

東 京 電 機 大 学
博 士 論 文

民生品 IoT システム開発における
製品およびプロトタイプの第三者検証

Third-party verification of product and prototype
in the development of consumer product IoT Systems

2019 年 3 月

堂坂 辰

1. 序章.....	5
1.1 研究の背景.....	5
1.2 研究の目的.....	6
1.3 論文の構成.....	8
2. 第三者検証に関する関連研究と課題.....	10
2.1 第三者検証の国際規格と実施例	10
2.1.1 第三者検証の国際規格	10
2.1.2 ミッションクリティカルな分野でのシステム製品の第三者検証実施例	11
2.1.3 民生品システム製品の第三者検証の現状	13
2.2 第三者検証および IoT システムに関する関連研究	13
2.3 民生品 IoT システム製品に対する第三者検証実施の課題.....	14
2.3.1 本研究の位置づけ.....	14
2.3.2 IoT システム構成の特徴と民生品 IoT システム	15
2.3.3 民生品 IoT システム製品の第三者検証実施の課題.....	16
2.3.4 民生品 IoT システム製品開発におけるプロトタイプの目的.....	18
2.3.5 民生品 IoT プロトタイプシステムの第三者的検証実施の課題.....	18
2.3.6 課題と研究の目的.....	19
3. 民生品 IoT システム製品の不具合推定を含んだ第三者検証方式	21
3.1 はじめに	21
3.2 現状の第三者検証とその課題.....	22
3.2.1 第三者検証の運用形態.....	22
3.2.2 第三者検証でシステム製品の不具合箇所と原因を推定するための課題	23
3.2.3 ミッションクリティカルな分野で適用される検証の手法	23
3.3 システム製品の不具合の推定を含む第三者検証手法の提案	23
3.3.1 検出障害の推定機能をもつ第三者検証方式	23
3.3.2 検出障害の発生箇所と原因の推定方式	24
3.4 家庭電化製品の遠隔操作システムへの本手法の適用.....	29
3.4.1 家庭電化製品の遠隔操作システムのモデル	29
3.4.2 検証対象 IoT システム	30
3.4.3 検証手順.....	31
3.5 実施結果と評価.....	33
3.5.1 不具合箇所と原因の推定	33
3.5.2 不具合推定手法の評価.....	40
3.6 考察	41

3.6.1	不具合推定が不十分になる理由	41
3.6.2	システム運用へのフィードバック	42
3.6.3	第三者検証を適用する際の工夫点および限界点	42
3.6.4	ミッションクリティカルな分野に適用される検証の手法との比較	43
3.6.5	テストの観点についての考察	44
3.6.6	本手法の価値と限界	44
3.6.7	今後の課題	44
3.7	まとめ	45
4.	IoT プロトタイプの開発者による第三者的検証方式	46
4.1	はじめに	46
4.2	IoT プロトタイプの第三者的検証の意義	47
4.2.1	IoT システムの構成	47
4.2.2	プロトタイプの第三者的検証	48
4.2.3	ハザード検証の必要性	48
4.3	IoT プロトタイプの第三者的検証方式の提案	48
4.3.1	プロトタイプ検証方式	48
4.3.2	プロトタイプ検証の検証項目	51
4.4	実装	53
4.4.1	水耕栽培プロトタイプ	53
4.4.2	害鳥駆除 UAV プロトタイプ	56
4.5	テスト結果と評価および考察	58
4.5.1	水耕栽培プロトタイプのテスト結果と評価	58
4.5.2	害鳥駆除 UAV プロトタイプのテスト結果と評価	61
4.5.3	考察	63
4.5.4	従来の第三者検証との違い	65
4.6	まとめ	65
5.	IoT システムのプロトタイプの第三者的検証による開発の方向付け	66
5.1	初めに	66
5.2	IoT プロトタイプに対する第三者的検証による製品開発への方向付け	67
5.2.1	プロトタイプから製品への方向付け	67
5.2.2	プロトタイプに対する第三者的検証による製品完成までの距離の算出手法	68
5.2.3	ベータ版のユーザ評価	70
5.2.4	プロトタイプに対する第三者的検証による製品仕様の確定	70
5.3	実装	72
5.3.1	水耕栽培プロトタイプ	72
5.3.2	害鳥駆除 UAV プロトタイプ	76

