

Semantic Patches for Adaptation of JavaScript Programs to Evolving Libraries

Benjamin Barslev Nielsen, Martin Toldam Torp, Anders Møller

Proceedings of the 2021 International Conference on Software Engineering (ICSE)

発表者：辻 裕真

背景

- JavaScriptのライブラリであるnpmは頻繁に更新される.
- 最新のライブラリには新機能やバグ修正が含まれているが、全ての更新に下位互換性があるわけではない.
- 最新のライブラリをしようするためには変更内容を確認し、コードに影響する部分を調べ適応させなければならない.

背景

- この作業は手作業のためエラーが発生しやすく時間もかかる。

```
1 - import { Observable } from 'rxjs/Observable';
2 - import { Subject } from 'rxjs/Subject';
3 - import 'rxjs/add/observable/timer';
4 - import 'rxjs/add/operator/takeUntil';
5 + import { defaultIfEmpty, take, takeUntil, tap }
6 +   from 'rxjs/operators';
7 + import { timer, Subject } from 'rxjs';

8 - const c$ = (new Subject()).take(1);
9 - Observable.timer(warnTimeout)
10 -   .takeUntil(c$.defaultIfEmpty(true))
11 -   .do(() => {...})
12 -   .subscribe();
13 + const c$ = (new Subject()).pipe(take(1));
14 + timer(warnTimeout).pipe(
15 +   takeUntil(c$.pipe(defaultIfEmpty(true))),
16 +   tap(() => {...})
17 + ).subscribe();
```

Fig. 1: Excerpts from *redux-logic* before (marked with red background and '-') and after (green and '+') adapting to the breaking changes in the library *rxjs* 6.0.0.

rxjsライブラリが更新された際に変更されたコードの例

- ライブラリの更新に合わせて、影響を受けるコードを検出して必要な変更を行うJSFIXを提案する。

JSFIX

- JavaScriptプログラムコードをライブラリの重大な更新に適応させる.
- 入力：古いバージョンのライブラリに依存するプログラム
セマンティックパッチ
- セマンティックパッチ
ライブラリの更新により変更が必要になる箇所のパターンと、コードを
適応させる方法を説明するコードテンプレート（パッチ）のこと。
パッチはライブラリの変更ログ等から手動で作成する.

JSFIX

セマンティックパッチ

Semantic Patch Language [\[edit\]](#)

The source code to be matched or replaced is specified using a "semantic patch" syntax based on the [patch](#) syntax.^[4] The **Semantic Patch Language (SmPL)** pattern resembles a [unified diff](#) with C-like declarations.^{[5][6]}

Example [\[edit\]](#)

```
@@
expression lock, flags;
expression urb;
@@

spin_lock_irqsave(lock, flags);
<...
- usb_submit_urb(urb)
+ usb_submit_urb(urb, GFP_ATOMIC)
...>
spin_unlock_irqrestore(lock, flags);

@@
expression urb;
@@

- usb_submit_urb(urb)
+ usb_submit_urb(urb, GFP_KERNEL)
```

JSFIX

- ・ セマンティックパッチとプログラムが与えられると, TAPIR[1]を使ってプログラム内の影響を受ける箇所を特定する.
- ・ 信頼度が低い場合, 質問を行なって誤検知を防ぐ.
(例) 引数xは関数ですか?
- ・ パッチに従ってプログラムを変更する.

A. Møller, B. B. Nielsen, and M. T. Torp, “Detecting locations in JavaScript programs affected by breaking library changes,” Proc. ACM Program. Lang., vol. 4, no. OOPSLA, pp. 187:1–187:25, 2020.

評価

Research Questions

RQ1：よく使用されているnpmパッケージの変更のうち、どのくらいパッチを作成できるか.

RQ2：パッチの適用後新しいバージョンとの互換性があるか.

RQ3：JSFIXはどのくらいの質問をするか.

評価

RQ1：コードテンプレートで表現できた数

- ・ 広く使用されているnpmパッケージのメジャーアップデートでの重大な変更に対するパッチを手動で作成した.

TABLE I: Experimental results for RQ1.

Library	BC	Patterns	Temp	No code template		
				U	NGP	?
<i>lodash</i> 4.0.0	51	123	123	0	0	0
<i>async</i> 3.0.0	4	7	6	1	0	0
<i>express</i> 4.0.0	18	24	23	1	0	0
<i>chalk</i> 2.0.0	3	3	3	0	0	0
<i>bluebird</i> 3.0.0	8	12	7	2	3	0
<i>uuid</i> 3.0.0	1	1	1	0	0	0
<i>rxjs</i> 6.0.0	26	55	53	0	2	0
<i>core-js</i> 3.0.0	26	41	35	0	5	1
<i>node-fetch</i> 2.0.0	9	9	7	0	2	0
<i>winston</i> 3.0.0	23	30	26	1	3	0
<i>redux</i> 4.0.0	2	2	1	0	1	0
<i>mongoose</i> 5.0.0	14	19	13	2	3	1
Total	186	326	298	7	19	2

BC：重大な変更の数

Patterns：影響を受ける箇所

Temp：作成されたパッチ

No code template

U：表現できない

NGP：単一の修正が存在しない

？：変更ログやIssueの情報不足

評価

RQ2：パッチの適用後新しいバージョンとの互換性があるか.

