

氏 名 Tao Zan

学位(専攻分野) 博士(情報学)

学位記番号 総研大甲第 1880 号

学位授与の日付 平成28年9月28日

学位授与の要件 複合科学研究科 情報学専攻  
学位規則第6条第1項該当

学位論文題目 A Putback-Based Approach to Bidirectional Programming

論文審査委員 主 査 教授 胡 振江  
教授 中島 震  
准教授 吉岡 信和  
助教 加藤 弘之  
助教 対馬 かなえ  
准教授 中野 圭介 電気通信大学

論文内容の要旨  
Summary of thesis contents

Bidirectional transformation is a promising approach for maintaining consistency between two related information (we call them source and view), and many bidirectional programming languages are designed that let programmer merely write one program which denotes both a forward transformation (get) from source to view and a backward transformation (putback) that updates the source with the view. Since usually source and view are not in a one-to-one correspondence, the backward transformation (change propagation from view back to source) is inherently ambiguous. Nevertheless, existing bidirectional transformation languages focus mainly on enforcing consistency and provide developers only limited control over the backward transformation. Programmer writes the get transformation, and one “suitable” putback transformation will be derived from this get. While in practice it is impossible to decide which one is “suitable” in general and what is more, programmer has no choices of defining the putback behavior he/she wants. We proposed a new programming by update paradigm for putback-based bidirectional programming based on the fact that a well-behaved put uniquely determines get. If we can design a user-friendly putback-based bidirectional programming language that lets programmer write the putback function which behaves simply like an update on the source and the language is well-designed to satisfy the well-behavedness, a unique get function can be derived for free. In this dissertation, we designed and implemented a putback-based bidirectional programming language BiFluX which can be used in many real world applications such as program refactoring, and self-adaptive systems. We also implemented a bidirectional update library for relational database.

We proposed a new core bidirectional update language for XML structured data which consists of bidirectional updates, patterns, and XML related expressions and paths. Since this core language contains many XML related features, it is hard to design a clear bidirectional semantics. So we distilled another clean generic core bidirectional programming language BiGUL from the core of the BiFluX language. BiGUL is a combinatorial language that consists a set of putback combinators which can be used to compose large bidirectional programs. The language is fully verified to guarantee that any program written in BiGUL is well-behaved. It has been implemented in Haskell and served as the core language for the putback-based bidirectional programming language BiFluX.

BiFluX is our first try of designing putback-based bidirectional programming languages, which targets XML structured data. XML is widely used in data exchanging on the web, and existing works such as bidirectionalization of XQuery has the same problem as explained before that programmer has no choices of specifying the backward behavior. We design the bidirectional programming language BiFluX simply as an update language for programmers that they only need to write the putback of a bidirectional transformation as an update program that uses a view XML to update a source XML. The most significant design of BiFluX is the source-view matching which matches a sequence of source elements with a sequence of view elements either by position or by some keys and each matched source-view pair is again handled by another subprogram. BiFluX is expressive enough that supports if-then-else condition, case analysis, pattern matching, and recursive definition of a bidirectional update program. A BiFluX program is firstly normalized into the core XML bidirectional update language and then compiled into BiGUL. The well-behavedness of a BiFluX program is guaranteed by the underlying BiGUL.

We also implemented a bidirectional update library Brul for relational database that provides two library functions to

(別紙様式 2)  
(Separate Form 2)

simply let programmer write update program by using view table to update source tables. The update program can also be interpreted as a query such as selection, projection and join in the get direction. The library function align covers the selection and projection, and unjoin covers join in relational algebra. Brul is more powerful than relational lenses, since it only provides limited control of the putback direction, while Brul allows programmer to describe flexible update strategies.

Our putback-based bidirectional programming languages have been used in many real world applications. Cheng et al. use BiFluX to support reflecting updates on refactored code to the original source code in Java and vice versa. Lionel et al. utilize BiFluX to implement the BXauthZ which is a policy language to express attribute based rules on XML views. Zhu et al. propose a declarative bidirectional language BiYacc which supports reflective printing and parsing implemented on top of BiFluX. Zhao et al. designed a rule-based language vRule for self-adaptation system which is implemented based on BiFluX. Colson et al. use BiGUL to implement a reusable self-adaptive system which synchronizes the configuration files of different servers such as Apache and Nginx.