5.4. 評価と考察.....	80
5.4.1 評価.....	80
5.4.2 考察.....	81
5.5. まとめ.....	81
6. 結言	83
謝辞	86
参考文献	87
(筆者関連発表論文)	87
1 章分	87
2 章分	88
3 章分	90
4 章分	92
5 章分	94

1. 序章

1.1 研究の背景

第三者とは、当事者ではないその他関係者のことを表す。情報システム製品に対して、開発者や利用者という当事者ではない、独立した組織で行う検証を第三者検証と言う。対象製品の開発部門および利用部門とは異なるテスト専門部門が、特定の基準および利用者の立場に基づいて製品の検証実施と結果判定を行う手法である。

システム製品は、製品の欠陥の顕在化を防ぐため、開発フェーズにおいて様々な開発手法やレビュー手法が提案され実施されている。しかし、システムに欠陥がないことを、開発者がテストによって完全に証明することは出来ない。システムを深く掘り下げた潜在欠陥の洗い出しを目的とした検証の実施はより基本的な事項であり、システム全体を把握する専任者による潜在欠陥の探索を目的としたシステム製品の検証は重要である。つまり、異なる評価軸から検証を行い多角的に品質推定精度を上げる、第三者検証の実施が必要である。

宇宙・航空・自動車・医療など、公共性が高く人命にかかわることから高信頼性が求められる、ミッションクリティカルな分野のシステム製品に対しては、開発部門とは異なる第三者の組織体制による第三者検証の実施が開発計画に組み込まれており、開発成果物を妥当性と有効性の観点で解析評価する第三者検証⁽¹⁾が行われている。

一般消費者の使用を前提として設計・開発される民生品分野のシステム製品は、USB、Bluetooth、スマートハウスで用いられる ECHONET Lite といった通信インタフェースなどの、単体機能は第三者検証が実施されている。しかしシステム製品全体に対しては、開発期間やコストが重要視されることから、開発部門とは異なる部門による第三者検証は実施されていない。

Internet of Things (IoT)⁽²⁾システムは独自に製品化され管理されている複数の製品が、オープンなインタフェースのネットワークを介して繋がりシステムを構成するという特徴がある。IoT システムの適用分野が社会インフラから民生品まで広がり、民生品のデバイスもインターネットと繋がることで機能が広がり、制御が自動化されてきている。IoT システムは構成要素が多くそれらは独自に製品化されていることから、開発者がシステム全体を把握することは困難になっている。このため、IoT システムに障害が発生した場合、原因の切り分けや責任分解が難しく、障害解決に時間がかかることが予想される。このことから、民生品 IoT システム製品についても、異なる評価軸から検証を行い多角的に品質推定精度を上げる、第三者検証を実施する必要があると考える。そこで、開発期間やコストが重要視される民生品 IoT システム製品に対して、開発者の視点で第三者検証を実施するという、新たな第三者検証方式を提案する。

さらに IoT システムは、新規性の高い開発要素があり、さらに自社の開発製品以外に、独自に製品化され内部処理をブラックボックスと見なす他のシステムをオープンなインタフェースで自社製品に繋いで一つのシステムとして機能させることから、プロトタイプを作成して開発システムの主要な仕様の実現性を確認することが有効である。プロトタイプを、開発項目の優先度付けやユーザ要求仕様の確認のためのマイルストーンとして位置づけ客観的評価を行うことで、製品化までに行うべき事柄が明確になり、開発期間の短縮を図ることができる。プロトタイプの客観的評価を行うためには、第三者検証が有効である。しかし、プロトタイプをそのまま継続開発して製品化できるかという観点での、プロトタイプに対する第三者検証の方式や評価項目はない。このため、そこで、民生品 IoT システム製品のプロトタイプに対して、開発者の視点で第三者検証を実施するという、新たな検証方式を提案する。

