

つながる世界の検証

株式会社ベリサーブ IT検証産業協会 (IVIA) 技術部会 副主査 冬川 健一

1 はじめに

IoTやCPSなどの言葉で表現される、つながる世界のシステムやサービスを実現する上では、検証やテストの必要性が高まり、その範囲や数量が増大することが意識されている。

つながる世界はその言葉のイメージ通り、様々なモノやシステムをネットワーク接続することで実現される、あらゆるサービスや仕組み全般を指し、そこで必要とされる検証もかなり広範囲となる可能性が高い。

本稿では、つながる世界のシステムやサービスを開発・提供する現場において、考慮して進めたい検証のアプローチについて幾つか挙げていく。開発・検証対象とするシステムを特定していないため、若干抽象的な表現となっているが、筆者が現場で感じていることをなるべく分かりやすく伝えたいと考えている。

2 つながる世界のシステムとは

2.1 検証対象とするつながる世界

つながる世界について、現状では以下のキーワードで表現されることが多い。

① IoT (Internet of Things)

インターネットを経由して、様々なモノやサービスやクラウドシステムなどがつながった世界

② CPS (Cyber Physical System)

現実世界の各種センサと、そこから得られる大量のデータをクラウドシステムなどで分析・活用する世界

③ SoS (System of Systems)

個々のシステムや、個々の目的のIoTやCPSを接続・組み合わせることにより、新しいサービスや機能を実現する世界

本稿で検証対象とするつながる世界は、つながる経路やつながるもの(製品やシステムなど)を特定せず、上記

①と②を包含した③のSoS(System of Systems)を前提に検討を進めるものとする。(以降「つながるシステム」と呼ぶ)

次に、検証対象とする個々の製品やシステムの開発状況については、以下の分類が可能である。

A. 出荷済のもの同士をつなげて検証

個々のものやシステムの機能は開発済で、主には接続したときの機能動作や、安全・安心にかかわる機能動作の検証を行う。

B. 出荷済のもの、開発中のものをつなげて検証

開発対象のものやシステムを、外部のいわゆる「既存環境」に接続し、目的の機能を実現できることを検証する。

C. 開発中のもの同士をつなげて検証

つなげて利用するための仕様を決め、それに合わせて個々のものやシステムを開発する。多くの場合、単体のシステム開発と時期が重なるため、検証範囲が広くなりやすい。

一般的にIoTは上記Aをイメージして語られることが多いように思うが、実際には個々のシステム機能や、つなげて使うための機能は日々開発されている。そのため、検証の現場では上記のBやCのボリュームも多く、検証範囲を網羅しつつ検証項目を増やしすぎないことが大きな課題となってきている。

2.2 システムへの要求の変化

SoS(System of Systems)によって新しいサービスや機能を実現する場合において、システム(SoS)に求めること、あるいはシステムの価値が、「人の要求を実行するシステム」から、「人に代わり要求するシステム」へ変化すると考えられる。

●人の要求を実行するシステム

従来のシステムの多くは、人がシステムへの要求として指示を与え、システムはその指示に基づき動作を行う。このようなシステムに求められること・要件は、人の要求を実行することであり、実行できない場合に適切な応答を示すことである。このシステム例としては、PCやスマートフォンのアプリ、銀行のATMや自動車のパワーステアリングシステムなどが挙げられる。

●人に代わり要求するシステム

SoSやCPS(Cyber Physical System)に属するシステム形態では、多数のセンサ情報や各種データに基づき、システ

ムが判断して動作するものが増えている(この場合、専用のシステムが判断して他のシステムに指示を与える形態と、各システムが個別に判断して動作をする形態がある)。

すなわちSoSは、ある状況と条件化において、人に代わり判断を行い、各システムへの動作要求も行うシステムと言える。また、このようなシステムの価値は、どこまで人に代わることができるか、どこまで適切に、安全・安心な判断ができるかということになり、そこが競争領域となる。このシステム例としては、自動車の自動運転やIndustry 4.0と言われるスマート工場、スマートハウス(「つながる住宅」とか「AI住宅」と言われる場合もある。防犯性や居住性などの向上を目的としている)などが挙げられる。

2.3 人の要求を検証する

システムが人に代わり要求するようになることへの変化は、時代やシステムの進化に伴うある意味自然な展開と言えるが、システムを検証する側にとっては質的に大きな転換を求められることとなる。

