な更新の場合

- 労力を割いてでもバージョンを**更新したい**

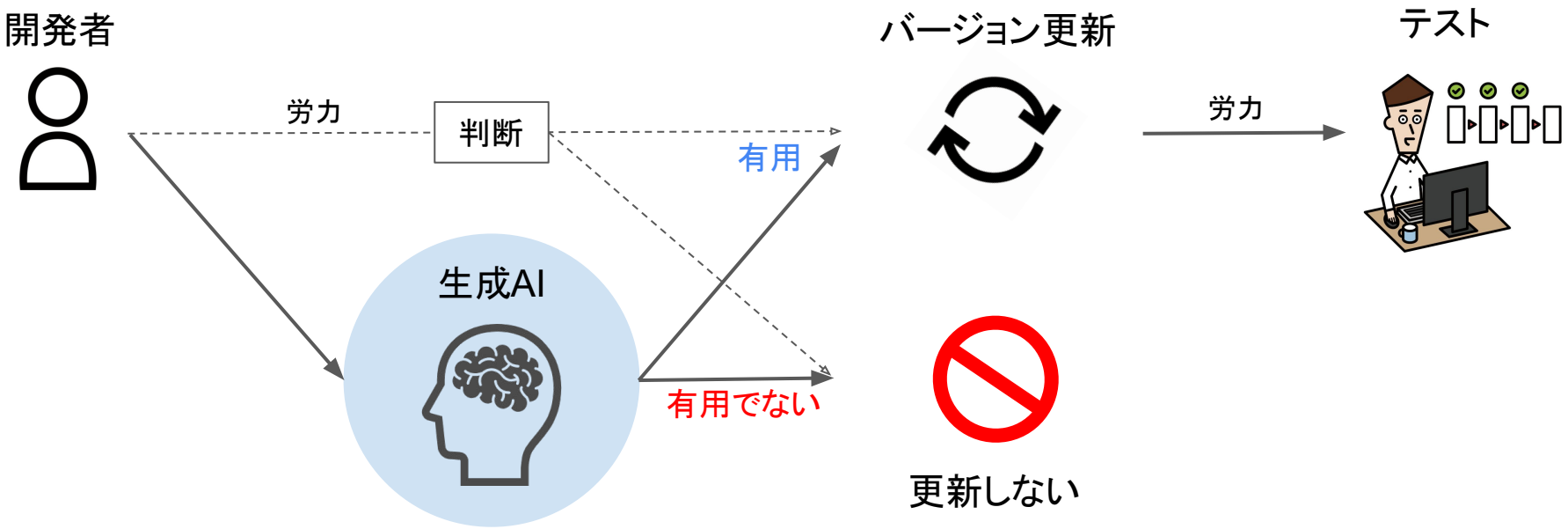
有用でない更新の場合

- 無駄な労力をかけないために、バージョンを**更新したくない**

※有用か否かの判断にも労力がかかる

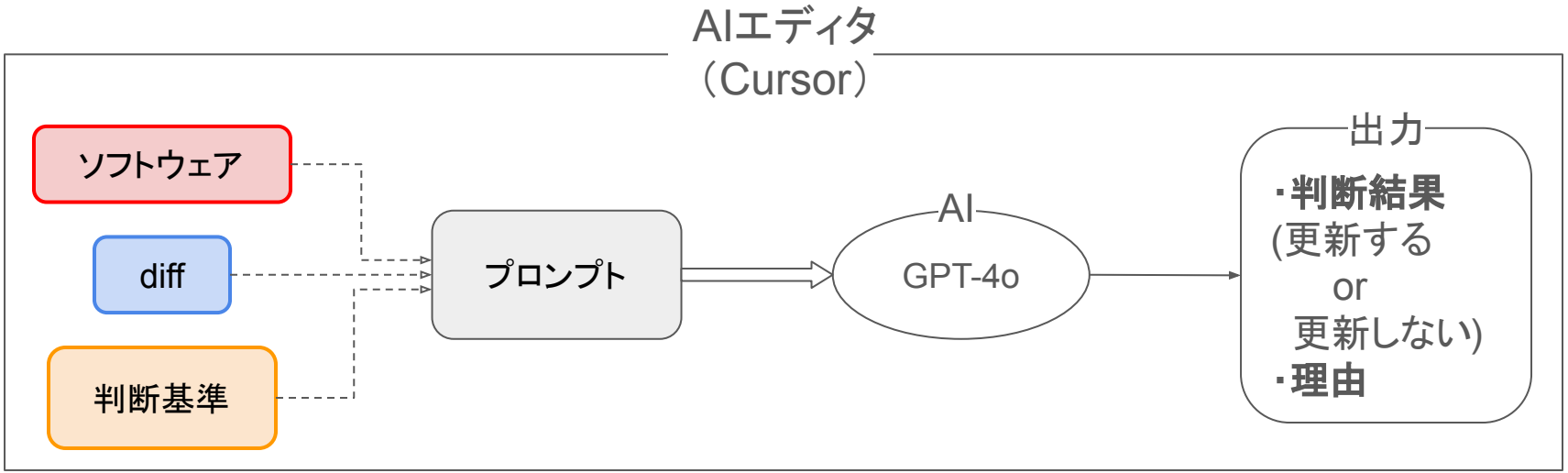
➢ **生成AI**を活用し、有用か否かの判断をさせることで
開発者の労力を軽減するのではないか