1.2 研究の目的

あらゆる「モノ」がインターネットへつながる社会を目指す IoT は、モバイル技術、無線通信、クラウド技術の発達により IoT 基盤が整い、社会インフラだけでなく一般消費者との係りが深い民生品分野での利用が始まっている。IoT システム製品は、自社の開発製品以外に一つ一つは独立して開発された他社製品をオープンインタフェースのネットワークで繋いでシステム化していることから、システム製品の品質・強靱性・安全性を高めるために、自社製品の開発者によるテストだけでなく、システム全体を対象とした第三者による検証が必要と考えられる。しかしながら、民生品 IoT システム製品の第三者検証の実施にあたっては解決しなければならない課題がある。本研究では、IoT システムの製品およびプロトタイプを対象とした、開発者による第三者検証実施という方式を用いて、課題の解決を図る。

本研究の目的はつぎの3項目である。(1)民生品分野のIoTシステム製品開発において、開発者による第三者の視点での第三者検証の実施方式を提案する。(2)IoTシステム開発で有効と考えられるプロトタイプ開発について、開発者による第三者の視点でプロトタイプの第三者検証の実施方式を提案する。(3)プロトタイプの第三者検証結果に基づき、製品化の難易度の把握とユーザ要求仕様の確定を行う、開発の方向付けの方式を提案する。

本論文においては、「第三者検証」および「第三者的検証」をつぎのように定義して使用する。「第三者検証」は、開発者とは異なる第三者が開発者とは異なる評価軸で、システム製品に対して実施する検証を表す。「第三者的検証」は、開発者が第三者検証を模して、プロトタイプに対して客観的に実施する検証を表す。

(1) 民生品 IoT システムの不具合推定を含んだ第三者検証の実施方式

開発者とは異なる観点で試験を実施し、製品の妥当性と有効性を検証するという意味で、第三者検証は有効であると考えられている。しかし、民生品システム製品に対する第三者

検証機関は存在しておらず、また製品品質とともに開発期間とコストも考慮する必要があることから民生品のシステム製品に対する第三者検証は一般には行われていない。

IoT システムを設計の観点から見た場合、複数のシステムが繋がって全体として一つのシステムを構成しているが、それぞれのシステムは独自に製品化され管理されていることから、システムのコンセプトは一人では設計できず、複数の専門家の共同作業となっている。

システムの潜在欠陥は、当該システムだけでなく、上位のシステムや、動作環境や他システムとの連携オペレーションに関するものなど、いろいろな箇所が存在する。これら潜在欠陥が現実的なものとなると、様々なシステム障害として現れる。これらのシステムの潜在欠陥の除去、あるいは除去不能なケースの場合は発生時の対処法を事前に準備して運用することで、障害の顕在化を未然に防ぐことが、IoT システムの設計者に求められる。

潜在欠陥の作りこみを防ぐため、開発フェーズにおいて、様々な開発およびレビュー手法が提案されている。一方システムを深く掘り下げた潜在欠陥の洗い出しを目的とした第三者検証の実施は、より基本的な事項であり重要であると考えられる。しかしながら IoT システムは、独自に製品化され管理されている複数の製品が繋がってシステムを構成していることから、開発者以外では検証対象の機能・性能・責任範囲の特定が難しい。このため第三者部門が検証を行うためには開発部門からの技術支援が必要となることから、実施にあたり検証部門との間で障壁とも言える困難さが発生すると予想される。

以上の要求に対して、IoT システムを構成する製品群をサブシステムの連携として定義し、開発部門が製品の機能・サブシステム構造・ユーザインタフェース・守るべき規格、および検証環境を、「検証要求仕様書」として第三者検証部門に提示することで、必要な時期に編成される検証部門で検証環境構築と検証項目と判定基準の設計が可能となる。さらに、第三者検証部門で検出した障害結果からサブシステム単位に「障害の発生箇所と原因」を推定し、開発部門にフィードバックする取り組みが可能となる。

このように本研究では、開発部門にて IoT システムの構成要素と機能仕様および検証環境の明確化を図ることで、民生品 IoT システムにおける開発部門主導による、検証部門での IoT システムの第三者検証を可能とする。特に検証部門で検出した障害の発生箇所と原因の推定を行う方式を提案する。