従来の人の要求を実行するシステムでは、各種状況を検知して判断するまでは人が行うことであり、システムの検証は人の行動・指示に対する結果を確認することである。すなわち人が検知して判断した内容は検証対象とはならない。例えばガスコンロの上に新聞が乗っている状態で点火することや、カーナビが示す道路上に横断者がいても前進することは、システムの問題とはならないためその検証を行うことはない。

それに対し人に代わり要求するシステムでは、各種センサが検知した状況やネットワークから得られる各種データから、システムがどのような判断をするかが検証対象となる(もちろん、その後の各システムへの要求に対する結果も検証対象となる)。その過程を人に置き換えると、ある状況において、人が何を見て、聞いて、動きを感じて、その後を予測して、どう判断して要求を抱くかが検証項目となる。

システムを検証する側に求められる質的な転換とは、人の要求過程を検証項目とするところにある。人の目に映る景色は皆同じでも、着目するところは人によって異なり、どの時点でどう判断するかも人によって異なる(すなわち、テスト項目での期待結果が単純には定まらないことにもなる)。人に代わり要求するシステムに対して検証したいことは、いわゆる「常識的な判断」がされることであり、「常識的な判断」に含まれる状況や条件やシーンの選択が重要な検証項目となる。しかし現実にはその常識域や非常識域の選択肢があまりにも多数あり、その中

で必要な検証項目を網羅しつつ、項目やコストを抑えるということが困難な状況となる。

2.4 つながるアーキテクチャ

システムの検証を行う上で、機能の追加・変更や条件の変化に対する影響範囲を把握することは、重要な要件となる。つながる世界は、システムをつなげることで色々な機能を実現しているため、この影響範囲がどんどん広がる。

システムの検証を行う際に、影響範囲を把握するためのアプローチはおおむね以下である。

- ① システム全体の構成・アーキテクチャを把握する
- ② 機能とサブシステム、ハードウェア要素、データなどの関係を把握する
- ③ 追加・変更される機能やハードウェア要素やデータ、変化する条件のつながりを辿る

つながるシステムでは、この①～③の把握が困難になりやすい。

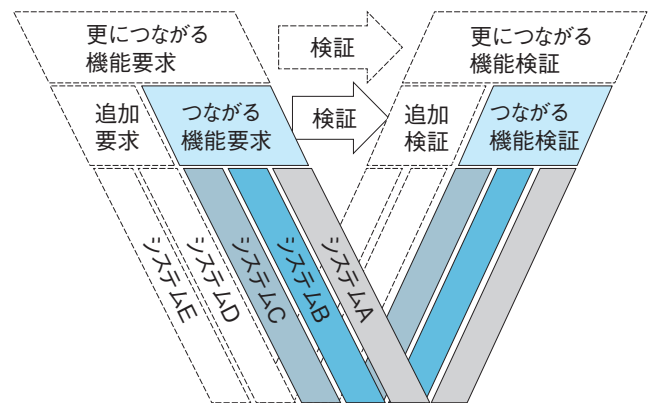


図1 つながるシステムのV字開発イメージ

つながるシステムのV字開発イメージを図1に示す。つながるシステムの機能要求は、システムA・B・Cをつなげることで実現したり、更に便利な機能をシステムD・Eもつなげて実現したりするイメージとなる。このような場合、システム全体のアーキテクチャや機能の影響範囲の把握が難しくなり、何かを変更する場合の設計や検証の難易度が高くなる。また、現実的には図1のように独立性の保たれた階層的な構造とはならず、システムやサブシステム間で直接やりとりされる場合もあり、アーキテクチャを荒らす原因にもなりやすい。

つながるシステムでは、図1のように上位要求とシステム、その下のサブシステムの間を階層化して理解する必要がある。このような複雑なシステムのアーキテクチャを可視化するために、システムズエンジニアリングやDSM (Design Structure Matrix)などの理解と活用が考えられる。

2.5 つながる開発組織・プロセス

つながるシステムの開発では、システムだけでなく開発組織やプロセスも必要に応じて適切につながる必要がある。図1のシステムA・B・Cの開発は、別の企業や組織で行われる場合が多く、つながる機能要求を実現するために、仕様を決めたり設計を分担したりすることが行われる。

