

# 機能共鳴分析法の図式表現を用いた視覚的シナリオベース設計支援

日下部 茂  
長崎県立大学  
kusakabe@sun.ac.jp

## 要旨

本稿ではソフトウェアシステムの開発上流工程におけるシナリオ分析に機能共鳴分析法 FRAM の図式表現を用いることを論じる。シナリオは、人やシステムの機能や活動に関する記述であり、開発されるソフトウェアシステムにより何が実行されるのか、どのような効果がもたらされるのかなどを記述できる。自然言語で記述されたシナリオは、ソフトウェアの専門家でなくとも理解や誤りの修正など可能であるが、文字情報の読み込みを前提とする。このようなシナリオの記述や分析を、機能や活動に着目してモデル化と分析を行う FRAM を用いて支援することを提案する。FRAM は図式表現を持っており、視覚的で直観的な理解の促進といったシナリオ分析支援が可能と考えた。本稿ではこのような FRAM の図式表現を用いたシナリオ分析支援の方法と事例について論じる。

## 1. はじめに

本稿では、ソフトウェアシステムの開発上流工程の、特にシナリオ分析に、機能共鳴分析法 (Functional Resonance Analysis Methods, 以降 FRAM と呼ぶ) [1] の図式表現を用いることを論じる。シナリオは、人やシステムの機能や活動に関する記述であり、開発されるソフトウェアシステムにより何が実行されるのか、どのような効果がもたらされるのかなどを記述できる。このようなシナリオの記述や分析を、機能や活動に着目してモデル化と分析を行う FRAM を用いて支援することを提案する。

自然言語で記述されたシナリオは、特別な知識やスキルなどなくともシナリオの理解や誤りの修正など可能で、多様なステークホルダが関わる上流工程での要求の記述や分析にも活用できる。しかしながら、開発側の要求分析者とステークホルダでは、経験も知識も使用する用語なども異なることがあり、自然言語以外の表現により、コミュニケーションを円滑にとりながら要求の獲得やその記述、分析を支援することは重要と考える。FRAM はテキスト表現だけでなく図式表現もあり、文字情報の読み込

みを前提としない、図式表現による視覚的で直観的な理解も含め、シナリオ分析を支援できると考えた。

FRAM では、機能や活動を記述の単位としたモデル記述を行う。後述するように各機能や活動は、入力(I)、前提条件(P)、時間(T)、資源(R)、制御(C)、出力(O)という六つの側面を持ち得る。このような側面を通して機能間のつながりを記述し、機能間の相互作用をモデル化し分析する。このような FRAM は、もともとレジリエンスエンジニアリングのもので、その分析法に厳密で唯一絶対的なものはない[2]。本稿では、FRAM の図式表現を、二次元の半順序グラフと考え、機能や活動の結合関係も意識し、記述ルールや便宜的補助線などを追加する方法を論じる。このような方法によりレジリエンスエンジニアリングだけでなくソフトウェア工学、特にシナリオベース設計[3]に有効な記述や分析が促進されるとの仮説を立てた。

本稿では、そのような仮説の下、記述ルールや便宜的補助線など追加して行う FRAM の図式表現やシナリオの記述、分析の方法について、特にスマートフォン向けアプリを対象に説明を行う。要求工程でのシナリオ記述時に、ユーザおよびスマートフォンの抽象的な機能や活動と、それらの間のつながりを記述した FRAM の図式表現も作成する。その際、機能・活動の実行主体を明示化するといういくつかの記述ルールを追加し、ユーザストーリーマッピング[4]を意識した記述を行う。

ユーザストーリーマッピングでは、ストーリー(ユーザにとっての価値)を付箋紙などに書き出し、ユーザの体験順に時系列的には左から右に、似た機能は上下(基本機能を上、派生的な機能は下)になるように、壁などに配置していく。ここでは付箋紙ではなく、FRAM の記法を用いて、以下のような手順で記述、分析を進める。

1. 自然言語によって記述されたシナリオから FRAM の図式表現を作成する。
2. ユーザ視点のストーリーの中で、開発者が生起を想定しているユーザ行動の明確化と、そのユーザ行動が生起する仕組みの分析のため、FRAM の図式表現と行動分析[5]の ABC 分析と組み合わせる。

3. FRAM の図式表現上で、ソフトウェアの MVP (Minimal Viable Product) 構造と、付加価値を高める追加の開発を意識した表現を行い、工程設計も支援する。
4. 画面のシナリオベース設計のための記述と分析を行う。スマートフォンのアプリケーションでは、画面上のユーザインターフェイス(以降 UI)も重要な役割を持ち Android アプリケーションでは画面単位で Activity クラスを開発する必要がある。このような点をふまえ、画面を介した機能の相互作用の記述と分析も FRAM の図式表現上で行う。

