操作対象の制約と操作の関係の明確化に関する研究

M2012MM011 鹿島一彦
指導教員：野呂昌満

1　はじめに

操作対象を直感的に扱う事が可能なユーザインタフェイス (以下，UI) は，操作対象の型や型の間の関係に関する制約によって異なる．操作対象の型や型の間の関係に関する制約が定義されている例として，ソフトウェアモデルがあげられる．型の間の関係に関する制約を考慮した操作を行なう場合，複数の対象を同時に扱うことが必要となりうる．マルチタッチインターフェイスでは，ユーザは複数の対象を扱うことができるので，操作対象を直感的に扱うことが可能である．

操作対象を直感的に扱うことが可能なマルチタッチ操作を，操作対象がもつ制約を考慮して設計する方法が必要である．グラフ構造で表現されたモデルに対するマルチタッチ操作を，Heydekorn ら [1] は定義している．定義されたマルチタッチ操作は，一点のみ操作可能な UI が提供する操作がもととなっており，複数の対象を同時に扱う必要がある操作の分析が行なわれていない．定義されたマルチタッチ操作では，ユーザがもつ対象の型や型間の関係についての理解だけで操作を行なうことは難しい．

本研究の目的は，図式表現で記述されたソフトウェアモデルの制約を考慮した操作の設計のためのガイドラインを提案し，直感的に扱うことが可能な操作の設計を支援することである．整理した制約と操作の関係を利用することで，操作対象を直感的に扱うためのマルチタッチ操作の設計が可能となる．

本研究では，事例を通して，操作対象がもつ制約と操作プロセスの関係の明確化と，操作の設計を行い，それらをもとに操作を設計するためのガイドラインを提案した．事例として，一般に利用され，問題の典型的な例を提示できると考えた Unified Modeling Language[2](以下，UML) 標準のクラス図を編集するエディタを採用した．関係を明確化する方法と，操作を設計する方法を整理することで，操作対象を直感的に扱うことが可能な操作を設計するためのガイドラインを提案した．

本研究では，操作対象の制約を分析して分類し，制約の分類と操作の対象を対応付けることで，制約と操作の関係を明確化した．操作対象のメタモデルと，一般に存在する制約を考慮して，制約を分析し，分類を行なった．分析した制約の分類に対して，操作を対応付けることで，制約と操作の関係を明確化した．

本研究では，明確化した関係を利用して行なった操作の設計を例示した．設計する操作が扱う対象をもとに，明確化した関係から操作プロセスを選択し，操作プロセスを組み合わせた．操作プロセスを組み合わせる際は，操作の識別が可能であることと，操作間の並行性を考慮した．

本研究の結果として，操作対象を直感的に扱うことが可能な操作を設計するためのガイドラインを提示した．いくつかの操作の設計を行ない，操作の前後の操作対象と制約の関係から，設計した操作が妥当性を確認した．

2　研究の背景

グラフ構造で表現されたモデルに対するマルチタッチ操作を Heydekorn らは定義している．Heydekorn らは，ユーザが好むマルチタッチ操作の分析を目的とし，マルチタッチ操作の定義に注力していない．

Heydekorn らは，一点のみ操作可能な UI が提供する操作をもとにして，マルチタッチ操作を定義している．定義されたマルチタッチ操作の一部を抜粋し，図 1 に示す．図 1 は，リンクを生成するための操作である．リンクは 2

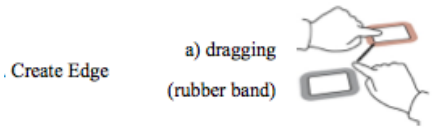

図 1　リンクの生成操作

つのノードを結ぶ要素であるので，直感的にリンクを生成しようとした場合，2 つのノードの指定が必要である．図 1 の操作は，マウスを利用した操作をもとに定義されているおり，ノード間のスワイプを含む．リンクを生成しようとした場合，スワイプが不要である．グラフ構造で表現されたモデルの構文の理解だけで，リンクを生成する操作を行なうことは困難である．

