

# COSMIC-FFP の拡張とその評価

長野 伸一, 西山 茂  
NTT 東日本 研究開発センタ

## 概要

COSMIC-FFP はソフトウェア機能規模測定手法のデファクトスタンダードとして知られているファンクションポイント法をリアルタイムシステムにも適用可能とした手法である。COSMIC-FFP では実現する機能と、プロセスを起動するイベントで表現することにより、リアルタイムシステムにも適用可能としている。我々は、この COSMIC-FFP の対象領域を拡大する拡張した定義を提案する。

本論文で、現行の COSMIC-FFP の定義でプロセスを起動するイベントが複数ある場合は測定困難であることを明らかにする。その後で計測できるのは単一のイベントのみで、複数のイベントを組み合わせた場合は測定困難であることを明らかにし、その問題を分析することから、本問題を解決するように拡張した定義を示す。さらに、エレベータの例題を通して、提案した定義で上手く測定できることを示す。

## Extension and its evaluation of COSMIC-FFP

Shin'ichi Nagano, Shigeru Nishiyama  
NTT-east Research and Development Center

## Abstract

In COSMIC-FFP for measuring the functional size of software, we propose extending a definition so that we can apply it for wider targets. COSMIC-FFP is one of function point methods and can be applied for also real-time system. The functional process concept, which expresses the requiring function, makes it possible to apply for. However, the concept of the functional process in COSMIC-FFP assumes the single event, and doesn't take into consideration the case where multiple events are combined. In this paper, it is only the single event that can be applied by current COSMIC-FFP, but it is difficult to be applied for functions initiated by multiple events with COSMIC-FFP.

We'll propose the extended definition of COSMIC-FFP, and show that we can measure functions initiated by multiple events.

Furthermore, we show that we can measure functions initiated by multiple events through measuring the sample of an elevator.

## 1. はじめに

COSMIC-FFP は、ファンクションポイント法を拡張した機能規模測定手法である。従来のファンクションポイント法に比べて、入出力以外の要件の比重が高いリアルタイムシステムにも適用が可能である。

COSMIC-FFP の適用領域は大きく分類すると、下記ようになる。多量のデータを管理する、いわゆる MIS タイプのソフトウェアや、実世界で起こるイベントを遅滞無く取り込み、制御することを目的としたリアルタイムシステムソフトウェア、左記の組み合わ

せのソフトウェアである。ところが、複雑な計算アルゴリズムや特殊で複雑な規則を特徴とするソフトウェアや連続的に変数処理するソフトウェアには不向きである。

例えば、MIS 系のソフトウェアとして、業務管理アプリケーション、リアルタイムシステムとしては、交換機・家電製品エレベータなどがある。逆に、適用範囲外のものとしては、エキスパートシステムやシミュレーションソフトウェア、音声・映像関連のソフトウェアなどがあげられる。 [2]

COSMIC-FFP の特徴の一つは、機能プロセスの集合によりソフトウェアを捉えてい

ることである。機能プロセスとは、ユーザが要求する機能をデータの移動であるサブプロセスの半順序集合で表したものであり、ユーザに識別可能なイベントが引き金となり起動する。COSMIC-FFP がリアルタイムシステムにも適用可能であるのは、機能プロセスによるソフトウェアの表現がリアルタイムシステムの機能要件を表現できるからである。

しかしながら、現行のCOSMIC-FFPの機能プロセスでは、引き金となるイベントが単一のもののみを対象としており、複数のイベントの組み合わせを引き金とする場合は考慮していない。リアルタイムシステムの場合、複数のイベントの組み合わせによる機能の起動は様々ある。そこで、我々は複数のイベントの組み合わせが引き金となる場合へのCOSMIC-FFPの適用を検討した。

複数のイベントを組み合わせた条件に起因した機能も計測を可能とするように、COSMIC-FFPの定義を拡張することを提案する。

本論文で、複数のイベントを組み合わせた条件に起因した機能が、従来のCOSMIC-FFPでは計測できないことを示し、計測可能となるように定義の拡張を提案する。更に、拡張した定義に対してエレベータを例題にとり計測可能であることを示す。