本稿では、著者が所属している組織の大学・学部レベルでのスマートフォン(Android)のアプリケーションの設計演習の事例を使い、提案する手順について説明する。その演習講義の受講生は学部3年生で、オブジェクト指向プログラミングは入門レベル、UML はその概要を知っているが記述したことはほとんどないというレベルの学生が多数派である。例題としては Android アプリケーション開発の入門書[6]の例題の一つ、じゃんけんゲームを用い、提案の手順に加え、演習講義の受講生に対して行ったアンケート結果も併記しながら説明を行う。

本稿の構成は次の通りである。2章でFRAMについて説明する。3章で上記の手順1であるFRAMとそのシナリオ分析への適用、4章で手順2である行動分析とそのシナリオ分析への適用、5章で上記手順3と4について、それぞれの手順とアンケートによる評価結果について述べる。最後にまとめと今後の課題を述べる。

## 2. FRAM

### 2.1. FRAM の概要

FRAM は複数の組織や人、機器といった構成要素が相互作用を行うシステムを対象にした、レジリエンスエンジニアリングの手法として提案されている。従来の安全性と異なる新しい安全性や、回復・復元力について広義のエンジニアリングを行う。従来のように、安全性を「物事が悪い方向へ向かわない状態」としてアプローチするものは Safety-I、「物事が正しい方向へと向かうことを保証する」と考えてアプローチするものを Safety-II として区別し、Safety-II の考えに基づくモデリング手法の一つとして提唱されている。以下の四つの原則に基づいて、ものごとが悪い方向に向かうことより、正しい方向に向かうことに焦点を当てて分析を行う。

- ① 成功と失敗の等価性の原則、

- ② 近似的な調整の原則、
- ③ 創発の原則、
- ④ 機能共鳴の原則

FRAM では、機能について以下に示す六つの側面によってモデル化を行う。

- 入力(I):機能が処理の対象とし、変化するもの。あるいは機能を開始するもの。
- 前提条件(P):機能が実行される前に存在・成立すべき条件。
- 時間(T):機能に影響を与える時間的制約(開始時刻、終了時刻、継続時間など)
- 資源(R):機能が実行されるときに必要とされるもの、あるいは出力を提供するために消費されるもの。
- 制御(C):機能がどのようにモニターされ、あるいは制御されるか。
- 出力(O):機能の結果、何らかの実体か状態変化。

FRAM の図式表現では、図 1のように、各機能を六角形で示す。図式表現でのモデリングでは、典型的には FMV (FRAM Model Visualizer) のようなツールを用いてモデルを記述し、視覚化する[7]。

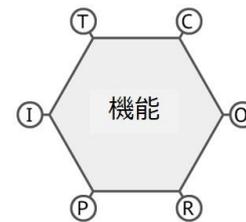


図 1 機能の図式表現

### 2.2. FRAM の分析

FRAM の分析には、複数機能間の相互作用に着目した、システム特性の分析が含まれる。機能の共鳴の観点から、複数の機能が相互に作用した結果、外乱に柔軟に対応して回復する場合もある一方、逆に変動の影響が増幅し、安全を脅かす場合もあるといった分析も行う。

FRAM では機能に着目したモデリングを行い、機能の目的、行っている処理、存在する入出力といったものを明らかにし、機能に名前を付け定義する。定義を進めるにあたり、各側面の要素と名前、相手の機能の名前も明らかにする。しかしながら、モデルの構築や分析の具体的な方法に唯一絶対のものはない。例えばモデルのテキスト表現を重要視したものもある一方[1]、図式表現でのネットワークポロジリーに着目した分析法もあり[8]、文

献[8]では以下のような分析の観点で紹介されている。

- 機能が受け取るもの
- 条件が変わった場合の適応
- 正常ではない条件への反応
- 正常ではない条件の発生頻度
- リソース供給は安定的か, 不安定要因は何か
- 外部要因はどのくらい安定か, 不安定要因は何か
- 前提条件の取りこぼしはないか
- 時間制約にかかるプレッシャーの場所
- 特別なスキル, 機能, 高信頼性の必要箇所
- 最適な実行方法はあるか

上記のような観点をふまえて機能の特徴を十分に引き出した後, シナリオの成功要因やリスク要因の分析を行う。今回は, ユーザ行動の可変性や調整を前提に, 想定する状況下での資源や制約を明確にし, 行動分析の手法を併用してユーザの行動の結果をより予測しやすくする。これにより, 期待しているユーザ行動が本当に生じるかどうかの分析を促進しようとするものである。

