

UML クラス図の自然言語化に基づくクラス図自己レビュー支援環境の構築

Construction of Self-Review Support Environment based on Natural Language Translations from UML Class Model

西畑 瞬^{*1}, 高倉 将希^{*1}, 岸本祐季^{*1}
 Shun NISHIHATA^{*1}, Masaki TAKAKURA^{*1}, Yuki KISHIMOTO^{*1}
 野口 靖浩^{*1}, 小暮 悟^{*1}, 近藤 真^{*1}, 小西 達裕^{*1}
 Yasuhiro NOGUCHI^{*1}, Satoru KOGURE^{*1}, Makoto KONDO^{*1}, Tatsuhiko KONISHI^{*1}
^{*1}静岡大学情報学部
^{*1}Faculty of Informatics, Shizuoka University
 Email: cs16070@s.inf.shizuoka.ac.jp

あらまし：クラス図を用いた設計の学習方法として、要求に基づくクラス図を作成して相互レビューする方法がある。レビューに際しクラス図を自然言語に置き換えて解釈することが良く行われるが、初学者が自信を持ってクラス図を解釈することは難しい。そこで、本研究では、学習者の選択したクラス図中の要素の組み合わせを自然言語化して提示する支援システムの構築とともに、学習者の気づきを促進する言語化のタイプを検討した。評価実験から、誤り・設計漏れの発見・修正に対して有用であると確認した。
 キーワード：UML モデリング, モデリング教育, ソフトウェア開発教育, レビュー支援, 自然言語化

1. はじめに

近年、システム開発の場ではシステムの内部構造の把握と開発者間での設計共有の重要性が高まっており、UML 等の設計記述言語でモデルを記述することや、記述されたモデルを読むことも増えている。

クラス図を用いた設計の学習方法のひとつとして、与えられた要求仕様に基づいて考案したモデルをクラス図で記述し、それを相互レビューすることで学習する方法がある。しかしながら、初学者にとってはクラス図で記述されたモデルを読み取り、仕様と照らし合わせて誤りや設計漏れを発見することができない場合がある。また、クラス設計は解がひとつに定まらないため、互いの指摘の妥当性を初学者同士で判断することが難しい。田中ら⁽¹⁾はオブジェクトモデリング演習を分析し、初学者が図の間違いに自分で気づけない場合があることを報告している。

クラス図の学習支援システム⁽²⁾⁽³⁾が存在するが、学習者の作成するクラス図と比較するために正解を予め設定しておく必要がある。そのため、課題として与える要求仕様は、学習者が用いるべきオブジェクトを規定するなど、ある程度の設計案を与える必要があり、演習の自由度は限定される。

クラス図の一般的なレビュー方法のひとつとして、クラス図の意味を自然言語に置き換えて解釈する方法がある。これはクラス図の各要素に意味が規定されていることに基づく方法で、クラス図と仕様間の定式的な変換についても研究されている⁽⁴⁾⁽⁵⁾。萩原⁽⁶⁾は、要求に含まれる目的語と動詞に着目し“クラス図を読み上げたとき機能を表現する文章になる”ようにクラス図に落とし込む手順と基準を整備することで、モデリングのトレーニングを実施し成果を上げている。しかしながら、UML 初学者にとっては、クラス図自体の学習と、手順と基準の学習を並行して行い、その手順と基準に正しく従っていることを

自己確認しながら演習を進めることは簡単ではない。また、クラス図の各要素に規定された意味に基づく自然言語表現からは、誤りや設計漏れに関する気づきを得ることができない場合がある。特にクラス図の要素によって明示的に記載されていない事柄についての気づきを得ることは難しい。

そこで本研究では、学習者がクラス図をレビューする際に自動的にクラス図を自然言語化して提示することのできる学習支援システムを構築した。その結果、学習者のクラス図自体のレビューを支援するとともに、クラス図と要求仕様との比較も支援することができる。更にクラス図の各要素に規定された意味に基づく自然言語化に加えて、誤りや設計漏れに関する気づきを促進する自然言語化を組み込んだ。

2. システムの概要

クラス図と仕様間の形式的な変換に関する研究⁽⁴⁾⁽⁵⁾では、クラス図の要素に規定された意味に基づく自然言語化をテンプレートによって実現している。

本研究ではクラス図の以下の要素の組み合わせに対するテンプレートを設計した。

- ・クラス・属性・操作
- ・可視性
- ・関連・ロール・多重度・誘導可能性
- ・集約・コンポジション
- ・汎化・実現・インタフェース
- ・依存

テンプレートの設計方針は以下の3通りである。

- (A) クラス図の要素に規定された意味に基づくテンプレート
- (B) クラス図の要素（の組み合わせ）により実現される内容に基づくテンプレート
- (C) クラス図の要素（の組み合わせ）の典型的な設計意図に基づくテンプレート

(A) のタイプのテンプレートは、例えば「会員」と「本」のクラスが関連の関係を持った設計 (図 1) において、学習者の定義した関連名、誘導可能性等から生成された「本が会員を借りる」という表現から、本来「会員が本を借りる」という設計を考えていたにも関わらず、正しくクラス図に反映できていないことに気づくことができる。