## 2. COSMIC-FFP 概要

以下に現行のCOSMIC-FFPの概要を説明する。

### 2.1. COSMIC-FFP の考え方

COSMIC-FFPは、利用者がソフトウェアに要求している機能の規模を計測する手法である。LOC<sup>1</sup>のように製造したソフトウェアを基に大きさを示すものではない。ソフトウェアとして実現を要求する機能の大きさを定量化するものである。

COSMIC-FFPで想定しているソフトウェアモデルを図1に示す。利用者が要求する機能を利用者機能要件(FUR: Function User Requirement)と呼び、ソフトウェアにより実現する。利用者機能要件は、機能プロセス(Functional Process)の集合により実現される。さらに、各々の機能プロセスは、データ移動を遂行するサブプロセス(Sub-process)

の集合であると表現している。[2]

COSMIC-FFPによる機能規模は、サブプロセスの数を集計した結果で表し、一つのサブプロセスを最小単位とし、その単位をCFSU(Cosmic Function Size Unit)と定義している。

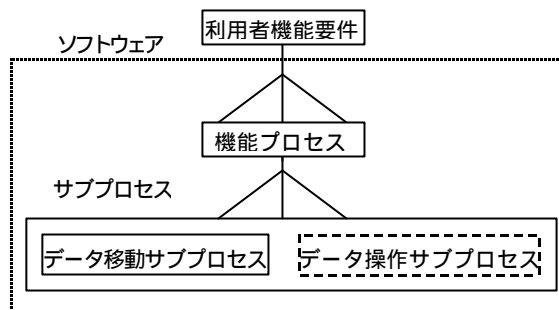


図1 COSMIC-FFPのソフトウェアモデル

### 2.2. COSMIC-FFP の計測モデル

COSMIC-FFPの計測手順に沿って、定義している構成要素を説明する。

まず、測定にあたって、その対象を明確にする。COSMIC-FFPでは、測定対象ソフトウェアとそれ以外の間に境界(boundary)を測定の目的に応じて、測定者が設定する。

次に、測定する機能の識別を行う。識別する機能を漏れも重複も無く、一意に識別するために、イベント毎に識別を行う。

要求した機能には、要求者の意図を反映すべく、当該機能の起動契機、機能内容と機能した結果がある。COSMIC-FFPでは、機能内容や機能した結果を、機能プロセスとして表現する。

また、この起動契機をトリガイイベント(Trigger Event)と呼ぶ。トリガイイベントは、境界の外で発生し、利用者機能要件を実現する機能プロセスの起動契機となる。

最後に、機能プロセスの構造を識別する。機能プロセスは、サブプロセスの集合である。

一つのサブプロセスは、一つのデータ移動である。COSMIC-FFPでは、サブプロセスは、ふたつの観点から、4種類のタイプに分類している。一つ目の観点は、データ移動が境界を跨ぐか否かである。もうひとつは、移動方向であり、測定対象のソフトウェアに取り込むか、送り出すかである。境界を跨ぎ、境界内にデータが入ってくるタイプをエントリ(Entry)と呼び、Eと表現する。逆に、境界内から境界外へデータが移動するタイプをイクジット(eXit)と呼び、Xと表現する。境界内でデータを測定対象ソフトウェアに

<sup>1</sup> LOC:Lines of code

取り込むサブプロセスをリード(Read)と呼び、Rで表現する。逆に、境界内でデータを測定対象ソフトウェアから送り出すサブプロセスをライト(Write)と呼び、Wで表現する。E,X,R,Wの総数をCOSMIC-FFPの機能規模とする。