(2) IoT プロトタイプシステムの開発者による第三者的検証方式

IoT システムは、構成要素が多様かつ適用の範囲が広いため、システムの製品化には、製品の狙いと基本となる技術基盤を決定し、解決すべき課題と目標を定めた開発が必要となる。このため一旦プロトタイプを作成し、主要な仕様の達成度を確認するが多いが、プロトタイプは開発過程の通過点の扱いであり製品と対比させた評価は行われていない。プロトタイプを開発の進捗をモニタリングするマイルストーンとして位置づけ、機能達成度について客観的評価を行うことで、プロトタイプからそのまま開発を続けて製品版とするために必要な不足部分の指摘を行うことが可能となり、製品化までに行うべき開発項目が

明確になり開発期間を短縮できるという効果が期待できる。しかし、IoT プロトタイプシステムを対象にした第三者検証機関は存在しない。このため開発者が第三者の視点で検証を行い、IoT プロトタイプを客観的に評価する方式が求められる。

以上の要求に対して、開発部門でプロトタイプに対して、機能・性能・規格・責任範囲を明確にしたプロトタイプ機能仕様書を a 作成し検証対象の明確化を図り、さらに①仕様達成度・②拡張容易性・③ハザード対策という客観的観点に基づいて、「検証設計書」を作成することで、開発者が第三者的視点でプロトタイプ検証を行うことが可能となる。

このように本研究では、開発者が IoT システム製品の「プロトタイプ機能仕様書」と「検証設計書」を作成することで、IoT プロトタイプシステムを対象とした開発部門による第三者的検証を可能とする。特に検証結果からプロトタイプをそのまま開発を継続し製品版とするための不足項目を指摘する方式を提案する。

(3) IoT システムのプロトタイプの第三者的検証の結果に基づく製品開発の方向付け

IoT システムは新しい利用分野を開拓すると期待されていることから、ユーザ要求の把握および市場に対する製品の投入のタイミングが重要と考えられる。プロトタイプの段階でプロトタイプをそのまま機能・性能を拡張して製品化するまでの距離(完成度)を数値化すると、製品完成時期の予測の確度を上げることができると期待される。またプロトタイプを実際に操作して対象を限定した市場のユーザ評価を行い、評価結果を開発にフィードバックすることにより、ユーザ要求に合った製品仕様を実現できると期待される。このためプロトタイプから製品化までの開発の難易度の数値化、およびプロトタイプのユーザ評価の手法が必要とされる。

以上の要求に対して、プロトタイプの第三者的検証結果をもとに、問題解決型意思決定の手法の一つである Analytic Hierarchy Process (AHP) を用いて数値化することで、プロトタイプから製品完成までの開発過程における、プロトタイプの完成度の客観的な把握が可能となる。また対象となる IoT システムが想定する年齢・性別・人数・経験などを限定したユーザに IoT システムのプロトタイプを操作してもらい、限定ユーザによる評価結果をもとに開発者がユーザ体験を把握することで、ユーザ要求のビジュアライズと対応する開発課題の抽出が可能となる。

このように本研究では、IoT プロトタイプを対象とした、第三者的検証結果に基づくプロトタイプから製品完成までの距離の数値化の方式、およびユーザ評価結果に基づくユーザ要求の明確化と開発課題の抽出の方式を提案する。

1.3 論文の構成

本論文は、6 章から構成される。

2 章では、第三者検証に関する関連研究と課題について述べる。具体的には、ミッションクリティカルなシステムの例として、JAXA で実施されている第三者検証の例を挙げて、第

