

単一方向性制約伝搬とパターンに基づく 業務システム構成法の提案

矢野寛将[†] 桑山浩希[‡] 金田重郎[†]

同志社大学理工学部[†] 同志社大学院理工学研究科[‡]

1. はじめに

著者らは単一方向性制約伝搬モデルによって業務処理を実現することを既に提案している¹⁾。しかしその実現方法については、実装が複雑となっており十分に議論されていない。

そこで本稿では、単一方向性制約伝搬とソフトウェアパターンに基づく業務システムの構成法を提案する。業務処理で扱うデータ間には制約の関係があり、入力側で値の追加・更新を行うと、その制約によって出力側の値も追加・更新が行われる。このモデルをアナリシスパターン²⁾の転記ルールを用いて設計を行う。

提案手法の評価のため、オブジェクト指向言語を用いて京都府と京田辺市の条例が定める23税すべての税務処理を実装し、業務システムに対する提案手法の適用性を確認した。また、税額計算後に入力条件を修正する等の複雑な処理も付加コードなしに実現できた。

2. 単一方向性制約伝搬

業務処理で扱うデータ間には制約の関係がある。具体的には地方税法の自動車税の場合、日付及び自動車の所有、税の賦課の3つの関係には制約がある。4月1日に自動車が所有されていれば税が賦課される。また、4月1日に税が賦課されていれば、自動車は所有されている。これをデータ間の制約という。

業務データ間の制約は単一方向に伝搬しているとみることができる(図1)。具体的には自動車税の場合、自動車が存在するかどうかや自動車の属性、日付から税の存在・非存在や税額を

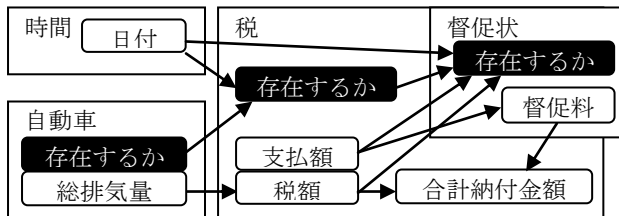


図1 単一方向性制約伝搬

決定する。また税から督促状の存在・非存在や督促料、合計納付金額を決定する。入力データの追加・更新によって出力データが追加・更新される。このように制約が単一方向に伝搬することで業務データ間の整合性を常に保つ。

3. 提案手法

3.1. パターンを用いた業務知識のモデル

本提案手法では入力オブジェクトと出力オブジェクトの1対1の関係に対して制約を記述し、入力データを出力側へ単一方向に伝搬させる(図2)。1対1の関係にすることで、自動車税への制約や軽自動車税への制約などをそれぞれ同じクラスへ汎化関係をつくることができる。

また本手法ではアナリシスパターン²⁾における転記ルールを基礎とし制約を用いる。システムにオブジェクトの入力や入力オブジェクトの値の変化が起こる。そのときオブザーバパターンによって入力オブジェクトが制約に伝搬され、制約がチェックされ、出力オブジェクトの生成や値の変化が起こる。入力オブジェクトの存在が出力オブジェクトの存在を決める。オブジェクト単位で考え単一方向性制約伝搬を実現する。

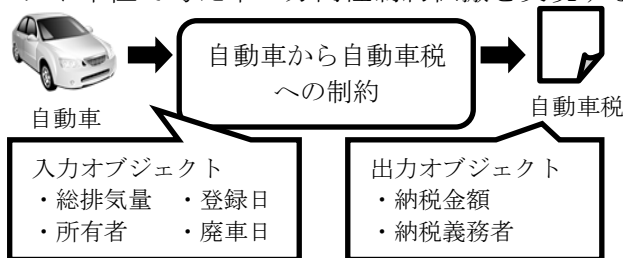


図2 自動車と自動車税の関係

3.2. 制約の表現方法

制約では、日付や入力オブジェクトの属性の入力状態によって出力オブジェクトの属性を決定する。地方税法では4月1日に自動車に対して課税される。そのため図2の制約は表1に示す制約として表現する。

具体的には自動車がシステムに入力されると、現在が4月1日という入力状態のとき、自動車税を生成し税額を計算する。4月1日より後のとき、自動車税の属性の更新を行う。4月1日より前のとき、タイミングホイールに登録し、4月1日に制約をチェックするように設定する。

An Implementation Methodology for Business Systems Based on Unidirectional Constraint Propagation and Patterns

[†]Hiromasa Yano, Shigeo Kaneda, Faculty of Science and Engineering, Doshisha University

[‡]Hiroki Kuwayama, Graduate School of Science and Engineering, Doshisha University

表1 自動車と自動車税の制約

入力状態		出力
日付	自動車	自動車税
現在は 4月1日以降	存在	税額の再計算（存在しなければ生成・更新。）
現在は4月1日	存在	生成して税額計算。
現在は 4月1日より前	存在	4月1日に制約をチェックするように設定。