### 3. 問題の所在と定義の分析

本章では、複数のイベントを組み合わせた条件に起因する機能を示し、従来のCOSMIC-FFPでは計測困難であることを述べる。

#### 3.1. 機能プロセスを起動する複数イベント

複数イベントの組み合わせで起動する機能を下記に示す。

(A) 2個以上のイベントが全て成立することにより、当該機能が生起する。

(B) 各々のイベントは、ユーザが認識できるものである。

(C) 各々のイベントは互いに独立であり、順序関係はない。

#### 3.2. 複数のイベントを組み合わせた条件により起動する機能の計測

上記に定義した機能に対して、従来のCOSMIC-FFPを適用すると、以下の4つのパターンが考えられるが、定義に違反するか、定義からは判断できない。

And型

独立型

分離型

Or型

以下に、各パターンについての問題点を述べる。

##### 3.2.1. And型と問題点

3.1の条件を素直に図式したものが図2である。図2では、条件となる全てのイベントを待ち合わせ、後続の一連のサブプロセスを実行していく。

これを以後、And型と呼ぶ。

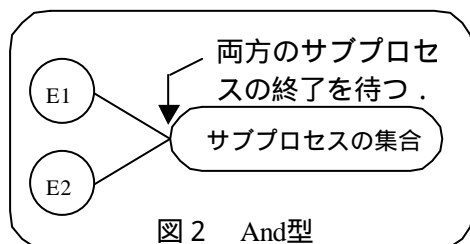


図2 And型

And型をCOSMIC-FFPのマニュアルに照らし合せると、下記の2つのマニュアル違反となる。一つは、「Subsets of triggering

events are not considered different triggering events. (トリガイベントの一部は、異なったトリガイベントとは見なさない。)」(規則 - 機能プロセス)への抵触である。この規則では、複数のトリガイベントの組み合わせによる機能プロセスの起動を制限している。もう一つは、「A functional process contains no more than self-induced wait state(which may occur when it is completed)(機能プロセスは、二つ以上の待ち状態を含まない。)」(原則 - 機能プロセス)への抵触である。機能が許容している待ち状態は、機能プロセスの最後のみであり、機能プロセスの途中で、待ち状態をとることはできない。しかし、And型は図2でE1とE2のサブプロセスの次で待ち状態にあると見なすことができる。

以上から、And型を正しいCOSMIC-FFPの機能プロセスとして認めることができない。

##### 3.2.2. 独立型と問題点

And型の問題点は、複数のトリガイベントを、待ち合わせたことに起因する。そこで、複数のトリガイベントを個別に扱い、トリガイベント毎に機能プロセスを識別することとする。その結果を図3に示す。

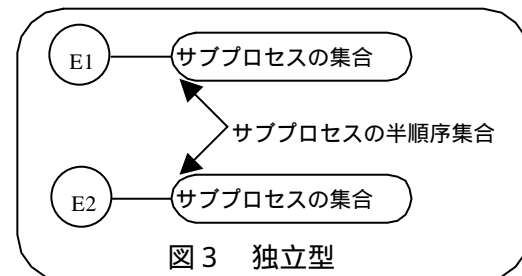


図3 独立型

これを以後、独立型と呼ぶ。

独立型は、And型の欠点を補っているが、同じ機能を重複して抽出しており、許容できない。従って、独立型をCOSMIC-FFPの機能プロセスとして許容できない。

##### 3.2.3. 分離型と問題点

独立型の欠点は、同じ機能を重複して計数してしまうことであった。そこで、複数のトリガイベントは個別に扱いながら、共通して同じ機能の部分をまとめて一つに扱うように機能プロセスを識別する。その結果を図4に示す。

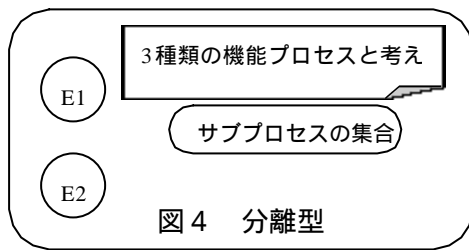


図4 分離型

これを以後、分離型と呼ぶ。分離型は、トリガイベントによる契機の部分と契機によって行う処理の部分とを分離した形態となっている。

分離型は、COSMIC-FFP マニュアルの「A functional process contain at least two data movements, an entry and an exit or a write.(機能プロセスは、少なくともふたつ(エントリとイクジット、場合によってはエントリとライト)のデータ移動を含む。)」(原則 - 機能プロセス)に違反する。トリガイベントによる契機の部分は、エントリのみで、イクジットやライトが無く、処理の部分は、イクジットやライトはあるもののエントリが存在しない。

従って、分離型も COSMIC-FFP の機能プロセスには成り得ない。

### 3.2.4. Or 型と問題点

以上の3パターンの欠点を除くため複数のトリガイベントを待ち合わせず、トリガイベントの生起を記録し、この記録と照合することとする。これを図5に示す。図5を以後Or型と呼ぶ。