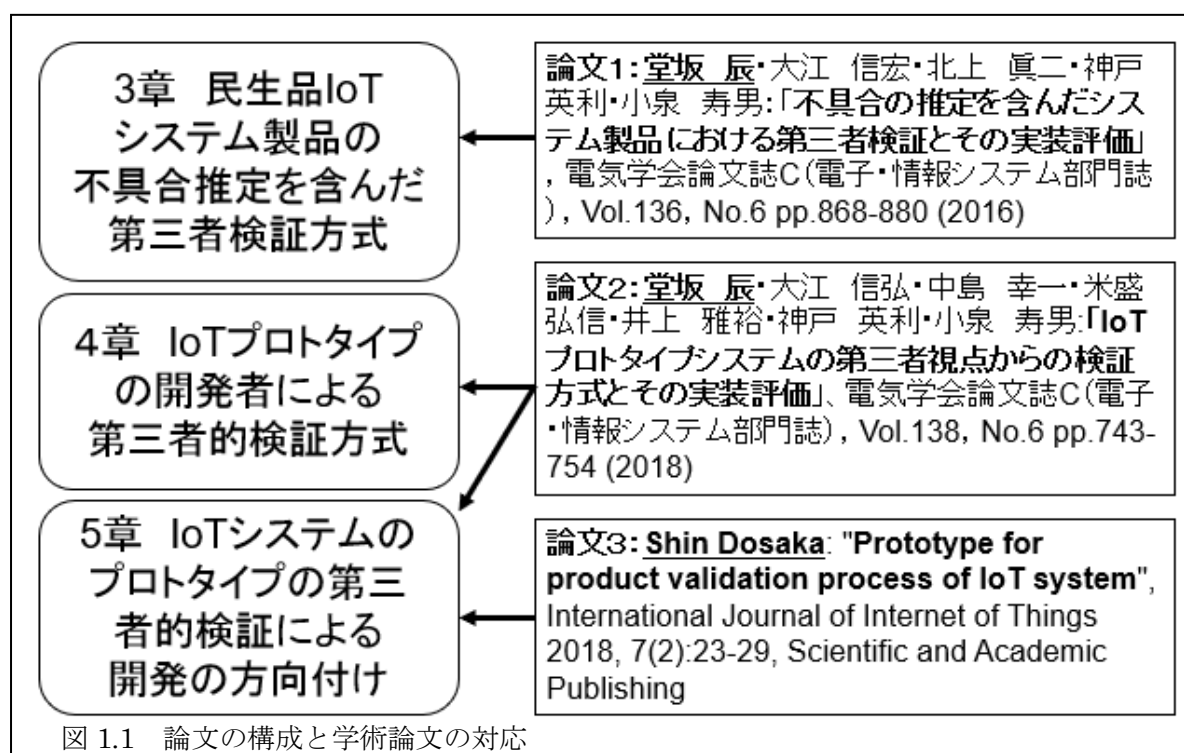
三者検証に関する関連研究を述べる。次に民生品システム製品に対して新たに提案する第三者検証実施の課題を述べる。IoT システム構成の特徴を示し、第三者検証が実施されない理由と、実施しない場合の問題点を述べる。さらに、IoT プロトタイプの目的を示し、プロトタイプに対する第三者的検証が実施されない理由と、実施しない場合の問題点を述べる。

3 章では、民生品 IoT システム製品の開発部門主導による第三者検証方式について提案する。具体的には、民生品 IoT システムにおける第三者検証を実施する場合の、開発者による検証要求仕様書の作成による検証部門での第三者検証実施方式、特に検証部門にて検出した不具合箇所と原因を推定し、開発部門へフォードバックする方式を提案する。

4 章では、IoT プロトタイプの開発者による第三者的検証方式について提案する。具体的には、開発者がプロトタイプを対象に第三者的検証を行う場合の、プロトタイプ機能仕様書と検証設計書に基づく検証の観点と試験の方式を提案する。

5 章では、IoT システムのプロトタイプの第三者的検証結果に基づく開発の方向付けの方式を提案する。具体的には、プロトタイプの第三者的検証結果の数値化によるプロトタイプから製品完成までの距離の算出、およびプロトタイプのユーザ評価に基づく利用目的の明確化と実現方式の明確化の方式を提案する。

6 章では、本研究の成果をまとめるとともに、今後の課題について述べる。
論文の構成と学術論文の対応を、下図に示す。



2. 第三者検証に関する関連研究と課題

2.1 第三者検証の国際規格と実施例

2.1.1 第三者検証の国際規格

第三者検証は、対象製品の開発部門および利用部門とは異なる、第三者のテスト専門部門が特定の基準に基づいてテストを行い、結果判定を行う検証方法である。製品に対する客観的な検証の実施と結果の判定が行える他に、開発者とは異なる経験や試験対象への知識、異なる手法、製品横断的な見方、利用者の視点に基づいて検証を行うことから、開発者では見つけられない潜在欠陥を検出できるという利点がある。図 2.1 に第三者検証の位置づけを示す。国際規格である ISO247645⁽¹⁾では第三者検証は「開発組織から技術面、管理面、および財務面で独立した組織が実施する検証と妥当性確認のことである」と定義されている。

