Vol.2016-SE-193 No.18
2016/7/14

“Dependability Assurance Framework for Safety Sensitive Consumer Devices” 標準化

宮崎比呂志[†1] 石崎直哉[†2] 田口研治[†3]
松野裕[†4] 春山浩行[†5] 大畠明[†6]

概要：自動車,スマートハウス,スマート家電,ロボット等の"消費者機械(Consumer Devices)"と呼ばれる一般消費者が使用する製品／システムに対し,安全性／ディペンダビリティを確保する開発方法が必要となる.そのために,これらの機器の開発方法の標準を OMG(Object Management Group)で制定した.本稿においては,その標準化された開発プロセスにフォーカスを当てそれについて述べる.

キーワード：消費者機械，安全性，ディペンダビリティー,システムズエンジニアリング，モデルベースシステムズエンジニアリング,メタモデル, BPMN, OMG

Standardization of “Dependability Assurance Framework for Safety Sensitive Consumer Devices”

HIROSHI MIYAZAKI[†1] ISHIZAKI NAOYA[†2] KENJI TAGUCHI[†3]
YUTAKA MATSUNO[†4] HIROYUKI HARUYAMA[†5] AKIRA OOHATA[†6]

Abstract: In accordance with emergence of IoT/Industrie4.0, it is required to establish a development method for safety/ dependable sensitive systems so-called “consumer devices”, such as automobiles, smart-houses, smart-appliances, robotics, etc. which are operated by end-user (general consumers) directly. To accomplish this purpose, we standardized the development method for such devices in OMG (Object Management Group) procedure. This article introduces the result of the standardization, especially focusing on the development process.

Keywords: Consumer Devices， Dependability, Safety, Systems Engineering, Model Based Systems Engineering, Metamodel, BPMN, OMG

1. はじめに

近年,IoT(Internet of Things)や Industire4.0 の出現等に伴いシステムの複雑性が高まる一方,その安全性／高信頼性を保証する手法の要求が切迫してきている.特に,一般消費者が直接操作するシステムにおいては,訓練等を受けた専門のオペレータが操作する機器に比べて,より高度な安全性／高信頼性が求められる.なぜなら,一般消費者は,特殊で高度な知識・技能を有する訳ではなく想定外の操作を行う可能性が高い,または,様々な環境において適切に操作する対応力が備わっている訳ではないからである.そのような状況においては,システムや製品がより高度な安全性／高信頼性を必要とする.

しかしながら,一般に ISO26262[1]のように安全性を規定する規格は存在するが,安全性ばかりでなく定常的に動作の継続性を確保する高信頼性について言及するものは存在しない.また,そのような高信頼性システム／製品の開発プロセス／品質保証の方法を規定するものが存在しない.今後の IoT の進展を考慮すると,安全／高信頼なシステム開発方法が必要となる.このような高信頼システムの開発標準を確立するために,Object Management Group(OMG)[2]において一般消費者が直接操作するシステム／製品の満たすべき要件及びその開発方法を規定する標準化を行った.この標準は Dependability Assurance Framework for Safety Sensitive Consumer devices[3](以下 DAF)と呼ばれる.ここで Consumer Devices とは前述の一般利用者が直接操作する自動車,スマートハウス,スマート家電,ロボット等を指す.

この標準は,2016 年 2 月に OMG 仕様として正式に発行されたので,ここではその結果について述べる.

DAF においては,3 つの観点に基づいて規格を規定している.第一は,Dependability の概念規定を行うためにその構成要素を意味論モデルによって規定する.第二は,Dependability を確保したシステムを開発するための工程のプロセスモデルを規定する.第三に,高信頼性システム開発の証跡を取るための,構造化された論証を表現するた

†1 富士通株式会社
Fujitsu Ltd.
†2 トヨタ自動車
Toyota Motor Corporation
†3 産業総合研究所
National Institute of Advanced Industrial Science and Technology
†4 日本大学
Nihon University
†5 情報処理推進機構
Information-technology Promotion Agency, Japan.
†6 株式会社テクノバ
Technova Inc

めの方法を規定している.これは,各種システムトラブルが頻発しているが,その際,その設計／製造の瑕疵の有無を明確にするためにも重要な役割を果たすものである.

本稿では,前述の 3 つ観点のうち,2 番目の Dependability システム開発のためのプロセスモデルについて焦点を当ててこれについて述べる.