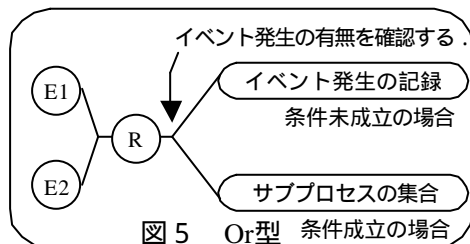


図5 Or型 条件成立の場合

Or型は、待ち合わせを取らないことがAnd型との相違点である。個々のトリガイベントが発生すると待ち合わせを行わず、もう一つのトリガイベントは発生済みであるか否かを確認する。発生済みの場合、条件が成立したと判断し、サブプロセスを進めて処理を実行する。もう一つのトリガイベントが未発生の場合には、当該トリガイベントが発生した記録を残し、処理を終える。このとき、トリガイベントの発生有無を記録に残すために、処理用のデータグループを追加する。Or型

は、待ち合わせを解消するために、トリガイベントの発生状況を記録するデータを追加し、各々のトリガイベントの発生を個別に扱うようにしたパターンである。

Or型には、下記の問題がある。

その問題とは、処理用に追加したデータである。COSMIC-FFPの測定モデルによるデータは、利用者機能要件から導き出すことができるデータに限っている。計測上の都合からデータを追加することはマニュアルに違反している。

従って、Or型もCOSMIC-FFPの正しい機能プロセスには成り得ない。

### 3.3.識別不能原因

前述で、上記に定義した問題が、現行のCOSMIC-FFPで計測不能であることを示した。ここでは、その原因を分析する。上記の計測不能理由は、定義の違反と定義から当該機能プロセスが適合しているか否かが判断できないことである。

定義が不十分であることに起因する。下記に、定義が不十分な理由を示す。

(1) 複数のトリガイベントの組み合わせが、機能の起動契機となる場合の扱いが示されていない。

COSMIC-FFPのマニュアルでは、複数のトリガイベントに関わる記述は、機能プロセスの規則a)のみである。当該規則は、同じデータに対して連続するトリガイベントに関わる記述である。複数のトリガイベントの組み合わせで起動する機能プロセスに条件を与えるものではない。従って、現行のCOSMIC-FFPの定義で、複数イベントの問題を解決することはできない。

(2) 複数のトリガイベントの待ち合わせは、COSMIC-FFPの待ち状態に当たるか、判断できない。

互いに独立なイベントが同期を取るためには、待ち合わせが必要である。この待ち合わせは、COSMIC-FFPで定義している待ち状態と同じであるか明記されていない。

COSMIC-FFPでは、待ち状態を非同期タイミング点と表現し、データ移動が順次続く中で、あるデータ移動が、そのひとつ前のデータ移動と同期を取らなくなった時点と説明されている。しかしながら、現行のCOSMIC-FFPでは、データ移動の契機が定義されていないことから、データ移動の同期を識別することはできない。

(3) 機能プロセス中のサブプロセスを構成する条件が明示されていない。

サブプロセスが、ある機能プロセスに属するか否か判断する条件があいまいである。

現行の COSMIC-FFP のマニュアルでは、機能プロセスを、下記のように定義している。

#### 定義 - 機能プロセス

A functional process is a unique set of data movements (entry, exit, read, write) implementing a cohesive and logically indivisible set of Functional User Requirements. It is triggered directly, or indirectly via an 'actor', by an Event(-type) and is complete when it has executed all that it is required to be done in response to the triggering Event(-type).

機能プロセスは、一意で順序だったデータ移動（エン트리、エグジット、リードおよびライト）の集まりであり、緊密な関係にある一連の利用者機能要件を実現するものである。機能プロセスは、ひとつのイベントにより起動するが、いったん実行したら、トリガイベントに関してソフトウェアを首尾一貫した状態に保たなければならない。

「logically indivisible(緊密な関係にある)」や「implementing a cohesive(ソフトウェアを首尾一貫した状態に...)」の定性化/定量化困難な表現が定義中にあるため、客観的な判断を行うのが困難である。また、機能プロセスは、サブプロセスの集合であると定義されているが、機能プロセスの中で先頭サブプロセスの条件、最後尾のサブプロセスの条件、そして中間のサブプロセスの条件が不明である。従って、機能プロセスの形態に関する疑問が生じてても、定義に照らし合わせて、疑問を解決することができない。