第三者検証は、検証専門部門が開発部門とは独立して開発部門から提供される要件定義書をもとに検証項目を設定し、開発された製品の検証目的に合った試験環境を構築して検証実施とテスト結果判定を行う検証形態である。日本では一般に開発部門と同一企業内の検証部門で実施されるケースが多いが、特定の通信規格に基づく製品などは、規格認定機関の企業が検証を実施し合否を判定する場合もあり、この場合は別会社での検証となる。典型的な第三者検証の運用形態を図 2.2 に示す。第三者検証部門はまず図 2.2(1)に示すように、検証対象となる製品の要件定義書をもとに、検証計画書を作成する。つぎに図中 (2)のように、検証対象製品に対して製品仕様および規格に合致していることを確認する検証テストを実施、得られた合格・不合格の検証結果を(3)の検証結果報告書にまとめ報告する。

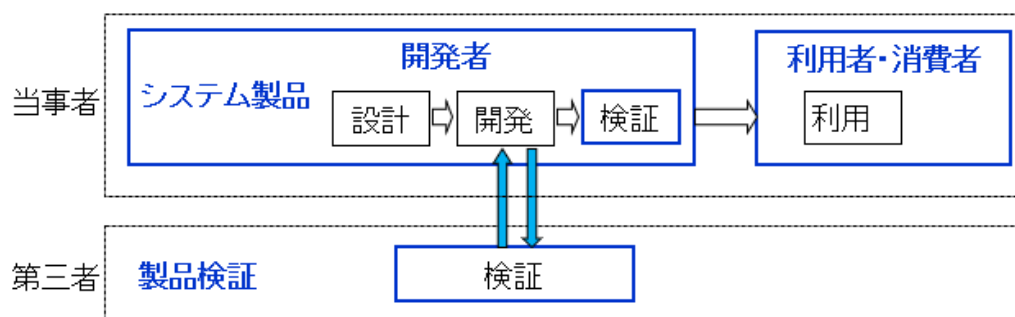


図 2.1 第三者検証の位置づけ

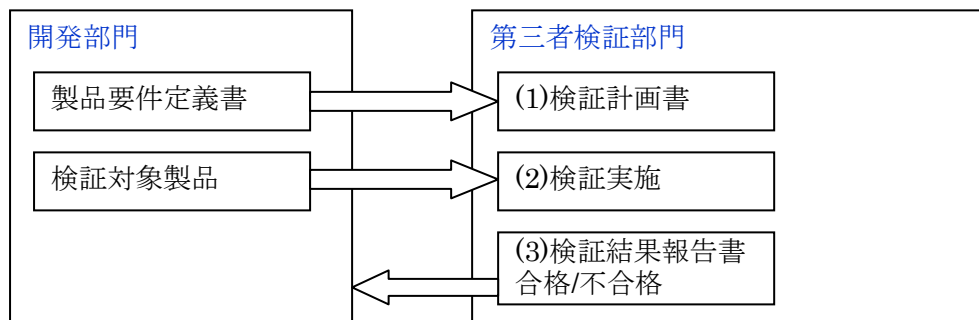


図 2.2 第三者検証の運用形態

2.1.2 ミッションクリティカルな分野でのシステム製品の第三者検証実施例

国際宇宙ステーション（ISS）計画では、参加する各国が第三者検証を実施することを覚書で取り決めている。日本における、人工衛星・探査機・ロケットなど宇宙開発に関するシステム製品は、宇宙航空研究開発機構（JAXA）で開発されているが、これら製品に対して、別会社である有人宇宙システム（株）（JAMSS）により第三者検証が実施されている。