(B) のタイプのテンプレートは、例えば「司書」と「会員」が誘導可能な関連を持ち「会員」が「返却依頼」の公開関数を持つ設計 (図 2) の場合に、「司書は会員に返却依頼を依頼できるか定かではない」という表現から“関連を持つクラスに公開関数が定義されているが、誘導可能性が未定義である”設計が示す内容を確認し、誘導可能性の定義が不十分なために意図した設計になっていないことに気付くことができる。

(C) のタイプのテンプレートは、例えば「会員」に対する「本」の多重度が“1”の設計 (図 3) の場合に「会員は常に1つの本を借りる」という表現から複数冊の本を借りる状態及び1冊も本を借りていない状態が規定されていないことに気づくことができる。



図 1 (A) のテンプレートが役立つ設計例



図 2 (B) のテンプレートが役立つ設計例

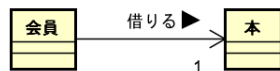


図 3 (C) のテンプレートが役立つ設計例

本システムのインタフェースを図 4 に示す。本システムはモデリングツール Astah*のプラグインとして実装しており、図中の①で任意の要素群を選択し、②の RUN ボタンを押すことで、選択した要素群に対応する自然言語表現を③に表示する。現在は (B) 以外の (A) (C) のタイプを中心に計 15 個のテンプレートを実装している。

3. 評価実験

学習者に対してクラス図内の誤りと設計漏れの発見と修正を支援する効果があることを確認するため、評価実験を実施した。クラス図の記述方法を学んだことのある被験者 6 名を対象とした。被験者には仕様書を与え、仕様を実現するクラス図を作成させ、各自が納得の行くまでレビュー・更新した後に提出させた。その後、システムを利用して自身の書いたクラス図をレビューさせ、クラス図内の誤りと設計漏れを指摘させた (制限時間 30 分)。更にその誤りと設計漏れを修正させた (制限時間 30 分)。評価実

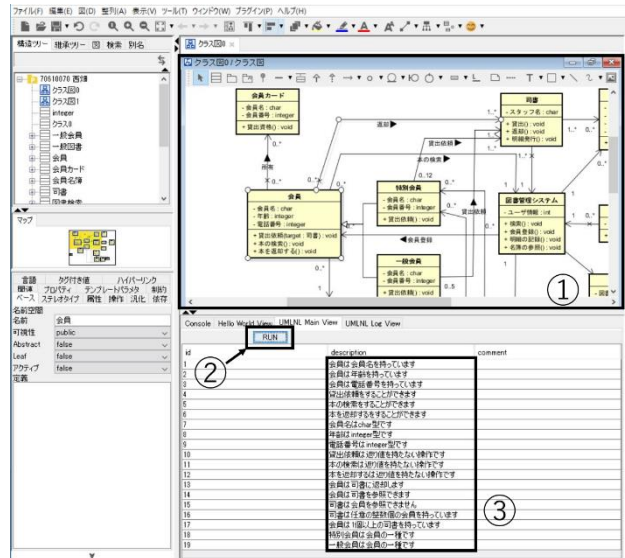


図 4 システム外観

験の結果、上述の効果が確かめられた。

4. まとめ

本稿では、自然言語化に基づくクラス図自己レビュー支援環境を提案した。評価実験により、学習者に対してクラス図内の誤りと設計漏れの発見と修正を支援する効果を確かめた。

今後は学習者に気付きを与えるシステムの自然言語生成に関して、特に (B) (C) のタイプのテンプレートを充実させていきたい。

参考文献

- (1) 田中昂文, 森一樹, 橋浦弘明, 樫山淳雄, 古宮誠一: “オブジェクトモデリング演習における学習者にとっての難所の検出方法と支援システムの提案”, 情報処理学会研究報告 Vol.2014-SE-183 No.10 (2014)
- (2) J. Schramm, S. Strickroth, L. Nguyen-Thinh, N. Pinlwart: “Teaching UML Skills to Novice Programmers Using a Sample Solution Based Intelligent Tutoring System”, Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, pp.472-477 (2012)
- (3) J. Soler, I. Boada, F. Prados, J. Poch and R. Fabregat, “A web-based e-learning tool for UML class diagrams”, IEEE EDUCON 2010 Conference, pp. 973-979 (2010)
- (4) F. Meziane, N. Athanasakis, S. Ananiadou: “Generating Natural Language Specifications From UML Class Diagrams”, Requirements Engineering, Vol.13, Issue.1, pp 1-18 (2008)
- (5) P. Liang, A. Romanczuk, J. Royer: “A Translation of UML Components into Formal Specifications”, Technology of Object-Oriented Languages, Systems and Architectures, Springer, Boston, MA, pp. 60-75 (2003)
- (6) 萩原豊隆: “セイコーエプソンのモデリングトレーニング～センスに依存しないモデリングを実現する～”, UMLTP モデリング技術部会 (2013)