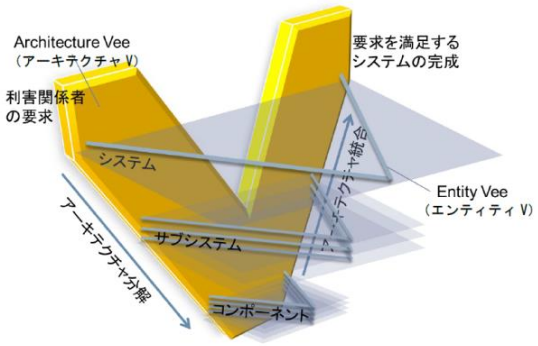
Section 2 では,DAF で規定しているプロセスのメタモデルについて述べる.Section 3 では,Section2 で規定したメタモデルに準拠する BPMN(Business Process Model and Notation)[4] のプロセスフローの記述について述べる.Section 4 では,考察を述べる.Section 5 においては,結論及び今後の課題を述べる.

2. Dependability プロセスモデルのメタモデル

2.1 基本方針

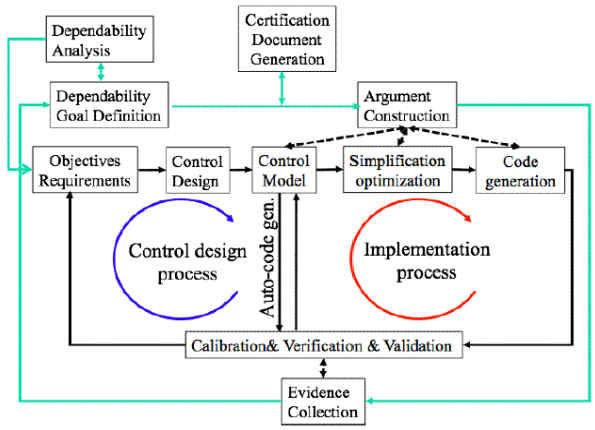
Dependable なシステム開発のプロセスモデルを構築するためには,基本的に Systems Engineering[5]もしくは,Model Based Systems Engineering[6]に従うことによって実現できる.したがって,本標準化においてもこの方針に従う.

したがって,Dependability プロセスモデルにおいても,システムの構成単位毎に適切なイテレーションを行うモデルとすることが必要となる.特に,重要なのは,イテレーションの判定基準となる Verification 及び Validation をすべての作業毎にこまめに行い不具合を早期に検出し,後工程での手戻りを減らすことが可能なプロセスにすることにある.[図 1]



[図 1]

このような考え方を背景に,Dependability プロセスを構築する.実際の Dependability プロセスのイメージは図 2 に示されるように,設計を行いながら実装を行い,Verification & Validation のフィードバックを即時に行いながら全体のリファイン／リファクタリングを行う.



[図 2]

このようなイメージのプロセスを規格化するために,本標準では,プロセスのメタモデルと具体的プロセスフローを規定した.

2.2 メタモデル

OMG/DAF RFP[7]では,Dependability プロセスのセマンティクス定義をするために,そのためのメタモデルを規定することを求めている.したがって,当標準の中で,UML[8]ベースのタメモデルを規定する.

2.1 で述べたように,Dependability プロセスは各作業をイテレーティブに行い,成果物が常に要件に適合するようにこまめにフィードバックをして確認しなくてはならない.特に,消費者機械のように要件が不明確であり,ターゲットが定かでないシステムに対しては,ある作業を行いその Validation & Verification を頻繁に行い,その結果に依存する形で,次の作業が非決定的に選択されるようにしなくてはならない.つまり,各作業が任意の順番で実行されるように構成しなくてはならない.

このようなプロセスに適したメタモデルを表すと,図 3 図 4 のようになる.