3　提案するガイドラインの概要

制約と操作の関係の整理方法，操作の設計方法をガイドラインとして提案する．それぞれについて述べる．

制約と操作の関係の整理は，制約の分析と，操作の対象の分析をもとに行なう．制約を分析し，制約の分類を定義する．操作の対象の分析結果と，制約の分類と対応付けることによって，制約と操作の関係を整理する．制約を考慮して，ユーザが行なうタッチ操作と，タッチ操作のプロセスを定義する．操作に必要な情報をもとに操作間の並行性を分析し，タッチ操作のプロセスを定義する．

操作の設計方法について述べる．制約と操作の関係と，設計する操作に関連する制約から，操作プロセスを選択する．選択した操作プロセスを組み合わせて操作の設計を行なう．操作プロセス間の並行性を考慮し，他の操作と識別可能な操作を設計する．

4　操作対象の制約と操作の関係の明確化

4.1　操作対象がもつ制約の分析

本研究では，UML 標準のクラス図のメタモデルから抽出した制約を分析した．制約が操作対象のメタモデルに起因するという認識のもと，メタモデルの要素と制約の関係を分析した．また，制約の種類によって，関連しうる操作の

種類が異なるという認識のもと，一般に存在する制約の種類を考慮し，制約の分析を行なった．一般に存在する制約として，制約記述のための標準である Object Constraint Language[3] (以下，OCL) の記述例を参照した．一般に存在する制約を記述可能である OCL では，UML の形式的に記述すべき制約が網羅されているので，考慮すべき制約の種類を整理できると判断した．

4.1.1　クラス図のメタモデルの記述による分析

UML 標準で示されているクラス図のメタモデルの記述をもとに制約を分析した．UML 標準では，メタモデルの要素ごとに制約が定義されている．また，各制約は，メタモデルの要素間の関係によって，対象を制限する．これらから，制約をメタモデルの各要素と，メタモデルの要素間の各関係に分類できると判断した．

4.1.2　OCL の記述による分析

OCL の記述例をもとに，抽出した制約を分析した．OCL の記述例から，OCL の構文要素の種類によって，制約を分類できると考えた．OCL の構文要素から，制約を存在，依存と，値に分類することができると考えた．

4.2　制約と操作の関係の分析

4.1.1, 4.1.2 項で分析した制約の分類ごとに関係する操作を分析した．モデルに対する操作は，メタモデルの要素のインスタンス間の関係を変更しうる．このことから，操作をメタモデルの要素間の関係ごとに整理できると考えた．分析した制約と操作の関係の一部を表 1 に示す．

表 1　制約と操作の関係の抜粋

対象の要素	制約	制約の分類	操作
Association	関連端は常に2つ以上存在する	存在 Association型のassociationと多重度が2以上のProperty型のmemberEndの関連	Associationの生成
	Associationが削除された際，削除されたAssociationに関連するProperty型のownedEndも削除する	Association型が全体を表す要素である合成を表すhas-a関係	Associationの削除
	別の関連を特化する関連は，他方の関連と同数の関連端をもつ	依存 Classifier型のclassifierとClassifier型のgeneralの依存	is-a関係をもつAssociationの生成
	一般化階層内で循環がない		

4.3　操作プロセスの定義

制約の分類ごとに，関連する操作のプロセスを定義した．制約と制約を満たす操作対象の例をもとに，必要となる操作を分析し，操作間の順序を整理した．整理した操作の順序からマルチタッチ操作を定義した．定義したマルチタッチ操作をもとに，操作プロセスを定義した．

指定された Association と is-a 関係をもつ Association の生成操作を例に，操作プロセスを定義する方法を示す．操作対象の制約を満たす，操作前後の操作対象の例を図 2 に示す．A, B 間の Association と is-a 関係をもつ Association を SubA, SubB 間に生成する．

4.3.1　操作間の順序の整理

制約と操作前後の操作対象から，必要となる操作を整理した．制約”別の関連を特化する関連は，他方の関連と同数の関連端をもつ”から，関連端 A, B に対応する SubA, SubB と，Association を指定する操作が必要である．

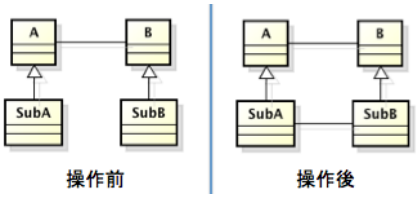

図 2　操作前後の操作対象の例

操作間の順序関係を整理した．操作を実行するために必要となる情報について考える．要素の指定操作の実行には，操作対象と触れた位置の情報のみが必要である．これらの情報は他の操作に依存しない．このことから，それぞれの操作は並行に実行可能であると判断した．