4. 評価

4.1. 地方税務処理システムの実装

提案手法を用いて、京都府と京田辺市の条例が定める 23 税¹すべての地方税の税務処理が 1 人によって 4ヶ月ほどで実装できた。システムの全体構成を図3に示す。自動車税は GUI, その他の税はコンソールを用いて実装できたことを確認した。実装言語は Scala と Java である。

税務処理は具体的に、税の賦課・督促・催告・還付の計算 (23 税), 納税通知 (7 税) を適切な日付に行った。ただし固定資産税・住民税・事業税の正確な計算を除く。作成した全クラスの総ステップ数は、View クラスを除き 6065 行となった。

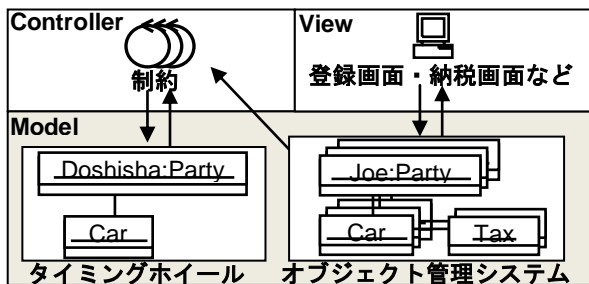


図3 システムの全体構成

4.2. 制約による複雑な修正処理の実現

単一方向性制約伝搬によって実現が確認された処理は、自動車税を例に挙げると、

- ・自動車登録日の変更による税額の再計算
- ・所有者の変更によって、3月であれば納税義務者の変更

・自動車の属性変更による税額の再計算
 ・自動車廃車日の変更による還付金金額の変更
 などである。View クラスでオブジェクトの属性を変化させる処理によって、税額計算後に入力条件を修正するなどの複雑な処理が実現できた。

4.3. 従来手法との比較

従来手法及び提案手法の軽自動車税プログラムそれぞれについて、COF(Coupling Factor)³⁾を測定した。また作成したクラスについて、制約クラス、制約クラスのうち関数が共通に利用され

¹当該都道府県の地方消費税は国が徴収する、かつ他の 46 都道府県への支払額計算のみのため評価対象から除いた。

るクラス及びスーパークラス、軽自動車・税・時間クラスなどデータのクラス、データのクラスのうちスーパークラス、データの追加・更新などを管理するデータ管理部のクラスそれぞれのステップ数を測定した。その結果を表2に示す。ただし、Controller と Model クラスを測定の対象とし、名前のみのクラスを対象外とした。

COF は(1)(2)に定義される。C_c⇒C_sはクラスC_cがクラスC_sの属性やメソッドなどを参照していること、C_c→C_sはC_cがC_sを継承していること、DC(C_i)はC_iを継承しているサブクラスの数を示す。TCは作成したすべてのクラスとした。

$$COF = \frac{\sum_{i=1}^{TC} [\sum_{j=1}^{TC} isClient(C_i, C_j)]}{TC^2 - TC - 2 \times \sum_{i=1}^{TC} DC(C_i)} \quad (1)$$

$$isClient(C_i, C_j) = \begin{cases} 1 & \text{iff } C_c \Rightarrow C_s \wedge C_c \neq C_s \wedge \neg(C_c \rightarrow C_s) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

表2 軽自動車税プログラムの比較

手法	COF (%)	作成したクラスのステップ数(行)			
		制約(関数 共通利用 クラス及 びスーパー クラス)	データ (スーパー クラス)	計	データ 管理部
提案	7.6	844 (687)	619 (363)	1463	621
従来	12.6	1490 (1)	295 (59)	1785	80

5. まとめ

入力オブジェクトと出力オブジェクトの1対1の関係に対して制約を記述しオブジェクトの関係性を減らすことで、提案手法の COF 値は従来手法の 0.60 倍となり実装の複雑性を低下させた。

またアナリシスパターンを用いて、自動車や自動車税などの「知識」、制約という「操作」を分ける。これによりオブジェクトが他のオブジェクトを操作する範囲を適切に分割できる。これらの結果と、23 税の基本処理が View を除き 6065 行で実装でき、設計の妥当性が確認された。

また単一方向性制約伝搬によって修正処理を容易に実現できた。これは制約によって入力オブジェクトの存在が出力オブジェクトの存在を決定し、入力データと出力データ間の整合性が常に保たれるためである。

主な参考文献

- 1) 桑山浩希, 中西義尚, 金田重郎: 単方向性制約伝搬に基づく業務システム構成法の提案, 第75回全国大会講演論文集, pp.771-773 (2013).
- 2) マーチン・ファウラー (著), 堀内一 (監訳), 友野晶夫 (訳): アナリシスパターン[新装版], 株式会社ピアソン・エデュケーション (2002).
- 3) Abreu, B.F., Goulao, M. and Esteves, R.: Toward the Design Quality Evaluation of Object-Oriented Software Systems, Proc. the 5th International Conference on Software Quality(1995).