3J-02

# SOFL 非形式仕様の図化支援ツールの開発

鈴木　優也†　　劉　少英‡

法政大学情報科学部†　　法政大学情報科学部‡

## 1. まえがき

ソフトウェア業界は急激に発展し，ソフトウェアの複雑化・大規模化が進んでいる．それに伴い，ユーザと開発者の厳密かつ実用的なコミュニケーションを促すことが課題になっている．SOFL (Structured Object-oriented Formal Language)[1]は非形式仕様，半形式仕様及び形式仕様の三段階の仕様記述形式からなる開発手法である．この内，最も抽象度の高い非形式仕様記述は，開発者がユーザの要求を抽象レベルで全面的に掴み，自然言語で文書化するために，機能，データリソース及び制約を構造化し，簡潔に表す記述法である．しかしながら，テキストベースの表現による，ユーザと開発者との間のコミュニケーションには限界がある．この問題を解決するため，SOFL 非形式仕様を半自動的に図化し，動的に編集する機能をもったツールを開発する．このツールによって，SOFL 非形式仕様で表された内容の表現に多様性をもたせ，フィードバックに応じた動的な編集を可能にすることで，ユーザと開発者の厳密かつ効率的なコミュニケーションの促進を試みる．

## 2. 設計

本システムは，テキストベースな表現である SOFL 非形式仕様を図化し顧客からフィードバックを得，それに応じて図を動的に編集することで，ユーザと開発者のコミュニケーションをサポートし，漸進的に非形式仕様を完成させるためのサポートツールである．

図表現の方法については UML ユースケース図を基盤とした表現で行う．理由は，要求分析の場面で使われることが多いこと，システムの機能に重点を置いていること，機能の表現に階層構造をもつことが SOFL 非形式仕様と似ているためである．

Development of a Software Tool for Visualizing SOFL Informal Specifications

†Suzuki Yuya・Hosei University Faculty of Computer and Information Sciences

‡Liu Shaoying・Hosei University Faculty of Computer and Information Sciences

### 2.1. SOFL 非形式仕様

SOFL 非形式仕様は，SOFL の三段階の記述の内最も抽象度が高く，最初に設計される仕様書であり，ユーザの要求を抽象レベルで全面的に掴み，自然言語で文書化するため，システム機能(Function)，データリソース(Data resource)，データと機能の制約条件(Constraint)を簡潔に定義する．定義された高いレベルの機能，データリソース，制約条件は低いレベルの機能，データリソース，制約条件に分解し，構造的に示すことが出来る．図 1 に SOFL 非形式仕様の銀行口座による事例を示す.

1 Functions
- 1.1 銀行口座の開設
- 1.2 預金 (F1.5)
- 1.3 お金の引き出し (F1.5)
- 1.4 残高の表示 (F1.5)
- 1.5 口座間の振り込み
  - 1.5.1 振り込み先の口座情報を得る。
  - 1.5.2 振り込み金額のチェック。
  - 1.5.3 振り込みを行う。
- 1.6 口座番号とパスワードの照合

2 Data Resources
- 2.1 一人のカスタマの口座情報 (F1.1) (F1.2) (F1.3) (F1.4) (F1.5) (F1.6)
  - 2.1.1 名前
  - 2.1.2 口座番号
  - 2.1.3 パスワード
- 2.2 全てのカスタマ口座情報 (F1.1) (F1.2) (F1.3) (F1.4) (F1.5) (F1.6)

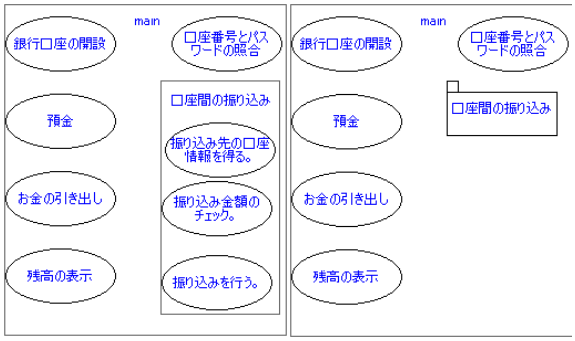
3 Constraints
- 3.1 最大五万人は貸すたまの口座を対応できる (F1.1) (D2.1)
- 3.2 一人のカスタマは，一つの口座しか持つことが出来ない (D2.1)