博士論文の審査結果の要旨  
Summary of the results of the doctoral thesis screening

本論文は、双方向変換の振る舞いを完全に記述できるプログラミング言語の設計と実現に関するものである。双方向変換とは、ソースデータをターゲットデータに変換した後、ターゲットデータ上の更新をソースデータに反映させることが可能な計算の枠組みのことである。古くはデータベース分野におけるビュー更新問題として扱われてきたが、近年は新しいプログラミングモデルと進化的ソフトウェア開発の手法として注目を浴び、プログラミング言語の観点から様々な双方向変換言語が提案されている。これらの双方向変換言語は、順変換言語に逆変換の意味を持たせることによって一つのプログラムを順変換でも逆変換でも実行できるようになる。しかし、ある順変換に対して一般に複数の逆変換が存在するので、ユーザの意図を反映した逆変換を自由に記述できず逆変換の予測性と信頼性が欠如している。一方、逆変換が決まればそれに合う順変換が唯一に決まるという理論的な結果があるが、逆変換は順変換よりかなり複雑であるため、それを簡潔で構成的に記述できるかどうかは未解決な問題である。本論文は、双方向プログラミングに対する逆変換を基軸とする新しい方法論を提案し、既存の更新言語の拡張により、逆変換を基軸として、XMLの双方向変換を記述しやすい双方向変換言語BiFLux及びテーブルの双方向変換を記述しやすい双方向変換言語Brulを設計・実現・応用を示している。これにより、逆変換を基軸とする双方向プログラミングを支援する環境ができ、双方向変換の実用化への大きな足がかりになっている。

本論文は英語で記述されており、全6章から構成されている。

第1章は序論である。研究の背景、研究目的、主要な貢献など論文全体の構成を述べている。

第2章は基礎知識の紹介である。双方向変換の歴史と基礎知識、順変換を基軸とする双方向変換言語の紹介とその問題点、逆変換を基軸とする双方向変換の理論的な枠組みについて議論している。

第3章では、双方向変換における「逆変換」がビューを利用してソースを「更新」という考え方に基づいて、XMLを更新するための更新言語を拡張し、逆変換を記述するためのコア言語の設計を示している。また、このコア言語をXMLに限らずより汎用的なものにすることができると示している。

第4章では、第3章で議論しているコア言語に基づいて、XMLを更新するための関数型言語FluXを拡張し、XMLを対象とする新しい双方向変換言語BiFluXの設計と実現を議論している。

第5章では、第3章の汎用的なコア言語に基づいて、関係データベース上の各種のビューの更新戦略を柔軟に記述できる双方向変換言語Brulの設計と実現を議論している。

第6章は論文のまとめと今後の課題である。BiFLuXは他研究グループによりコードリファクタリング、コードクローン、自己適応システムへ応用されることを報告し、提案した逆変換を基軸とする一連の双方向変換言語の新規的で有効であると結論づけるとともに、言語の拡張とシステムの最適化についての将来の課題を論じている。

審査会において、出願者は上述の内容に沿って説明を行い、そのあと審査委員との質疑応答を行い、的確な回答がなされた。なお、論文成果は海外論文誌論文1件（査読あり）、査読付き国際ワークショップ論文2本、共著雑誌論文1本、共著国際会議論文4本の発表という業績をあげている。また、開発したシステムもウェブで自由にダウンロードできるようになっている。

(別紙様式 3)

(Separate Form 3)

以上、本論文では、双方向変換の実用に向けて、逆変換を基軸とする双方向変換プログラミング言語の実現を行い、逆変換を基軸とする双方向変換を系統的に構築する言語基盤とプログラミング環境を与えた。本研究は双方向変換を一層身近なものとなり、スマートシステムや自己適応・自己進化ソフトウェアなど種々の複雑な同期システムの構築への応用が期待できる。よって、6名の審査委員全員一致で、本学位請求論文は学位を授与するのに十分なレベルであるものと判定した。