### 2.3. 要求工程への FRAM の導入

要求の記述や分析ではステークホルダの共通理解が重要とされており, FRAM の図式表現がそのような共通理解に有効と考える。例えば, ユーザストーリーを記述では, 特定タイプのユーザがやりたいと思うことを表す短い動詞句を書くカードを用いたりするが, そのような短い動詞句を FRAM での機能と対応づけ可能と考える。ストーリーの起点や終点, その間のナラティブフローのつながりといったものも FRAM の記述で明確化できると考えた。

例として, Android アプリケーション開発の入門書[6]の例題の一つ, じゃんけんゲームを用いて説明を行う。ユーザがじゃんけんの手を決め, システムがじゃんけんの勝敗を表示するという流れを, 二つの活動・機能からなる FRAM としたものを図 2 に示す。ユーザが手を決める機能の出力 O とじゃんけんの勝敗を表示する機能の入力 I が, 側面「ユーザの手」でつながっている。

上記のように FRAM を用いることにより, ユーザストーリー

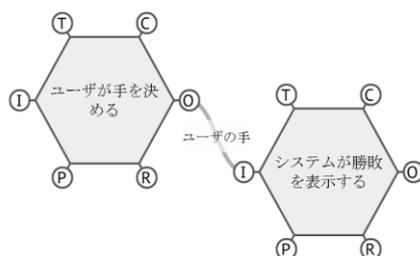


図 2 じゃんけんアプリケーションの初期 FRAM 図

ーにおいて, ユーザの行動とシステムの動作が連携して起点から終点までつながっていること, またそれがどのように成功となるかといった確認が適切な抽象レベルで行いやすくなると考えた。しかしながら, FRAM はシナリオベース設計のためのものでも, ユーザストーリーマッピングのためのものでもないため, オリジナルのままではそのような記述や分析はできない。そのために, 追加的なルールや補助線の導入を検討する。

手順 1 ではユーザ側とシステム側の区別を明確化するため, 機能の配色や並べ方のルール, 境界を示す補助線を導入する。ユーザの行動をユーザ側の機能とし, システムの動作をシステム側の機能として, 色で区別したり, 横に揃えたりする。また, ユーザ側とシステム側の境界を示す補助線の追加も行った上で, ユーザストーリーのシナリオが完結するか, 相互作用がインターフェースを介して適切につながっているかといったことを確認する。

手順 2 では, ユーザストーリーに沿ったシナリオの分析に, 行動分析の ABC 分析を取り入れる。行動(Behavior)そのものだけでなく, 行動のきっかけを与える先行刺激(Antecedent stimulus)と行動した結果与えられる後続刺激(Consequent stimulus)や, 確立操作も分析する。これにより, 対象ユーザ行動が実際に引き起こされるかについて, 先行刺激や後続刺激などの存在やつながりなど, 行動分析の知見に基づく分析が出来るかと考えた。そのため, A,B,C に相当する部分を独自の記法でマークする。

手順 3 では, アジャイル型の開発のように, 小さな単位で開発を繰り返すことを念頭に FRAM の図式表現を作成する。MVP から始まり, 開発が終了した機能から随時提供可能となるような開発工程の図式表現を作成する。

手順 4 では, 画面やクラスのシナリオベース設計に向けた記述と分析を行う。スマートフォンのアプリケーションでは画面上のユーザインターフェイスは重要で, Android アプリケーションでは画面ごとに Activity クラスを開発する。これを念頭に画面の切り替えを示す補助線を追加する。その上で, 画面を介した円滑な相互作用のため, ユーザが視認し操作する画面上に何がどうあるべきかといった分析を行う。画面の仕様を考える際に, 先行刺激, 行動, 後続刺激や確立操作が画面要素とどう対応し実現されるか, といった分析も FRAM 上で行う。

### 3. シナリオ記述

まず人やシステムの機能・活動に関するシナリオを自然言語で記述する。ソフトウェアシステムにより何が実行されるのか, どのような効果もたらされるのかなどを記

述する。前述のじゃんけんの例題について、まず、以下図 3 のようなユースケースを考えシナリオを記述する。

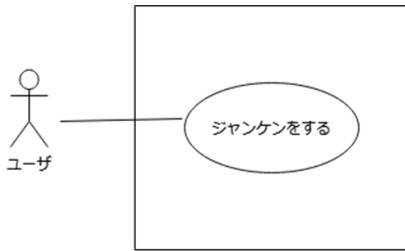


図 3 じゃんけんアプリのユースケース

自然言語のシナリオ記述では以下の下線部のように行動主体を明記する。

レベル:海面レベル

主成功シナリオ(MSS):