図 1 SOFL 非形式仕様の例

### 2.2. SOFL 非形式仕様と図の対応

SOFL 非形式仕様について，ユースケース図を元にした図によって表現を行う．非形式仕様の一つの機能について，図の一つのユースケースつまり楕円で表現される図形を対応させる．しかし，非形式仕様内の一つの機能がさらに小

さい機能に分解されている場合は，ユースケース図のサブジェクトの形，もしくはパッケージの形で表現する．また，一つの機能がパッケージによって表現された場合，下位の機能を表現に含まないため，分割されている機能については，分割されている機能を根とした部分木に対応した図も作成する．SOFL 非形式仕様のデータリソースと制約条件に関しては，新たに表現を加える必要がある． SOFL 非形式仕様の自動変換の例を図 2 に示す.




サブクラスによる表現　　パッケージによる表現

図 2 変換の例

### 2.3 SOFL 非形式仕様から図への変換

本システムは，SOFL 非形式仕様を読み込むことで，半自動的にユースケース図を元にした図へ変換することが出来る．具体的には，SOFL 非形式仕様の機能に対応したユースケース，もしくはサブクラスやパッケージを作成することである．その他の，図内のアクターを示すスティックマンや，関係を表す実線は，本システムの利用者が編集を行うことが必要である.

### 2.4 本システムの編集機能

本システムは，自動で変換できない部分利用者による補填や，フィードバックによる修正の適用のため，図形のユースケースやアクターの名前の変更及び移動，構成内容の追加と削除などが可能である.

## 3. 実装

実装は，visual C#で，IDE は Visual Studio 2013 を利用した．図 4 に，実行の画面を示す.

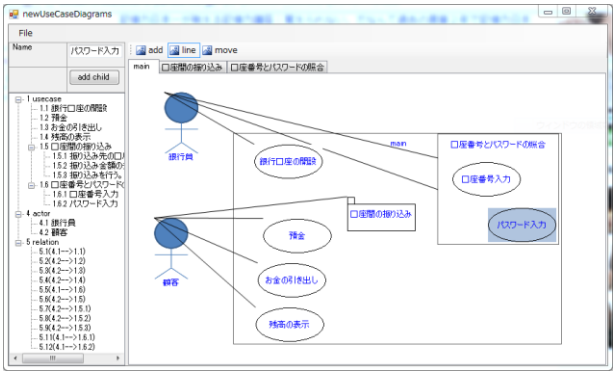



図 4 実行画面

## 5. まとめ

要求分析におけるユーザと開発者の厳密かつ効率的なコミュニケーションを促すことを，形式工学手法である SOFL の非形式仕様を半自動的に図化し，それを動的に編集できるツールを開発することで試みた．SOFL 非形式仕様に対する図は，UML ユースケース図を参考に行った．このツールにより，テキストベースな SOFL 非形式仕様に図による表現が与えられた上，フィードバックに応じてその場で図表現を編集することが可能になった．しかしながら，この表現は SOFL 非形式仕様のシステム機能の記述に対する表現に偏重しており，データリソースや制約条件に対する表現があまりなされていない上，本システムを操作できない状況(画像としてやり取りしたい場合など)や，すべての情報をなるべく人目で確認したい場合などでは利用できないため，有効な表現の提案が期待される．本ツールの図表現などの情報から，目的とするソフトウェアに対する GUI を自動で作成し，現実に近い表現でさらにフィードバックを得やすくするツールの開発[2]などが期待される.

## 文　献

Formal Engineering for Industrial Software Development Using the SOFL Method, by Shaoying Liu Springer-Verlag, March 2004 , ISBN 3-540-20602-7

Shaoying Liu and Fauziah binti Zainuddin and Mo Li, “Integrating Animation into Informal Specification Writing for Requirements Analysis,” @SDIWC , pp.137－143, 2014.