## 4. 定義の拡張

3章で述べた問題点を解決する COSMIC-FFP の測定モデルの拡張を提案する。

### 4.1. 定義の拡張の方針

定義の拡張では、(1)機能プロセスの構成条件、(2)複数のトリガイベントの扱い、(3)待ち状態の3つを明確にする。その基本方針を以下に示す。

#### (1) 機能プロセスの構成条件の明確化

機能プロセスの起動条件と機能プロセス

中で順次続くサブプロセスの継続の条件、最後のサブプロセスの条件を明らかにする。

#### (2) 複数トリガイベントの取扱いの明確化

複数のトリガイベントを許容するか否か、許容する場合、その条件を明らかにする。

#### (3) 待ち状態の明確化

待ち状態に関する明確な定義を行う。

## 4.2. 具体的な規則の拡張

定義1：サブプロセスは、データとその移動に加え、移動する契機を与えるキックから成る。(図6参照)

定義2：トリガイベントはキックである。

定義3：サブプロセスは他のサブプロセスのキックとなることができる。(図7参照)

定義4：機能プロセスは、サブプロセスが他のサブプロセスのキックとなる半順序集合である。

定義5：ひとつのサブプロセスに対して、複数のキックがある状態を待ち状態とする。

定義6：待ち状態が許容されるのは、「トリガイベント」と「トリガイベントに固有なサブプロセス」の組み合わせのみである。

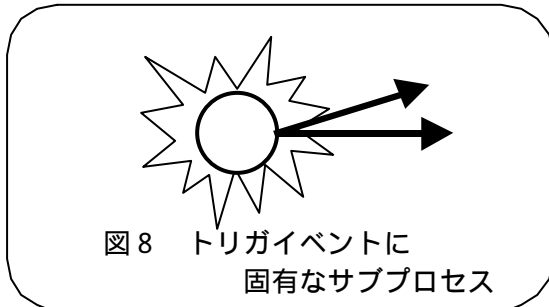
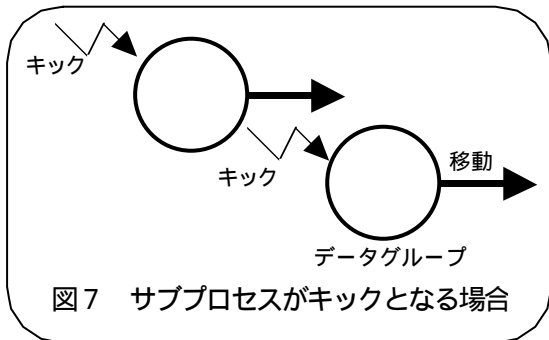
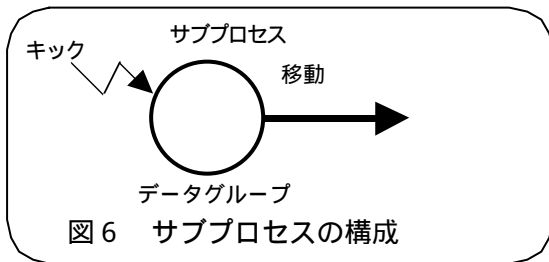
定義1にて、サブプロセスを実行する契機を明らかにし、定義2で機能プロセスの起動条件を設定している。さらに、定義3によりサブプロセスが連続する仕組みを与え、定義4により機能プロセスの構成条件を示している。定義4の機能プロセスの構成条件から機能プロセスの終了条件を得ることができる。即ち、あるサブプロセスが境界内の他のいかなるサブプロセスのキックにもなくなると、機能プロセスは終了する。

定義1から4で定義した機能プロセスでは、ひとつのサブプロセスに、複数のキックがあることについて言及していない。実際のリアルタイムシステムでは3章に定義した複数のイベントの組み合わせ条件を契機とすることは多くある。そこで、当該問題を扱えるように定義5、6を設定する。

複数のトリガイベントの扱いは、定義6にて示している。トリガイベントには、2種類ある。ひとつは、イベントの内容が個別のデータに関係なく、特定のサブプロセスのキックとなっている場合である。代表的なものに、時計がある。時計は、機能の起動契機となることができ、トリガイベントに成り得るが、時刻データのデータ移動があるとはかぎらない。