1. ユーザーがアプリを起動
2. システムが画面を表示
3. ユーザーが手を決める
4. システムが手を決める
5. システムが勝敗を決定

このようなシナリオのユーザーストーリーの起点や終点の機能、その間の機能のつながりといったものを FRAM で記述する際に以下のようなルールを用いる。(図 1 図 4 参照)

- 動作主体ごとに、機能の色を揃え、水平方向に揃えて並べる。図中、ユーザーが動作主体のものが青、システムが動作主体のものが緑である。
- FRAM 用のツール、FMV で作成した FRAM の図に、別途、シナリオ記述での海面、ユーザーとシステムの境界に補助線を引く。図では中央の破線の横線が補助線である。

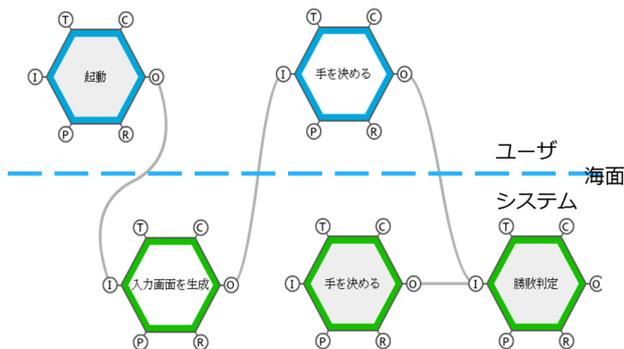


図 4 じゃんけんアプリの FRAM 図の例

その FRAM の図式表現を図 4 に示す。上述の自然言語シナリオの 1 と 3 に対応する「起動する」と「手を決める」がユーザー側なので、それらの横並びの配置と色を揃えている。上述の自然言語シナリオの 2, 4, 5 に対応するものがシステム側なので、それらの横並びの配置と色を揃えている。さらに、ユーザー側とシステム側との間に補助線を引きその境界の存在を表現している。

## 4. 行動分析

さらに FRAM のモデリングと ABC 分析を組み合わせることにより行動分析の知見の活用を試みる。前述のような FRAM によるモデリングで、ユーザーストーリーの起点や終点、その間の機能間のつながりといったものを行動分析学の知見を用いて分析する。行動分析の知見の活用については、一般向けの書籍[9]も出版されており、行動分析の非専門家でもその知見は活用可能と考える。

### 4.1. ABC 分析

有機体の行動には、刺激によって反射的に誘発されるレスポナント行動と、そのような誘発刺激によらずに自発されるオペラント行動がある。このような行動のそれぞれのタイプに対し、その変容の二つの原型として、不随意的な行動の変容過程(条件反射)であるレスポナント条件付けと、随意的な行動の変容過程であるオペラント条件づけがある[5]。

ここではユーザーの随意的なオペラント行動を対象とした分析を行う。オペラント条件付けは、有機体が環境などに働きかけるために能動的に自発する行動であるオペラント行動の条件付けであり、あるオペラント行動に刺激を随伴させるとそのオペラント行動の頻度が増加するとき、その行動は強化されたといい、そのような操作をオペラント強化という。そのような操作でオペラント行動の頻度が増加することをオペラント条件付けという。オペラント条件付けがなされたオペラント行動は、自発されたときに強化されないと頻度が減少する。このような操作と事実をオペラント消去と呼ぶ。ユーザーストーリーで、ユーザーの苦痛や喜び、報酬といったものも考慮する際に、行動が強化されるか消去されるかといった観点を導入する。

提示されることで、あるオペラント行動の自発が強化される強化刺激を正の強化刺激や好子と呼ぶ。除去されることであるオペラント行動の自発が強化される刺激を負の強化刺激や嫌子と呼ぶ。好子の出現や嫌子の消失は直前のオペラント行動の自発頻度を増大させる。

上述のような随意的なオペラント行動の仕組みに着目

して分析を行うにあたって、行動分析学の三項随伴性の分析を用いる。三項随伴性は、きっかけとなる先行条件・弁別刺激(Antecedent stimulus)の下に生じたオペラント行動(Behavior)の後に引き続く強化刺激(Consequent stimulus)が随伴すると、その行動が増えるという関係を示したものである(図 5 参照)。先行条件・弁別刺激、行動、強化刺激の英語表記の頭文字をとってこの分析を ABC 分析と呼ぶ。