4.3.2　タッチ操作の定義

整理した操作の順序をもとに，タッチ操作を定義した．タッチ操作の意図とモデルに対する操作内容を対応付ける．タップ，ダブルタップやロングタッチで触れた位置にモデルの要素が存在する場合，タップ等はモデルの要素の指定という意味をもつと直感的に考えた．２つのモデル要素に対して，一方から他方へのスワイプは，要素間の対応付けという意味をもつと直感的に考えた．

タップやスワイプを駆使し，4.3.1 項で必要と判断した操作を行なうためのマルチタッチ操作の例を図 3 に示す．図 3

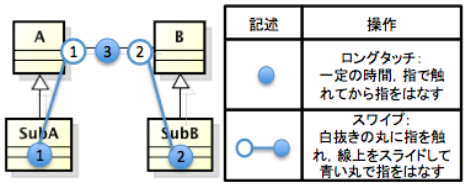

図 3　is-a 関係をもつ Association を生成するタッチ操作

中の番号は，操作を行なう指の識別子である．Association の関連端と対応する型を指定する操作は，Association の関連端と，生成される Association が結ぶ型の対応関係を指定する操作であると考えられる．Association の関連端と対応する型を指定する操作をスワイプと対応付けた．同様に，Association の指定操作をロングタッチと対応付け，マルチタッチ操作を定義した．

4.3.3　タッチ操作プロセスの定義

他の操作と識別可能な操作として，操作プロセスの定義が必要であると考えた．4.3.1 項で整理した操作の順序では，並行に実行可能な操作を順に実行することが許可される．操作を順に実行すると，他の操作プロセスの一部と識別することが困難である．識別が可能な場合でも，すべての操作が完了するまでの間，操作内容を保持する必要があるので，操作の内部処理が複雑化する．操作の順序関係を再整理する必要があると判断した．

定義したマルチタッチ操作をもとに，タッチ操作プロセスの定義を行なった．定義したタッチ操作プロセスを図 4 に示す．タッチ操作を，指をふれる動作と指をはなす動作に分割し，それらの順序を整理することで，並行に実

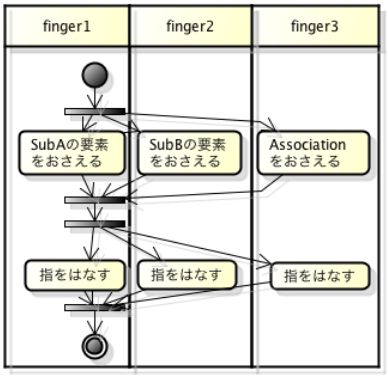

図 4 定義したタッチ操作プロセス

行可能な操作が順に実行されることを防止した．定義した操作前後の操作対象が制約を満たすことを保証する．

定義した操作の前後の操作対象が，操作対象の制約を満たすことを状態遷移機械を用いることで保証する．操作対象が満たす制約を状態と捉え，モデルに対する操作の実行をイベント，モデルに対する操作をアクションとすることで操作対象ごとに状態遷移機械を記述した．タッチ操作プロセスにイベントをステレオタイプとして記述することで，タッチ操作プロセスと遷移を対応付けた．Association の状態遷移機械と操作プロセスの対応の抜粋を図 5 に示す．状態遷移機械で定義した遷移が起こる場合，操作対

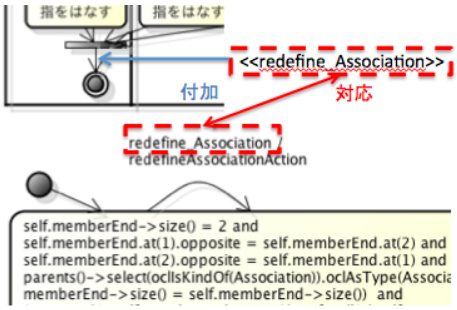

図 5 状態遷移機械の抜粋と操作プロセスとの対応

象は常に操作対象の制約を満たす．

5 操作の設計

制約と操作プロセスの関係を利用した操作の設計例を提示する．異なる型を結ぶ Association と，その Association と is-a 関係をもつ Association を同時に生成する操作を設計する．操作前後のモデルは，図 2 の操作前のモデルから Association を削除した前後のモデルと等しい．この操作では，Association と Property の関係に関する制約と，Association の is-a 関係に関する制約を考慮する必要がある．制約に関連する操作として，Assocaition の生成操作と，is-a 関係をもつ Association 生成操作がある．