単一のシステムAの開発でも、各組織やチームで分担して作業が行われるが、マネジメントの主体が明確なため統制が効きやすい。つながるシステムであるSoS(System of Systems)は、全体をマネジメントする主体が存在しないものであるとも言われており、組織面においても同様にシステムA・B・Cを束ねてマネジメントする主体が存在しない状態となる。その結果、各システムの開発組織への役割や責任の分担が上手く行われず、つながる機能要求が正しく実現されなかったり、手戻りが発生したりということが起こりやすい。また、現状ではつながるシステムの開発にかかわる情報量が非常に多く、特定の情報の曖昧さが他の情報の決定を阻害するという伝播が、状況をより悩ましいものにしている。

システムの品質を高めるには、V字左側及び上流のプロセスで不具合を作り込まないようにする必要があり、つながるシステムの開発において、組織やプロセスを適切につなげることが課題となる。

3 つながるシステムの検証活動

3.1 検証範囲の考え方は同じ

ここまでつながるシステムについて色々述べてきたが、システムに対する検証範囲を考える場合において、つながるシステムであることを特別に意識する必要はないと考えている。

図1にも示してあるが、検証はシステムの開発階層・レベルごとに検討して実施するものであり、単一システムでも組み合わせたシステムでも、その開発階層・レベルで検証すべきことを組み立てることである。前述の人の要求を検証するテスト設計(検証項目の作成)は質的な転換ではあるが、そこに関してもその階層でシステムに要求されていることを検証することと言える。

テストの実行についても、上位階層のシステムを組み合わせたテストは、下位階層の各システムのテストが完了してから実施するなど、単一システムのテスト実行の組み立てと基本的には同じである。しかし前述の通り、つながるシステム全体の開発をマネジメントする主体が

存在しない場合があるため、テスト実行の組み立てには注意が必要で、実際には開発組織・プロセスへの関与・働きかけが必要となる。

また、メーカーや提供元が異なるシステムを組み合わせる行う検証範囲やコスト、及びその保証範囲をどのように分担するかについては、現状でも課題であり今後いっそう悩ましくなると考えられる。検証を行う側は、少なくとも検証する範囲、しない範囲をできるだけ明確にしておくことが求められる。

3.2 検証要求分析の重要性

検証要求分析とは、システムアーキテクチャや開発の経緯を把握して、検証すべき範囲や内容を抽出し、絞り込むことである。検証範囲や内容は、各システムの稼動実績や検証実績、追加・変更状況などによって範囲を絞り込むことが可能で、すべてのシステムのすべての機能の検証が必要なものではない。

依頼者の要求がイコール検証要求とはならない場合や、前述のように保証範囲やコストの関係で範囲を絞り込む場合もある。また、つながるシステムの場合には、つなげられるモノやシステムが明確ではない場合もある。そのような状況においても、検証するところとしないところを理由と共に明確にするのが検証要求分析である。

つながるシステムのように、対象とする範囲が広く責任範囲も明確になり難いシステムは、検証要求分析の重要性が高まる。また、そのためにはつながるシステムのアーキテクチャを把握することが重要で、そこが明確でない場合には、開発・設計側と共に情報を整理してまとめる活動を進めることも必要と考える。

IT検証産業協会(IVIA)のIT検証標準工法ガイドでも、テスト要求分析として行うべき項目を定義している。つながるシステムを特定したものではないが、適用は可能と思われるので参考にさせていただきたい。(http://www.ivia.or.jp/item/121.html)

3.3 対象箇所を特定したテスト設計

一般的に、システムが持つ機能のテストは、対象箇所(サブシステムやモジュール)を特定して行うことが多く、そうあるべきと考えている。対象箇所を特定することにより、どのようなテストで検証範囲が網羅されるのかが分かり、テスト項目を絞り込むことも可能となる。

一方で、テストの対象箇所を特定しないシナリオテストなどもあるが、その目的は実際の手順・操作の流れにおいて問題がないことを確認するものであり、網羅性確

認を目的としたものではない。

つながるシステムでも、対象箇所を特定したテスト設計をすべきであることは変わらないと考えているが、機能の実現にかかわる対象箇所が複数にわたるため、その思考が難しい場合がある。その結果、ある機能のテスト設計で、組み合わせ可能な複数条件を組み合わせることで、網羅性を高めると共にテスト効率を上げられると考えてしまう場合がある。そのようにして作られたテスト項目は、一見そこで何を確認したいのかが分かり難くなり、また、そのテストを実施して不具合が発生した場合の原因が特定し難い状況になり得る。