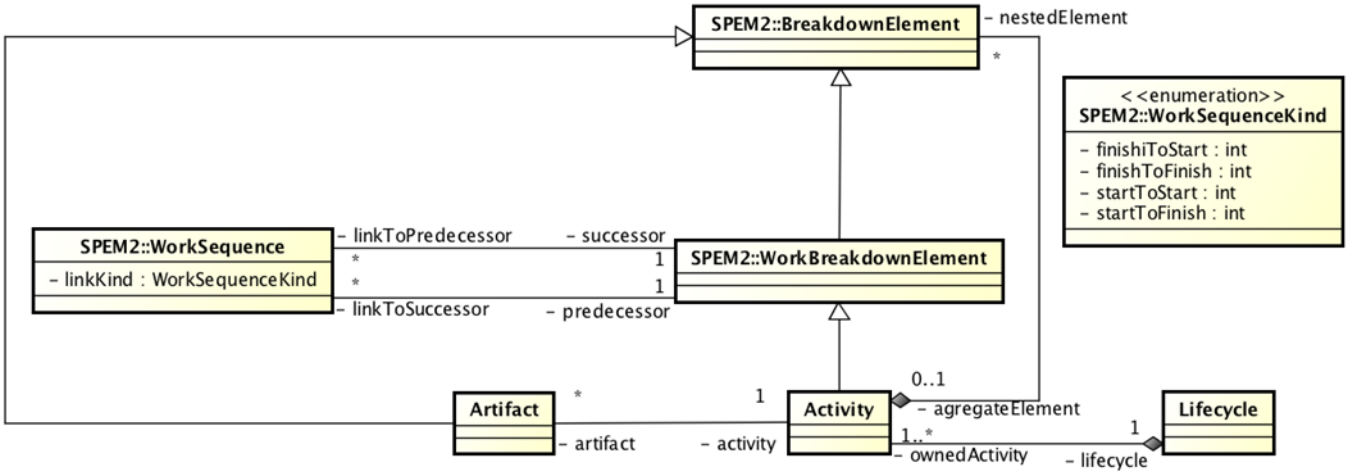
つまり,ある作業を行い,図 2 のようにその Validation & Verification を行い,その結果に基づき任意の後続する作業を行うことが可能なプロセスを表わさなくてはならない.

図 3 は Dependability プロセスの全体図であり,図 4 は図 3 の中のクラス Activity 配下の個々の具象クラス(要素)を表す.

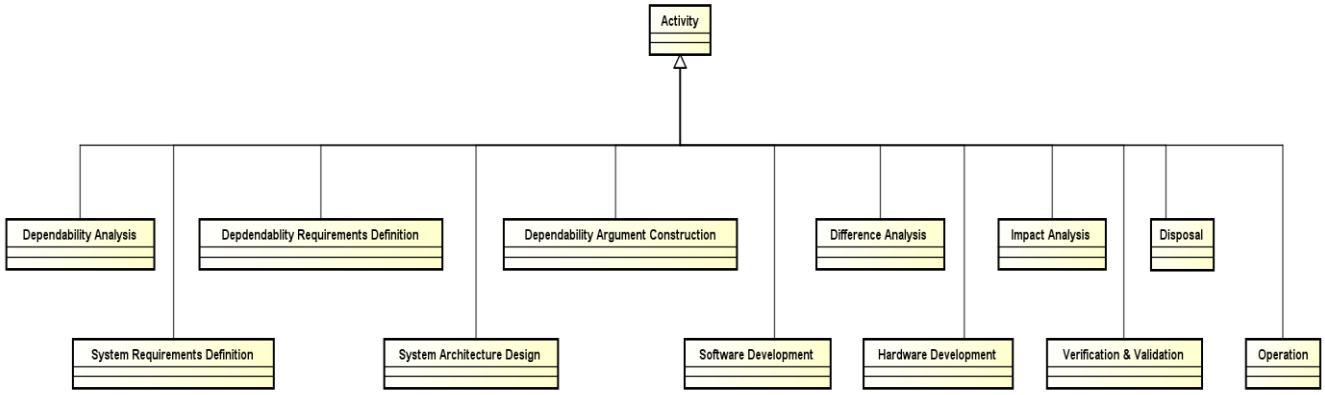
ここで,BreakdownElement は抽象クラスであり,その specialization がクラス Activity であり,この Activity が BreakdownElement を aggregate することで任意のリカーシブ構造を実現する. 個々の Activity は WorkSequence で示される 2 項関係で表されることによって,シーケンシャルな前後関係,並列関係を表現可能としている.Activity の Specialization として具体的な具象クラス(個々の具体的作業)であり,任意の作業順序を表すことが可能である.

図 4 は,Activity の Specialization クラスを表している.つ

まり,具体的 Activity の内容として Dependability Analysis, System Requirements Definition, Dependability Requirements Definition, System Architecture Design, Dependability Argument Construction, Software Development, Hardware Development, Difference Analysis, Impact Analysis, Verification & Validation, Disposal, Operation が Activity の具象クラスになるので,実際の作業としてこれらが具体的なものとなり,これらの任意の組み合わせによってプロセスが構成される.