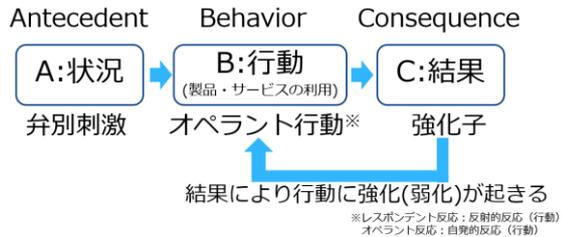


図 5 三項随伴性

行動分析では前述のような行動の随伴性を考えるときは慣習的に随伴性ダイアグラムと呼ばれるものを作成し、行動の直前直後を記述して分析する。ダイアグラム記述により行動の随伴性による影響を整理し、行動の強化・弱体化について言葉だけで記述するより分かりやすく捉えることが可能とされている。しかしながら、記法の統一された定義やツールなどはない[10][11]。ここでは、随伴性の分析を、エンジニアリングの観点で効果的に実施することを目指し、FRAM のモデリングと関連付けた随伴性分析を行う。ユーザの行動に加え、弁別刺激や好きや嫌いの機能と関連づけて FRAM のモデリングを行う。機能間のつながりを、三項随伴性の刺激と反応の連鎖の観点で、行動分析学の知見と照らし合わせた分析を行う。

#### 4.2. FRAM 図式での ABC 分析

前述の例題のシナリオに対して、ABC 分析を行った例を以下に示す。全体的にも、部分的にも、の連鎖も行動と結果(強化刺激)の連鎖があることを確認する。また、

その連鎖に入る弁別刺激や確立操作があること、それらの連鎖がこれらの連鎖が強化で終わっていることを確認する。前述の自然言語でのシナリオは以下のように書き換えられる。下線付き太字の部分追加された部分になる。ユーザ視点では、ユーザが手を決めるという動作の結果がなかったため、ユーザが結果を見るという動作と、そのためのシステムによる結果表示が加えられている。

レベル:海面レベル

主成功シナリオ(MSS):

1. ユーザがアプリを起動
2. システムが画面を表示
3. ユーザが手を決める
4. システムが手を決める
5. システムが勝敗を決定
6. システムが画面を表示
7. ユーザが結果を見る(再度じゃんけんなら 2 へ)

図 6 にこのような分析の試みを、FRAM の図式表現上で行った例を示す。じゃんけんゲームを起動してプレイするのが標的行動 B だとして、その先行刺激 A や結果の提示 C が何になるかを FRAM 図上で分析するため、図の上側に ABC の範囲に相当する楕円を記入している。この表記は、FRAM のツール FMV には無い機能なので、図式表現をエクスポートして別ツールで書き加えた。そのような全体的な ABC 連鎖を詳細化したレベルの各部分でも、ABC の連鎖がつながっている必要がある。その分析のため B1 がゲームを起動するユーザ行動、B2 がじゃんけんの手を決めるユーザ行動として詳細化している。

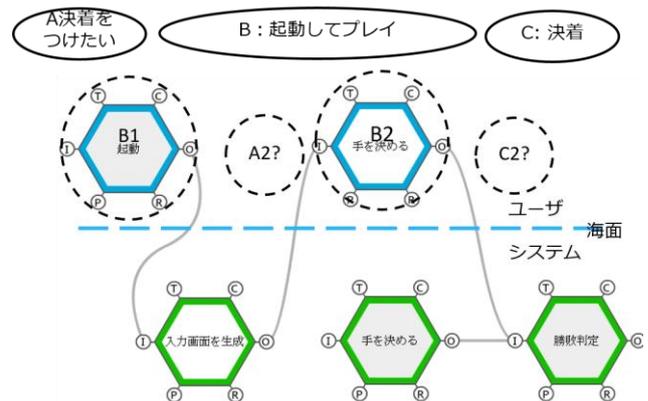


図 6 FRAM 図上での ABC 分析の試み

FRAM 図上で、ABC 分析を進め、行動だけでなく先行刺激や結果の提示を明確化したものが図 7 になる。FRAM 図全体で中心となる標的行動 B は「起動してプレイ」であり、その先行刺激 A と結果 C と並べて、実線の楕円で図示している。そのような全体的な ABC 連鎖を分解したレベルでも、ABC の連鎖がつながっている必要があり、図中、点線で囲った A1, B1, C1, A2, B2, C2 として記入している。図中点線の楕円で囲まれた「C1/A2」は、B1 の結果である画面表示 C1 が、B2 に対する先行刺激 A2 となっていることを示している。このように上位レベルの ABC を実現する、下位レベルの ABC の連鎖を分析している。また自然言語のシナリオ 7 に対応する再じゃんけんに対応する要素も記述した。