つながるシステムでは、機能の実現にかかわる対象箇所が見え難くなるため、単一のシステム以上に対象箇所を特定する意識を持ち、テスト条件を明確にするように心がけたい。なお、論理的な意図によるテスト条件の組み合わせは、検出すべき不具合を検出する上で必要なため、適宜進めるべきである。

3.4 開発プロセスへの関与が有効

検証で検出される不具合は、ほとんどがV字開発プロセスの左側で作られるものであり、上流工程である要求分

析や要件定義の工程で作られるものも多い。つながるシステムは、図1で示したように上流の要求階層が多段になり（上流に積み上げられていく状態）、要件の不備や抜け漏れを作ってしまう機会を増やすことになる。また、前述のマネジメント主体が存在しない組織・プロセス構造も、不具合を作り込む可能性を高めやすい。

検証活動として必要性が謳われているV&V(Validation & Verification)は、つながるシステムのように組織とプロセスが入り組んだ開発では、プロセスのV&Vの必要性も高まると考えている。プロセスのValidationは、組織として正しく作れる仕組みとなっているかを確認し、必要に応じて改善することであり、プロセスのVerificationは、取り決めた仕組みに従って組織が開発しているかを確認することと言える。図2にプロダクト（製品やシステム）とプロセスのV&Vの関係を示す。

近年、プロセス標準化などの必要性が多く取り上げられるようになってきていることも、つながるシステムやつながる組織・プロセスが増えていることと関係しているのではないかと考えている。つながるシステムの検証活動として、開発プロセスの改善や運用に積極的に関わっていくことは、プロダクトの検証と同様に重要な活動であると考えている。

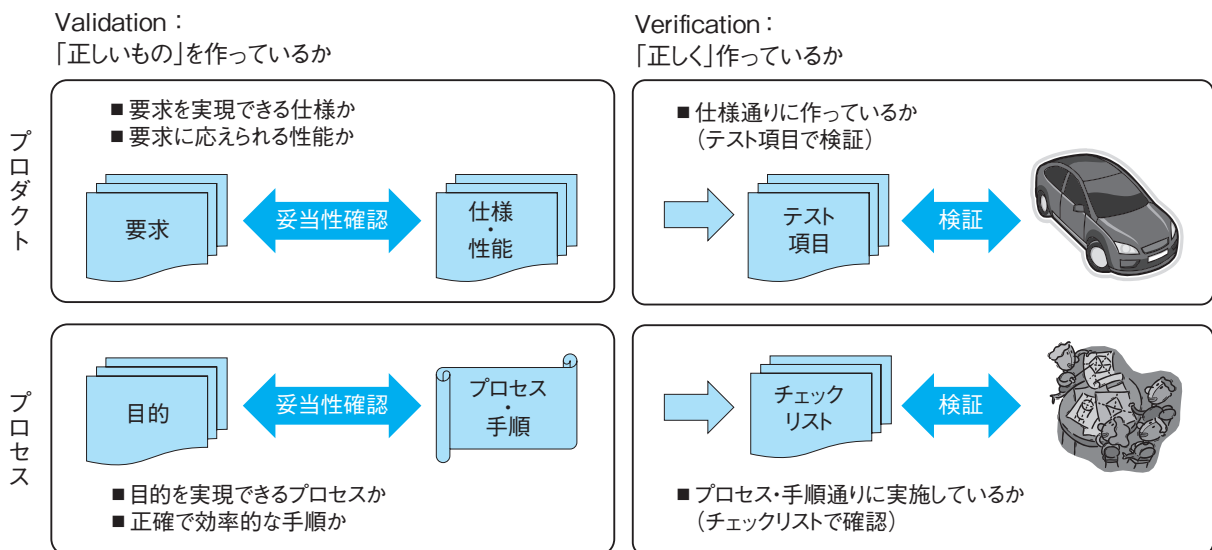


図2 プロダクトとプロセスのValidation & Verification

4 おわりに

以上、つながる世界の検証として概念的なことを幾つか述べたが、実際の開発・検証の現場では更に悩ましい各種課題に対して、試行錯誤しながら進めている状況と

推測している。

IT検証産業協会 (IVIA)でも、IoT検証・評価研究会として、安全・安心、接続性、V&V、品質説明などをテーマに、課題やニーズに対応するための検討を進めている。ここで述べたことに興味を持たれた方は、ぜひIVIAにご参加いただきたいと思います。