もう一つは、イベントの内容とサブプロセスが一对一に対応し、トリガイベントそのものがデータの移動を意味する場合である。例えば、電卓やエレベータなどのボタンスイッチ(以下ボタンと略す)の押下などがこれにあたる。ボタンを押し下げること自体がイベントになり、ボタンのデータのデータ移動が発生する。このようにデータに付随したイベントで、トリガイベントであるものをトリガイベントに固有なサブプロセスと呼ぶ。トリガイベントに固有なサブプロセスを図8のように表記する。

定義6で、複数のキックを許容できるのは、トリガイベント及びトリガイベントにより直接キックされるサブプロセスが引き起こすキックのみである。機能プロセスを起動する役割を担うトリガイベントは、境界の外部で発生し、ユーザから識別可能であることが必要であることから、機能プロセスの途中のサブプロセスでは、許容しない。



## 5. 拡張した定義の検証

本章では、前述の定義により得られる様々な機能プロセスを示す。

### 5.1. 様々な機能プロセス

以下に、様々な機能プロセスの形態を示し、拡張した定義に適合しているか否かを示す。

単純な例を図9に示す。トリガイベントの発生が最初のキックとなり、最初のサブプロセスを起動し、そのサブプロセスが次のサブプロセスのキックなり、一連のサブプロセスを順次実行していく。

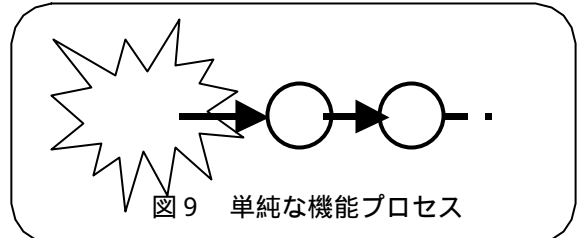


図10は、ひとつのトリガイベントがふたつの機能プロセスを起動する例である。ひとつのトリガイベントが2つの機能プロセスのキックとなっていること以外、図9の場合と同じである。

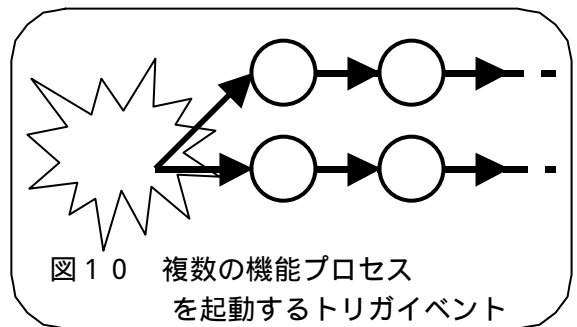
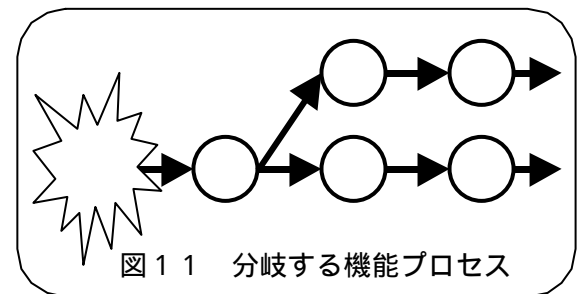


図11にひとつの機能プロセス内で、途中でサブプロセスが分岐する例を示す。前後2つのサブプロセスは、半順序関係が成立すれば十分で、途中から分岐することは許容できる。しかし、途中で合流する機能プロセスは、半順序関係を満たすものの、定義6から許容できない。(図12参照)