設計する操作に関連する操作対象の制約をもとに，操作プロセスを選択する．選択した操作プロセスを図 6 に示す．is-a 関係をもつ Association の生成操作の③は，操作前には存在しない要素を指定するので，実行できない．

選択した操作プロセスのタッチ操作を組み合わせてタッ

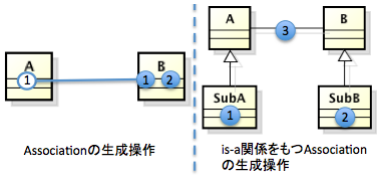

図 6 選択した操作プロセスのタッチ操作

チ操作を設計した．設計したタッチ操作を図 7 に示す．Association の生成操作で生成する Association を，③で

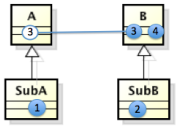

図 7 設計したタッチ操作

指定する Association であると捉えることで，図 6 の操作を組み合わせることができる．

設計したタッチ操作をもとに，選択した操作プロセスを組み合わせた操作プロセスを図 8 に示す．finger3, 4 の

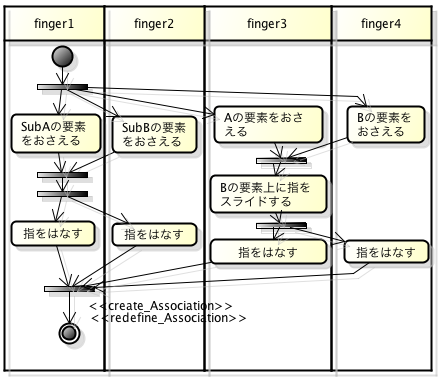

図 8 選択した操作プロセスを統合した操作プロセス

操作が finger1, 2 の実行前に完了した場合，finger3, 4 の操作を Association の生成操作と識別することができない．操作の順序関係を整理し，設計した操作プロセスを図 9 に示す．指をふれる動作と，指をはなす動作の順序関係を整理することで，他の操作と識別可能になった．

6 考察

6.1 設計した操作について

設計した操作について，操作前後の操作対象と制約の関係と，ユーザにとっての直感的な操作の点から，妥当性を確認する．それぞれについて述べる．

操作前後の操作対象と操作対象の制約の関係から，設計した操作について考察した．設計した操作は，制約を考慮して定義した操作プロセスの組み合わせである．タッチ操作プロセスに付加したステレオタイプは，操作対象の状態遷移機械のイベントと対応することから，設計した操作後の対象が制約を満たすことを保証する．このことから，制約を考慮した操作として妥当であると判断した．

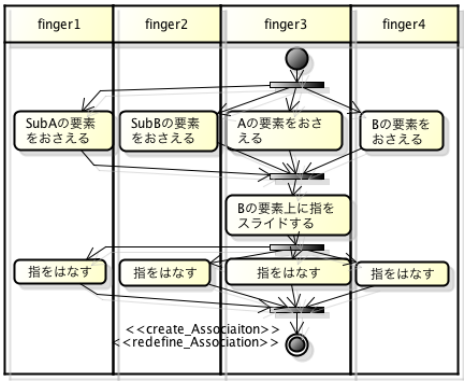

図 9　設計した操作プロセス

設計した操作が，ユーザにとって直感的な操作であるか考察した．ユーザにとって直感的な操作とは，ユーザが好む操作と同じであると考えた．Heydekorn らは，ユーザが好むマルチタッチ操作が，ユーザの知識や経験だけでなく，状況によっても変化しうることを示している．これを，Heydekorn らは状況に応じて，より単純な操作が異なることに起因すると推測している．このことから設計した操作は，必ずしもユーザが望む操作ではないであると考える．しかし，制約と操作の関係の中に，より単純な操作プロセスを定義することができれば，設計する操作を改善できると考えている．単純な操作プロセスを定義する例の一つとして，コンテキストによって，操作を簡略化することを考えた．4.3 節の is-a 関係をもつ Association を生成する操作は，指定する Association が異なる型を結ぶ場合，簡略化可能である．簡略化したタッチ操作は図 6 に示した is-a 関係をもつ Association の生成である．指定された Association の関連端と，①と②によって指定される型の対応が一意に決定するので，タッチ操作の指をスライドさせる動作を排除可能である．