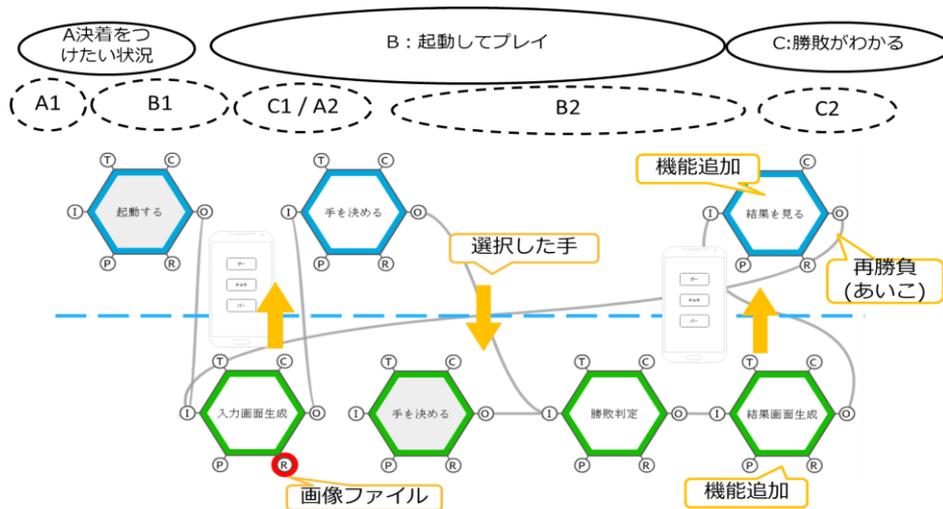


図 7 FRAM 上での ABC 分析の結果

#### 4.3. アンケートによる ABC 分析支援の評価

上記のような ABC 分析をシナリオの自然言語記述に対して行うのと、FRAM の図式表現に対して行うのと、どちらが良いか問うアンケートを、演習の受講学生に対して Google Form を用いて行った。その結果を図 8 に示す。FRAM の図式表現の方が良いとの回答が多いが、そうでない回答も約三分の一ある。



図 8 行動分析を行いやすいシナリオ表現の評価

### 5. ソフトウェア構造の分析

ここでは、以下の二つの観点でのソフトウェア構造の分析につながるシナリオ分析について論じる。

- アジャイル型の開発のように、機能ごとの小単位で開発を繰り返し、MVP から始まり、開発が終了した機能から随時提供可能となるような開発工程を想定し、FRAM の図式表現を作成する。
- 画面やクラス構造の、上位レベル設計につながる記述と分析を行う。画面を介した機能・活動の円滑な相互作用の実現のためには、ユーザが操作する画面やシステム内のソフトウェアの構造がどうあるべきかについての分析を行う。

#### 5.1. MVP と追加機能の段階的リリースの構造

ここでは、最初のリリースでは4.2節で紹介したようなシナリオを実現するものを、二回目のリリースではよりゲーム性を高めるようなものをリリースするという、段階的リリースのための分析について論じる。ゲーム性を高める二回目のリリースでは、システム側のじゃんけんの手の決定にこれまでの勝負の履歴を反映させるシナリオを考える。

##### リリース 2

##### 主成功シナリオ(MSS):

1. ユーザがアプリを起動
2. システムが画面を表示
3. ユーザが手を決める
4. システムが履歴を読み出す
5. システムが履歴を使い手を決める
6. システムが勝敗を決定
7. システムが履歴を保存
8. システムが画面を表示

9. ユーザが結果を見る(再度じゃんけんなら2へ)

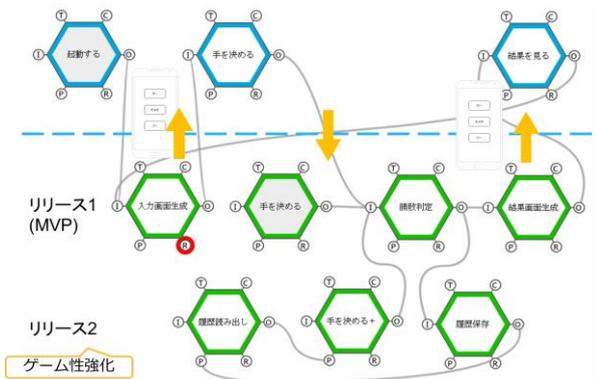
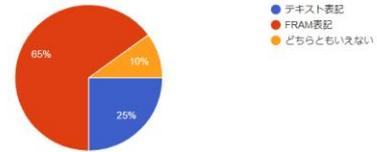