JAXA および JAMSS で定義されている第三者検証のプロセスを図 2.3 に示す。図中 (1) は JAXA で行われる開発プロセスを示し、要求分析・設計・製造・試験のプロセスを経て製品が開発される。図中 (2) はそれぞれの開発プロセスで作成される成果物を表す。これら成果物に対して別組織である JAMSS で (3) 第三者検証が行わる。(4) 検証作業（Verification）では、前工程で決められたとおりに次の工程で成果物が作成されることを確認する。(5) 妥当性確認（Validation）では各開発プロセスの成果物が、要求仕様に対し正しく作成されていることを確認する。

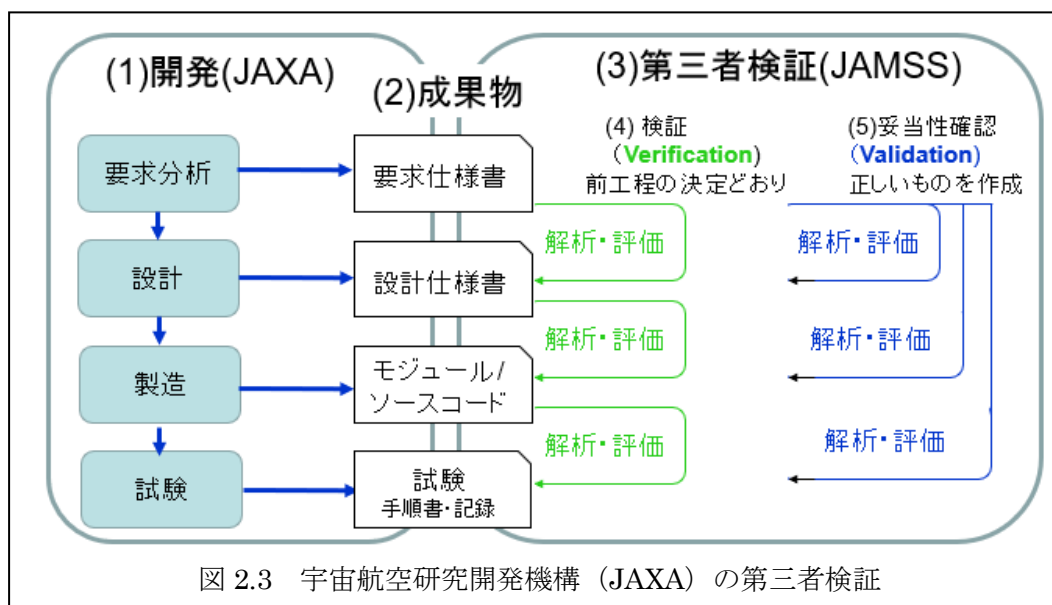


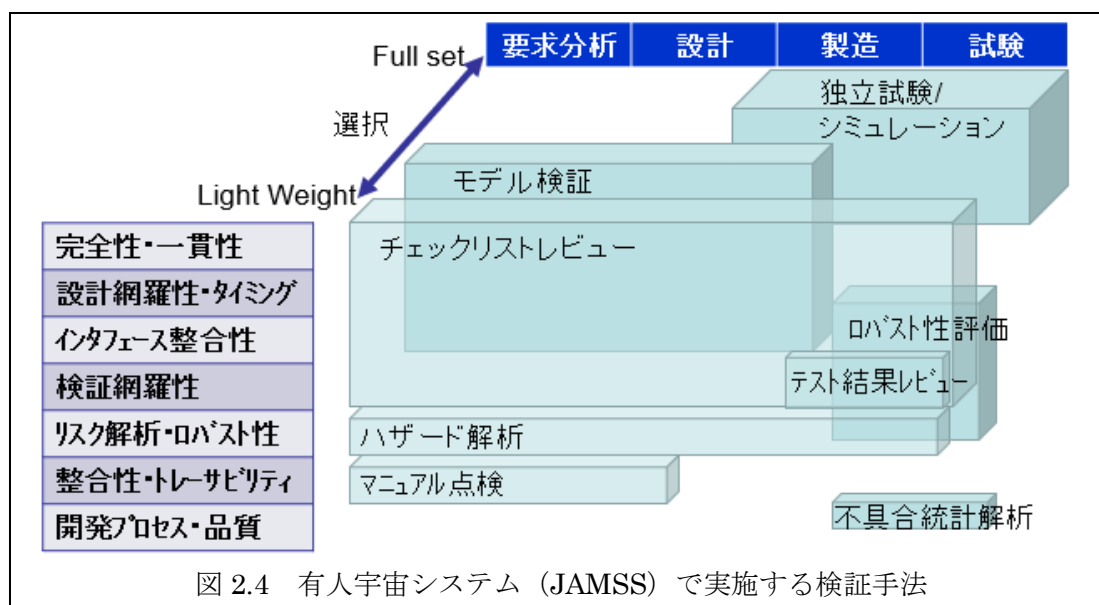
図 2.3 宇宙航空研究開発機構（JAXA）の第三者検証

表 2.1 有人宇宙システム（JAMSS）における開発プロセスと第三者検証作業の対応

開発 / 検証作業		要求分析	設計	製造	試験
開発	解析	○	○		
	チームレビュー	○	○	○	
	ウォークスルー	○	○	○	
	審査	○	○	○	○
	シミュレーション			○	○
	試験			○	○
第三者検証	チェックリストレビュー	○	○	○	○
	モデル検査	○	○	○	
	独立試験			○	○

有人宇宙システム（株）（JAMSS）で実施する第三者検証プロセスを、表 2.1 に示す。表 2.1 の検証作業において、チェックリストをベースとしたレビュー、モデル検査および独立試験が第三者検証に対応している。チェックリストをベースとしたレビューは、開発の要求分析・設計・製造・試験の全てのプロセスに対して実施する。モデル検査は、要求分析あるいは設計から導かれた論理モデルが状態遷移の一貫性や網羅性など形式仕様を満足するかどうかを検証する手法で、開発の要求分析・設計・製造のプロセスに対して実施する。開発とは独立して実施する独立試験は、製造および試験のプロセスに対して実施される。

有人宇宙システム（JAMSS）では、前述のチェックシートレビュー・モデル検証・独立試験の他に、検証の深さによって図 2.4 に示す様々な検証手法を選択して実施している。



2.1.3 民生品システム製品の第三者検証の現状