- ・ パッケージの更新前にテストが成功し，更新後に失敗した89のプロジェクトでJSFIXを使用した.

TABLE II: Experimental results for RQ2 and RQ3.

	Client test suite experiment										Pull request experiment									
Library	C	Tr	✓	✗	—CT	Time	OBJ	CALL	EXTRA	MINOR	PR	Acc	Rej	Tr	Time	OBJ	CALL	EXTRA	MINOR	
lodash	13	70	13	0	0	1.76	6	2	3	4	10	1	0	25	2.01	5	10	7	3	
async	8	10	8	0	0	2.29	2	0	0	0	10	2	1	2	0.63	0	0	0	0	
express	10	18	7	0	3	1.19	11	0	0	2	4	1	0	17	5.89	13	0	0	2	
chalk	10	54	10	0	0	0.21	0	0	0	0	10	3	0	0	0.78	0	0	0	0	
bluebird	9	18	9	0	0	0.28	3	0	18	3	10	3	1	2	1.21	0	0	24	1	
uuid	1	2	1	0	0	0.14	0	0	0	0	10	3	0	0	0.95	0	0	0	0	
rxjs	6	200	5	1	0	2.20	40	0	0	0	10	3	0	112	1.18	106	0	0	0	
core-js	8	28	8	0	0	6.25	2	0	0	0	10	4	0	140	17.46	5	0	0	0	
node-fetch	8	7	7	0	1	0.46	7	1	6	17	10	3	1	3	0.24	11	0	20	14	
winston	8	36	7	0	1	0.80	16	14	7	7	10	2	0	19	1.12	34	50	10	4	
redux	4	4	3	1	0	0.33	0	0	0	3	10	2	0	0	3.59	0	0	0	6	
mongoose	4	4	4	0	0	0.29	5	0	5	1	10	0	1	5	1.81	16	6	18	3	
Total	89	451	82	2	5	1.53	92	17	39	37	114	27	4	325	2.92	190	66	79	33	

82のプロジェクトでテストに合格した

27/114のプルリクエストが受け入れられた

評価

RQ3 : JSFIXはどのくらいの質問をするか.

198のプロジェクトで計553の質問をした. (平均 : 2.8)

TABLE II: Experimental results for RQ2 and RQ3.

	Client test suite experiment										Pull request experiment									
Library	C	Tr	✓	✗	—CT	Time	OBJ	CALL	EXTRA	MINOR	PR	Acc	Rej	Tr	Time	OBJ	CALL	EXTRA	MINOR	
<i>lodash</i>	13	70	13	0	0	1.76	6	2	3	4	10	1	0	25	2.01	5	10	7	3	
<i>async</i>	8	10	8	0	0	2.29	2	0	0	0	10	2	1	2	0.63	0	0	0	0	
<i>express</i>	10	18	7	0	3	1.19	11	0	0	2	4	1	0	17	5.89	13	0	0	2	
<i>chalk</i>	10	54	10	0	0	0.21	0	0	0	0	10	3	0	0	0.78	0	0	0	0	
<i>bluebird</i>	9	18	9	0	0	0.28	3	0	18	3	10	3	1	2	1.21	0	0	24	1	
<i>uuid</i>	1	2	1	0	0	0.14	0	0	0	0	10	3	0	0	0.95	0	0	0	0	
<i>rxjs</i>	6	200	5	1	0	2.20	40	0	0	0	10	3	0	112	1.18	106	0	0	0	
<i>core-js</i>	8	28	8	0	0	6.25	2	0	0	0	10	4	0	140	17.46	5	0	0	0	
<i>node-fetch</i>	8	7	7	0	1	0.46	7	1	6	17	10	3	1	3	0.24	11	0	20	14	
<i>winston</i>	8	36	7	0	1	0.80	16	14	7	7	10	2	0	19	1.12	34	50	10	4	
<i>redux</i>	4	4	3	1	0	0.33	0	0	0	3	10	2	0	0	3.59	0	0	0	6	
<i>mongoose</i>	4	4	4	0	0	0.29	5	0	5	1	10	0	1	5	1.81	16	6	18	3	
Total	89	451	82	2	5	1.53	92	17	39	37	114	27	4	325	2.92	190	66	79	33	

まとめ

- JavaScriptプログラムを進化するライブラリに適応させるための作業を自動化するJSFIXを提案した.
- JSFIXはテストに失敗したプロジェクトのほとんどを修正することができた.
- JSFIXが生成した31のプルリクエストが受け入れられた.

所感

- ・ パッチは手動で作成しなければいけないのかと感じた.
- ・ パッチを実際にどう適用しているのかの説明があまり理解できなかった.