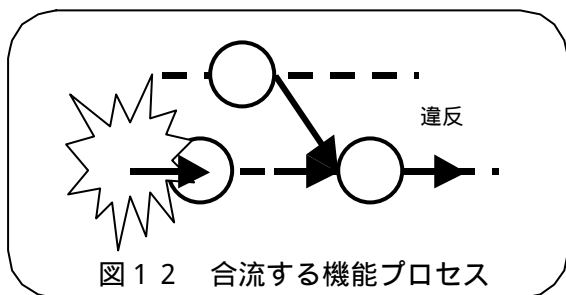


図 1 2 合流する機能プロセス

複数のトリガイベントによるキックで、許容している例を図 13 に、定義 6 から許容していないものを図 14 に示す。

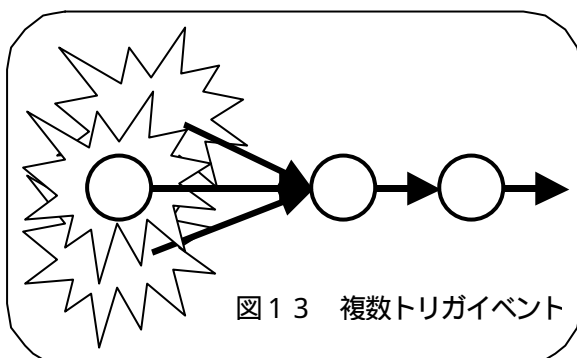


図 1 3 複数トリガイベント

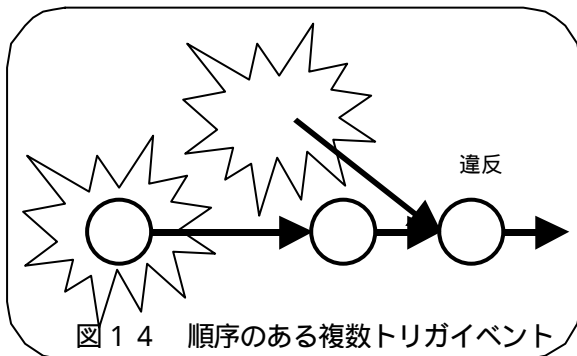


図 1 4 順序のある複数トリガイベント

## 6. エレベータの例による妥当性の検証

ここでは、エレベータを例題として、複数のイベントを組み合わせた条件に起因する機能の例を挙げ、本論文で提案した拡張定義の妥当性を示す。

### 6.1.エレベータ

エレベータは、かごの昇降と戸の開閉の動作に分け、各々個別に考えることができる。かごの昇降中には戸の開閉は閉状態のみで、戸の開閉動作はない。逆に、戸の開閉中には、かごの昇降動作はない。

前章までで述べた問題は、エレベータ例では戸の開閉時に発生する。

戸の開閉時に取り得る状態の中には、戸を開けているが、タイマーや閉ボタンで戸を開状態に遷移させることができる状態(以後こ

れを開状態と呼ぶ)とタイマーや閉ボタンで戸を閉じることができない状態(以後これを開維持状態と呼ぶ)の 2 種類がある。戸を閉じさせない条件は 2 つあり、一つは、利用者により開ボタンが押下されたままである場合、他は重量オーバーである場合である。開維持状態は左記 2 つの条件の内、少なくとも一方が成立している状態である。2 つの条件とも成立しなくなった時、開維持状態から開状態に遷移する。この様子の状態遷移を図 1 5 に示す。

### 6.2.エレベータ問題の計測

上記のエレベータの例で、開維持状態から開状態への遷移を、拡張定義を用いて計測する。

計測に当たっては、まず境界を設定する。計測するのは、エレベータの戸の昇降や開閉を制御するソフトウェアとし、エレベータの各種ボタンやセンサ等は境界外とする。

次に、トリガイベントを識別する。エレベータの仕様は、状態遷移図で記述され、状態間の遷移は、利用者のボタン操作や各種センサからの信号に起因する。これがトリガイベントとなり、状態遷移の契機となっている。従って、開維持状態から開状態へのトリガイベントを識別すると、開ボタンの解放と重量オーバーの解除となる。即ち、開ボタンの解放と重量オーバーの解除の両条件が成立することが、状態遷移の契機となる。

### 6.3.エレベータ問題における複数イベントの計測

上記 6.2 で示した開維持状態から開状態への状態遷移の場合が、3.1 章で述べた問題に該当することを示し、4 章で示した拡張定義に従って計測した結果を示す。

エレベータの状態遷移で、開ボタンの解放と重量オーバーの解除がそれぞれエントリとなる。この 2 つのイベントが 3.1 章の(A)~(C)の条件を満たしていれば、前述した問題に該当する。

開ボタンの開放と重量オーバーは、それぞれがイベントであり、この 2 つのイベントが両方成立して初めて状態を遷移する。どちらか一方のイベントのみでは、状態は遷移しない。従って、3.1 章の(A)を満たしている。また、2 つのイベントはエレベータの利用者が行う操作で、互いに異なる操作であることから、3.1 章の(B)と(C)を満たしている。従って、開維持状態から開状態への状態遷移は、複数