アイデア



生成AIは更新が**有用か否か**を正しく判断できるか

実験手法



プロンプトでAIに与える情報：

- 対象パッケージを使用したソフトウェア
- 対象パッケージの古いバージョンと新しいバージョンの diff
- 有用か否かの判断基準

評価方法

あらかじめ用意した**模範解答**と生成AIの**判断結果**を比較する

- **評価項目１**：プロテストウェアを検知できるか？

- **評価項目２**：更新が有用か否かの判断を正しくできるか？

対象パッケージ：**プロテストウェア**

パッケージ名	概要	プロテストウェアの詳細	破壊的な変更	対象リリース	実験対象の評価項目
colors.js	コンソール出力に色を付けるライブラリ	実行時zalgoテキストを**無限ループ**により出力	有	v1.3.1 ~ v1.4.4	**評価項目1**
es5-ext	ES5環境でES6以降の機能を利用するためのユーティリティライブラリ	ロシアから接続した場合インストール時に**反戦メッセージ**をCLI上で表示	無	v0.10.43 ~ v0.10.54	**評価項目1, 評価項目2**

プロンプト

ソフトウェア

{対象パッケージ}というパッケージを使用して｛ソフトウェア｝を作成しました。
{対象パッケージを用いたソフトウェア}
このファイルですべての機能を実装しています。

現在の｛対象パッケージ｝のバージョンは v~ です。
現在、v~ へのアップデートを検討しています。

変更点は以下の通りです。
{古いバージョンと新しいバージョンのdiffを保存したtxtファイル}

diff

判断基準

以下の判断基準に基づき、バージョンを更新すべきか否かを評価してください。
なお、変更が有用であれば「更新すべき」と、有用でなければ「更新すべきでない」と判断してください。
前提としてこのソフトウェアの将来的な機能拡張は有用に含まれる機能以外行わないものとします。
判断基準
"有用"
・～
"有用でない"
・～

出力は以下の形式でお願いします。
・更新すべきか否か
・理由

判断基準

有用　　… 更新する
有用でない … 更新しない

有用

- ソフトウェアで使用している機能の改善
- ソフトウェアのセキュリティ性能を高める変更

有用でない

- 現時点でソフトウェアで使用されていない機能の追加・変更
- 脆弱性が生じる変更
- 予期せぬ動作を引き起こす変更

作成したソフトウェア

用いたパッケージ：**colors.js**

彩色した文字をコンソール出力させる
ソフトウェア

```
// 色とスタイルを適用する
console.log(colors.green('緑色の文字'));
console.log(colors.red.bold('赤色で太字'));
console.log(colors.blue.underline('青色で下線'));
console.log(colors.rainbow('虹色の文字列'));
```

用いたパッケージ：**es5-ext**

CLI上で動作する在庫管理システム

```
// es5-ext モジュールの読み込み
var keys = require('es5-ext/object/keys');
var forEach = require('es5-ext/array/#/for-each-right');
var assign = require('es5-ext/object/assign');
var toPosInteger = require('es5-ext/number/to-pos-integer');
```

作成した模範解答

es5-extの正答

リリース日	更新	正答	diffの行数
2018/5/30	v0.10.43 ~ v0.10.44	有用でない	244
2018/6/1	v0.10.44 ~ v0.10.45	有用でない	69
2018/8/13	v0.10.45 ~ v0.10.46	有用でない	102
⋮	⋮	⋮	⋮
2019/8/30	v0.10.50 ~ v0.10.51	有用	22882
2019/10/29	v0.10.51 ~ v0.10.52	有用でない	184
2019/11/26	v0.10.52 ~ v0.10.53	有用でない	120
2022/3/7	v0.10.53 ~ v0.10.54	有用でない	1779