図 9 段階的リリースの FRAM 図式表現

このような段階的リリースに対応する FRAM の図式表現を図 9 に示す。破線より下のシステム側で、機能の連鎖をリリースの回数に対応させた並びにする。今回の例では二回リリースがあるので、一回目のリリースの MVP の機能の並びと、ゲーム性を高めた二回目のリリースで、並びを変える。後続のリリースでは、これまでのリリースの機能を含むことを前提とするが、機能の置き換えなどがある場合はそれがわかるような命名規則を使い、FRAM のテキスト表現に書いておくなどする。

Q5. 段階的開発戦略のシナリオ分析はどちらがやりやすいですか？

20 件の回答



Q6. Q5に関してコメントなどあれば書いてください。

2 件の回答

言葉だけだと今の時点がどこかわからなくなるかもしれないので、図がある方が今どこについて考えているかがわかりやすい。

FRAMが少し複雑になるためテキスト表記の方が変更点がわかりやすい

図 10 段階的リリース構造の分析支援の評価

5.2. 段階的リリース構造支援のアンケート評価

上記のような段階的リリースの分析を、シナリオの自然言語記述に対して行うのと、FRAM の図式表現に対して行うのと、どちらが良いか問うアンケートを、演習受講生に対し Google Form を用いて行った。その結果を図 10 に示す。そうでない回答も約三分の一あり絶対ではないが、FRAM の図式表現の方が良いとの回答が多かった。

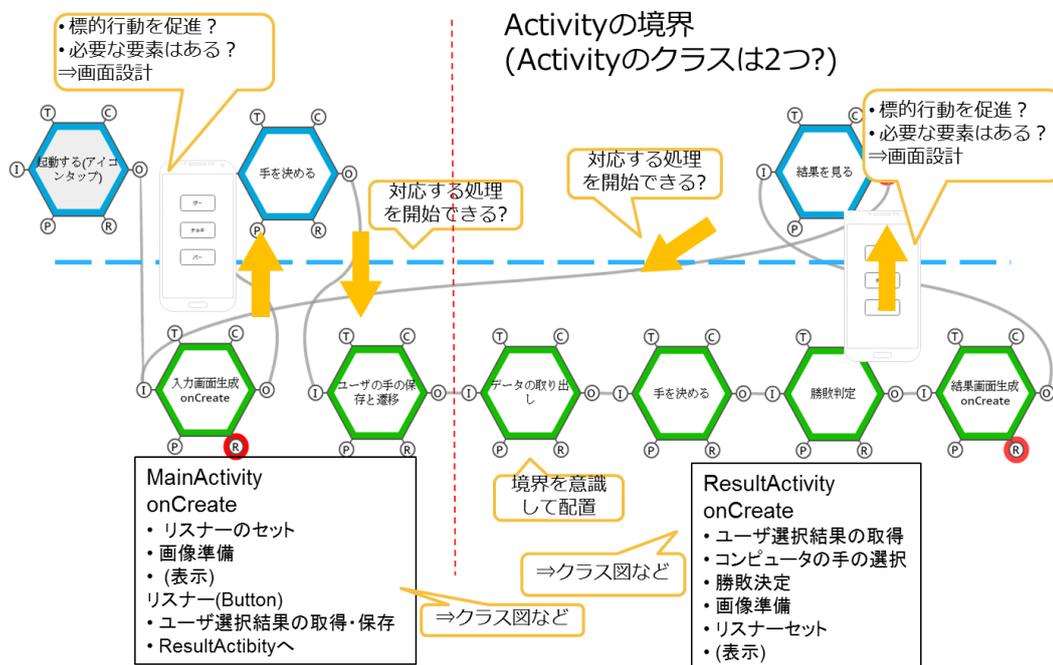


図 11 画面の仕様やクラスの分析着手時の FRAM 図

### 5.3. シナリオベースの画面仕様やクラスの分析

画面仕様やクラス構造などのシナリオベース設計に向けた分析を行う。スマートフォンのアプリケーションでは画面を介した機能・活動の円滑な相互作用を分析することは重要である。Android アプリケーションでは、画面単位で Activity のクラスをつくる必要があるといった点も考慮しながら、ユーザが操作する画面やシステム内のソフトウェアの構造がどうあるべきかについての分析を行う。