これらのことから，設計した操作は，制約を考慮した直感的な操作として妥当であるが，ユーザにとって，より直感的な操作とするための洗練が必要であると考えた．組み合わせる操作プロセスの定義方法の洗練によって，設計する操作の改善が見込めると考えている．

6.2　異なる体系における操作の設計について

異なる体系にもとづく操作対象を扱う操作の設計に対して，操作対象の制約と操作プロセスの関係と，操作の設計指針を応用できるか考察した．例として，一般に利用される機会が多い GoF デザインパターンを考える．

コンポジットパターンを適用する操作を考える．一般に，パターンを適用する際は，モデルを直接修正するか，パターンの構造を生成し，生成した構造を修正する操作が行なわれる．本研究では，UML 標準で定義されたステレオタイプを利用する．パターンの構成要素であることを表すステレオタイプをモデル要素に対して付加する操作をパターンの適用操作であると考えて，操作を設計する．UML 標準で，ステレオタイプは Association を特化した Extension によってクラスと対応付けられることから，Association の生成操作を応用する．コンポジットパターン適用後のモデルには，Client, Component, Leaf, Composite にあたるクラスが必要であるという制約がある．制約を考慮し，設計したタッチ操作と操作プロセスを図 10，図 11 に示す．4.3.3 項で示した方法で，対象が

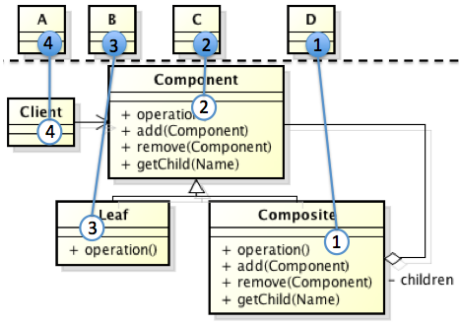

図 10　コンポジットパターンを適用するタッチ操作

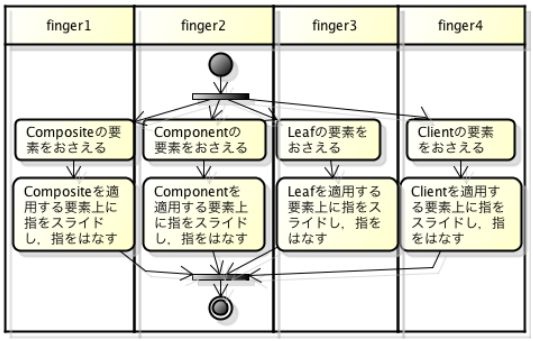

図 11　コンポジットパターンを適用する操作プロセス

制約を満たすことを保証可能であると考える．

7　おわりに

本研究の目的は，図式表現で記述されたソフトウェアモデルの制約と操作の関係の整理方法を提案し，直感的に扱うことが可能な操作の設計を支援することである．整理した操作対象の制約と操作の関係を利用して，操作を設計するガイドラインを示した．

いくつかの操作を設計することで，操作対象の制約を考慮した直感的な操作として，設計した操作が妥当性であることを確認した．今後の課題として，他の図式表現を対象とした事例検証によるガイドラインの妥当性の確認と，操作プロセスの定義方法の改善がある．

参考文献

[1] J. Heydekorn, M. Frisch, and R. Dachselt, “ Evaluating a User-Elicited Gesture Set for Interactive Displays, ” *Mensch & Computer*, pp. 191-200, 2011.

[2] Object Management Group, *OMG Unified Modeling Language(OMG UML) Superstructure*, http://www.omg.org/spec/UML/, 2011.

[3] Object Management Group, *Object Constraint Language*, http://www.omg.org/spec/OCL/, 2010.