[図 3]



[図 4]

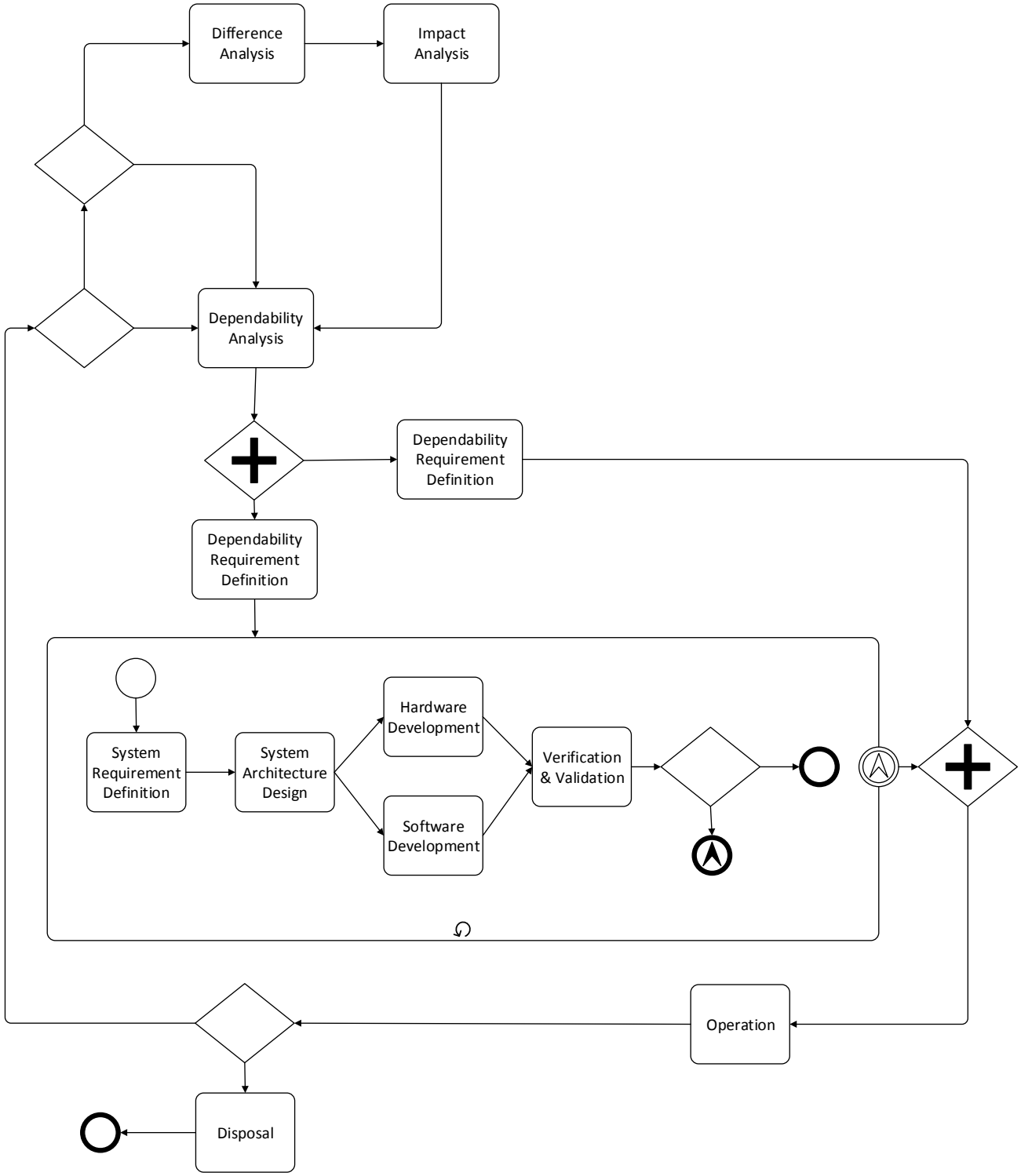
3. BPMN による Dependability プロセスモデル

次に,このメタモデルに準拠したプロセスモデルを表す.DAF RFP では BPMN を用いてプロセスモデルを表すことを求めているため,それに従う.何故なら,メタモデルによってセマンティクス定義は行うが,これはアブストラクトシンタクスであり具体的なプロセスモデルにはなっていないため,抽象度が高く具体的なプロセスが曖昧となり,Dependability プロセスとしての固有のプロセスを理解することができない.これを解決するために具体的なプロセスとして表す.