図 11に、図 7相当の FRAM の図式表現をベースに、画面仕様や Activity のクラスの分析を行った図を示す。縦の点線が Activity の境で、この線をまたぐデータの移動には、別途明示的な処理が必要になる。また、ユーザ側とシステム側の境界を示す点線をまたぐ相互作用について、下向きのはユーザがシステムを起動するもの、上向きのはユーザに画面を提示して行動を促すものとして考える。このような分析を行うことで、より実装に近いレベルの機能を抽出することが可能になる。

### 5.4. シナリオベースの画面仕様やクラスの分析の評価

上述の、画面仕様やクラスの分析での FRAM の図式表現が支援として役立ったかのアンケート結果を図 12に示す。そうでない回答も約五分の一あり絶対ではないが FRAM の図式表現の方が良いとの回答が多かった。

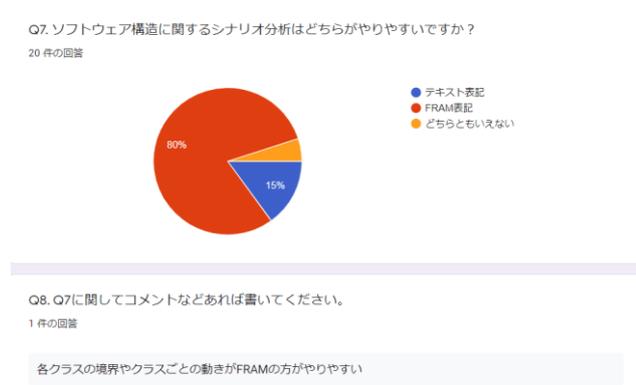


図 12 画面仕様やクラス分析支援の評価

## 6. おわりに

本稿ではソフトウェア、特に UI が重要なスマートフォンのアプリケーション開発上流工程で、シナリオ分析に機能共鳴分析法 FRAM の図式表現を用いることを論じた。FRAM にはテキスト表現だけでなく図式表現があり、文字情報の読み込みを前提としない、視覚的で直観的な理解の促進も含めたシナリオ分析支援が可能と考えた。

提案手順では、まず、シナリオから FRAM の図式表現を作成する。次に、生起を想定するユーザ行動の明確化と、その生起の仕組みの分析のため、行動分析の ABC 分析を併用する。さらに、FRAM の図式表現上で段階的リリースに対応するソフトウェア構造を検討する。スマートフォンのアプリケーションでは、画面上のユーザインターフェイスも重要な役割を持ち、Android アプリケーションでは画面ごとに Activity クラスの開発も行う必要がある。このような点を踏まえ、画面を介した機能の相互作用も FRAM の図式表現で記述、分析することを論じた。

著者が担当のスマートフォン(Android)アプリケーションの設計演習(学部)で、自然言語によるものと、提案した FRAM の図式表現による記述と分析を比べるアンケートを行った。その結果、約三分の二程度の受講者が FRAM の図式表現のものが良いとする回答結果を得た。

モデル構築や分析、シミュレーションなどを支援する新しいツールも出ており[12][13]、ソフトウェア開発も含め FRAM の活用法について引き続き研究を行う予定である。

## 参考文献

- [1] E. Hollnagel (小松原明哲監訳). 社会技術システムの安全分析-FRAMガイドブック, 海文堂出版, 2013
- [2] E. Hollnagel, Personal communication, 2018
- [3] J. M. Carrol, Making Use --- Scenario-Based Design of Human-Computer Interactions, MIT press, 2000
- [4] Jeff Patton (川口恭伸監訳), ユーザーストーリーマッピング, オライリー, 2015
- [5] 佐藤方哉, 行動理論への招待, 大修館書, 1976
- [6] 金田浩明, はじめての Android プログラミング 第4版, SB クリエイティブ, 2019
- [7] FMV FRAM Model Visualizer, <https://functionalresonance.com/FMV/index.html>
- [8] 野本秀樹, 道浦康貴, 石濱直樹, 片平真史, FRAM-機能共鳴分析手法による成功学に基づく安全工学, SEC journal, 14(1), pp.42-49, Aug. 2018
- [9] 島宗理, 使える行動分析学, ちくま新書, 2014
- [10] M. A. Mattaini, Contingency Diagrams as Teaching Tools, The Behavior Analyst, 18(1), pp.93-98, 1995
- [11] S. Toogood, Using contingency diagrams in the functional assessment of challenging behavior, Int'l J. of Positive Behavioural Support, 2-1, pp.3(10), 2012
- [12] R. Patriarca, G. D. Gravio, F. Costantino, myFRAM : an open tool to support FRAM. 12th FRAMily meeting and workshop, pp.11-13, 2018.
- [13] FRAM Model Interpreter, <https://functionalresonance.com/the-fram-model-interpreter.html>