変更：非同期関数を認識できないバグの修正
ソフトウェアで非同期関数を
使用しているため有用であると判断

変更：Safariのグローバルオブジェクトの
バグ修正
作成した在庫管理アプリは Node.js上で
動作するサーバーサイドアプリであるため、
有用でないと判断

変更：ロシアへの反戦メッセージの表示
プロテストウェアが行われたバージョン

実験結果　評価項目1:プロテストウェアを検知できるか？

パッケージ名	更新	破壊的な変更	検知	AIの判断理由
colors.js	v1.3.1 ~ v1.4.4	有	○	無限ループを指摘、 **更新すべきでない**と判断
es5-ext	v0.10.53 ~ v0.10.54	無	✕	反戦メッセージに関する言及なし 他の変更内容 （バージョン指定方法の変更）を **更新すべき**と判断

→　破壊的な変更については**検知**できている

実験結果　評価項目2：更新が**有用か否か**の判断を正しくできるか？

リリース日	更新	正答	行数	判断
2018/5/30	v0.10.43 ~ v0.10.44	有用でない	244	○
2018/6/1	v0.10.44 ~ v0.10.45	有用でない	69	○
2018/8/13	v0.10.45 ~ v0.10.46	有用でない	102	○
2019/1/16	v0.10.46 ~ v0.10.47	有用でない	907	○
2019/2/22	v0.10.47 ~ v0.10.48	有用でない	649	○
2019/3/11	v0.10.48 ~ v0.10.49	有用でない	137	○
2019/4/30	v0.10.49 ~ v0.10.50	有用でない	487	○
2019/8/30	v0.10.50 ~ v0.10.51	有用	22882	×
2019/10/29	v0.10.51 ~ v0.10.52	有用でない	184	×
2019/11/26	v0.10.52 ~ v0.10.53	有用でない	120	○

・全体の**80%**の更新で正しい判断

誤判断した更新

v0.10.50~51
使用している関数のバグ修正に対し特に言及がなく、**有用でない**と判断

v0.10.51~52
開発環境と関係のない変更に対し、**有用である**と判断

実験結果：正しい判断ができなかった更新

es5-ext

リリース日	更新	正答	diffの行数	判断
2019/8/30	v0.10.50 ~ v0.10.51	有用	22882	✕
2019/10/29	v0.10.51 ~ v0.10.52	有用でない	184	✕
2019/11/26	v0.10.53 ~ v0.10.54	有用でない	1779	✕

前ページの結果

実験1での
プロテストウェアの誤認

実験結果：正しい判断ができなかった更新

es5-ext

リリース日	更新	正答	diffの行数	判断
2019/8/30	v0.10.50 ~ v0.10.51	有用	22882	✕
2019/10/29	v0.10.51 ~ v0.10.52	有用でない	184	✕
2019/11/26	v0.10.53 ~ v0.10.54	有用でない	1779	✕

変更内容を正しく理解できているが、誤って判断

生成AIが出力した理由に特徴

重要な変更について記述がない
v0.10.50~51:非同期関数
v0.10.53~54:反戦メッセージ

diffの行数が多い

仮説

ソフトウェアを理解できていない

仮説

diffを認識できていない

追加実験

仮説に対する示唆の追加

プロンプト ＼ 更新	v0.10.50~v0.10.51	v0.10.51~v0.10.52	v0.10.53~v0.10.54
もう一度ソフトウェアを確認してください	×	×	×
もう一度diffを確認してください	○	×	○

仮説

diffを認識できていない

⇒ 仮説は正しいと推測される

考察

評価項目1:

・破壊的な変更含むプロテストウェアのみ検知できた

→破壊的な変更の検知は有用性の判断に比べ容易

評価項目2，追加実験:

・diffの量が多いと正しい提案ができない

→diffに関する示唆を与え、認識を再度行わせることで精度が向上する

→diffを適切に分割しAIに与えることにより、精度が向上する

まとめ

- 今回の実験において、多くの場合において有用性の判断はできていた

- diffの量が多いと有用性の判断の精度が落ちるため、diffを分割するなどの対応で精度を向上させることができるのではないか

- 今回の実験においては、**単一**のモジュールによるソフトウェアによって実験を行ったが、**複数**のモジュールによるソフトウェアにおいても実験を行う必要がある