Dependable なシステム／製品を開発のためには,実際に使用され品質が保証された実績のある部品／半完成品を利用することによって構築可能である.したがって,基本方針としては,既に開発され,高信頼性が保証された部品／製品の再利用を前提にした開発プロセスとする.それゆえ,再利用前提の開発プロセスとなるようなモデルを構築する.そのために,再利用部分と新規開発部分の差異を分析するための Difference Analysis, 新規開発による既存部品／製品への影響を分析する Impact Analysis を行うことが必要となる.

また,Dependability を確保するために,Dependable とするための要件とそれを実証するための証跡から構成される Dependable Case[9] を構築するための Dependability Argument Construction を開発の初期に行い,Dependability が確保できるプロセスを構築する.BPMN で表すと図 5 のよう

になる.図中で円形の矢線がマークされている Activity があるが,これはその Activity を任意回繰り返すことを表している.この Activity では,System Requirement Definition から Verification & Validation までの Activity を繰り返し行うことを意味している.つまり,System Requirement Definition, System Architecture Design, Hardware Development, Software Development の通常の開発を行うが各イタレーションが終了する毎に Verification & Validation を行い,必ず要件に適合しているかを検証し,高信頼性(要件)を確保しているかを確認する.それによって,迅速で手戻りの少ない開発を実現している.



[図 5]

4. 考察

前述のプロセスでは,System Requirements Definition～Verification & Validation までを逐次的に行うモデルとした.つまり,固定的に System Requirements Definition→System Architecture Design → Hardware Development ｜ Software Development→Verification & Validation(ここで,“｜”は並行処理を表す)を実行するということを規定している.しかしながら,MBSE 及び図 2 に従うと Verification & Validation は各作業の直後に行なわれ,Verification & Validation の結果に基づき後続する作業が状況によって動的に決定されフィードバックを行い,手戻りを少なくすることとなっている.それを考慮すると上記の逐次的な部分を固定的に行うというのは,現実的なプロセスとしては適切ではない.また,2.2 のメタモデルの準拠性からしても各作業は任意の後続作業が動的に決定されるようになっているため,固定的な逐次プロセスは適合しない.

この問題点を解決するためには,System Requirements Definition～Verification & Validation の各作業を任意に行えるようなモデルにしなくてはならない.そのプロセスモデルを図 6 に示す.