のイベントの組み合わせにより起動する場合に相当する。

次に、エレベータの例題における開維持状態から開状態への状態遷移の計測を行う。

エレベータの例の場合、開ボタンの開放と重量オーバの解除が 3.2 章で説明した 4 パターンの図中の E1 及び E2 に対応する。サブプロセスの集合には、状態遷移に関わる一連のデータ移動が対応する。

計測では、3.2.1 章で述べた And 型で計測することができる。現行の COSMIC-FFP において、And 型には、2 件のマニュアルへの抵触があったが、拡張定義では上手く当てはまる。現行の COSMIC-FFP のマニュアルへの抵触は、複数のトリガイベントが一つの機能プロセスを起動することと、待ち状態を含むことであった。拡張定義では、定義 5 及び 6 で複数のトリガイベントによる一つの機能プロセスの起動を許容している。また、エレベータの開ボタンの解放と重量オーバの解除から状態遷移を行う一連のサブプロセスへの個所は、待ち状態にあたるが、定義 6 で

許容している場合にあたり、定義どおりである。

## 7. おわりに

COSMIC-FFP の定義にキックの概念を導入することで、複数のトリガイベントの組み合わせ条件により起動する機能の計測を可能とした。また、エレベータのモデルを計測することで、拡張した定義が妥当であることを示した。

### 参考文献

- [1]A.Abran, "An Implementation of COSMIC Functional Size Measurement Concepts", Keynote presentation at the FESMA 99 Conference Amsterdam, Oct. 7, 1999
- [2]A.Abran, J.-M. Desharnais, S.Oligny, D. StPierre, C.Symons, "COSMIC-FFP Measurement Manual, Version 2.0", 1999
- [3]A.Abran, C.Symons, S.Oligny, "An overview of COSMIC-FFP field trail results", ESCOM 2001 London, April 2-4, 2001

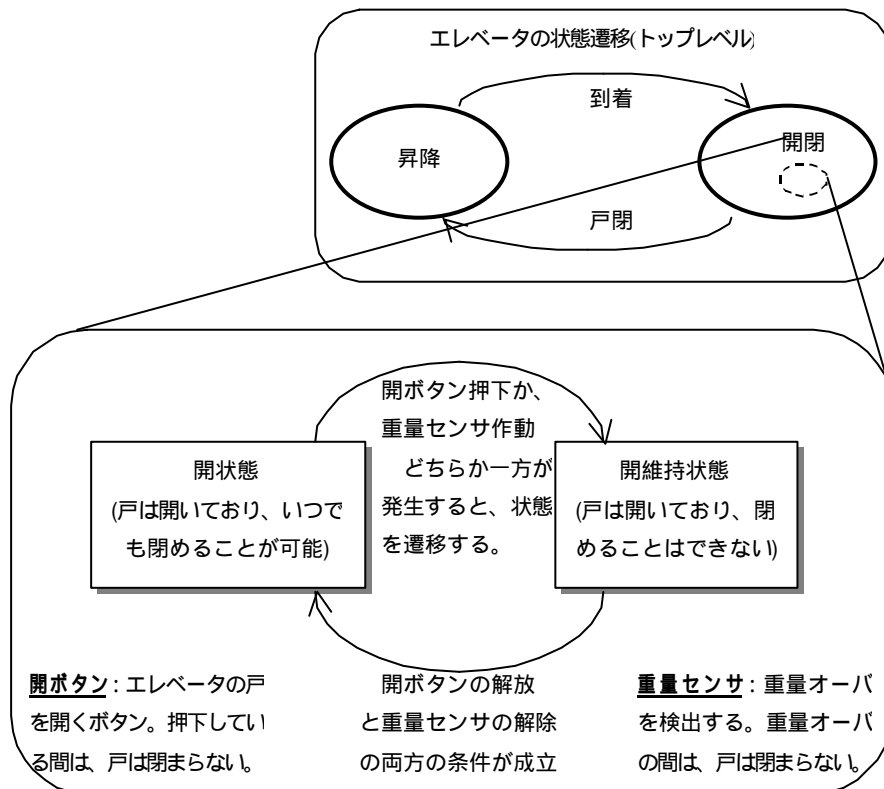


図15 エレベータの状態遷移