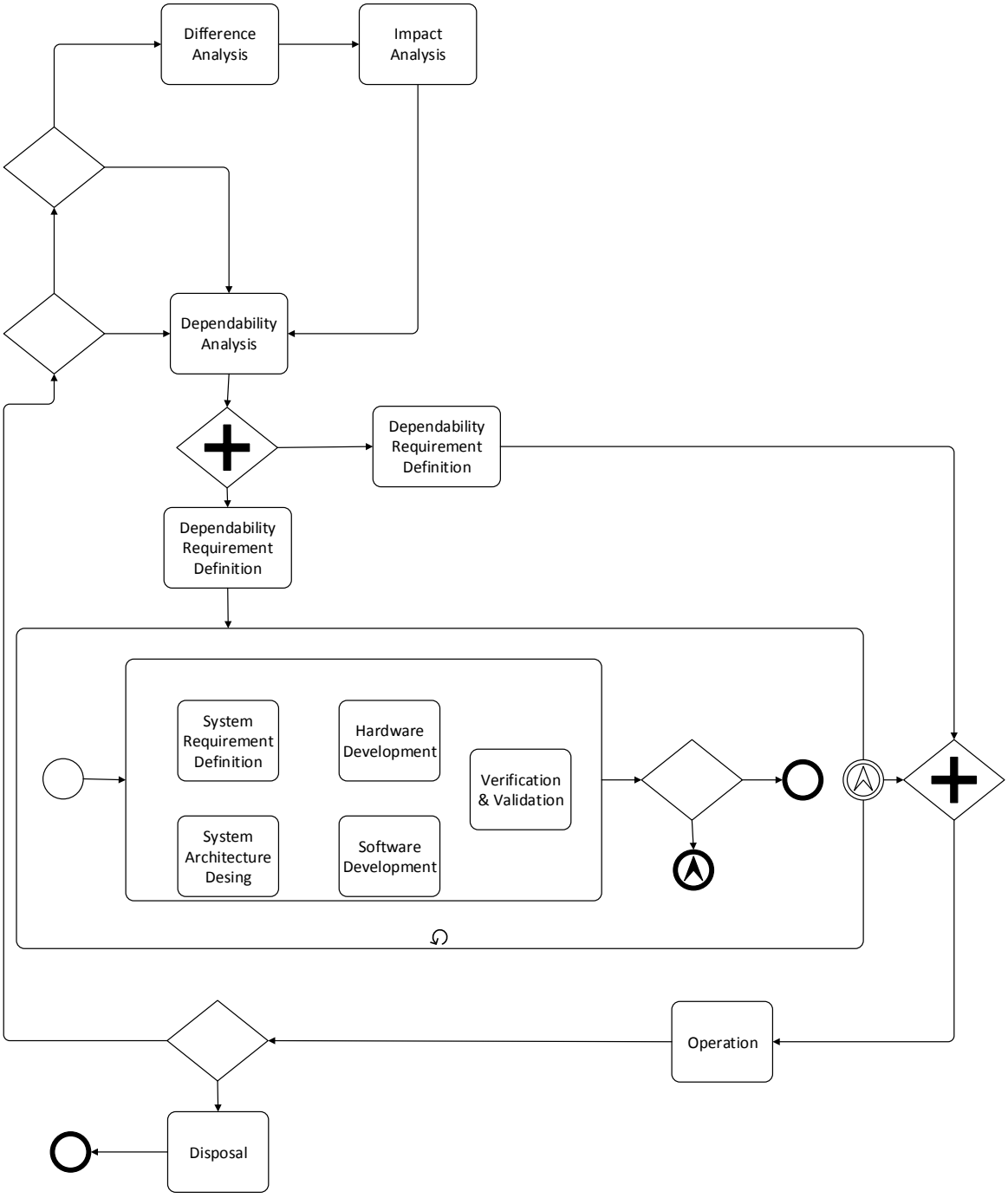


図 6

図 6 では,System Requirement Definition ～ Verification & Validation までを任意の手順でイテレーションできるようにしたが,実際には,手戻り等は本来 Difference Analysis, Impact Analysis 等にフィードバックされる場合もありえる.これらを考慮すると,System Requirement Definition～Verification & Validation だけを繰り返しの範囲とするのではなく,更に上流の作業を繰り返しの範囲に入れるモデルとすべきと考える.

5. まとめ・今後の課題

本稿においては,消費者機械開発に対し Dependability を確保するための方法のうち,特に開発プロセスに着目し OMG 標準化を行った成果について述べた.

基本的には,Verification & Validation を頻繁に行うことによって,品質の確保を行うことができるようなプロセスが必要であり,そのためのメタモデル／BPMN のプロセスモデルを構築することができた.

当規格は,Dependability を必要とする製品／システムすべてに対して有効な方法である.今後は,実際にこの方法を適用し,その有効性を実証することが必要である.

6. 謝辞

当標準化および本稿を執筆に関し,ご協力いただいた皆様に謝辞を表します.IPA　委員の皆様にお礼を申し上げます.
また,OMG 標準化において審議いただいた皆様に感謝いたします.

7. 参考文献

[1] ISO-26262: Road vehicles Functional safety – Part1- Part 9 (2011).
[2] http://www.omg.org
[3] Dependability Assurance Framework for Safety-Sensitive Consumer Devices (SSCD) Specification, formal/16-02-01 (2016)
[4] ISO/IEC 19510, Information technology - OMG Business Process Model and Notation
[5] ISO/IEC/IEEE 15288 Systems and software engineering -- System life cycle processes
[6] モデルに基づくシステムズエンジニアリング, 西村秀和　総監修, 日経 BP, 2015.
[7] Dependability Assurance Framework For Safety-Sensitive Consumer Devices Request For Proposal, sysa/13-03-20
[8] ISO/IEC 19505, Information technology - OMG Unified Modeling Language (OMG UML)
[9] Yutaka Matsuno, Kenji Taguchi. Parameterized argument structure for GSN patterns. In Proc. IEEE 11th International Conference on Quality Software (QSIC 2011), pages 96–101, 2011.