民生品システム製品では、Universal Serial Bus (USB)⁽²⁾や Bluetooth⁽³⁾などの規格認定、ハードウェアやコンパイラソフトウェアの検証など単体製品を対象とした検証が行われている。組み込みシステム製品について製品の高度化に伴いソフトウェアで実現される領域が増大したことから、第三者検証が求められるようになってきており、国際基準に基づくセキュリティ評価基準である ISO15408⁽⁴⁾や、HEMS(Home Energy Management System)の標準プロトコルである ECHONET Lite⁽⁵⁾などの第三者検証が実施されている。しかしながら、これらはシステム製品における単体機能の第三者検証であり、ミッションクリティカルな分野で実施されているようなシステム全体の第三者検証は、開発期間および開発コストの制約から、民生品システム製品では実施されていない。

2.2 第三者検証およびIoTシステムに関する関連研究

第三者検証の調査によると⁽⁶⁾、システム製品では、宇宙・航空や自動車の自動運転など、公共性が高く人命にかかわるミッションクリティカルな分野で第三者検証の手法が用いられているが、民生品システム製品の分野での第三者検証は実施されていないことが報告されている。

システム製品では、宇宙・航空や自動車の自動運転など、公共性が高く人命にかかわるミッションクリティカルな分野で第三者検証の手法が用いられている。宇宙航空研究開発機構(JAXA)では、システムの第三者検証活動の定量的評価を行った研究「不具合履歴に基づくソフトウェア IV&V 活動の定量的見えるか手法」⁽⁷⁾、第三者検証の体系的な取り組みを示した研究「Strategy to enhance IV&V activity in JAXA」⁽⁸⁾が行われている。さらに JAXA 出版社の「IV&V ガイドブック」⁽⁹⁾では、人工衛星の開発において採用されている第三者検証の手法が報告されている。

民生品分野では、ハードウェアを対象とした第三者検証は、製品の設計検証に対応したプログラム TMP(Test and Maintenance Program)を準備し検証に適用している事例が報告されている⁽¹⁰⁾。ハードウェア仕様に基づいてテストプログラムを準備し、単体機能検証、組み合わせ機能検証、RAS(Reliability, Availability, Serviceability)機能検証、さらに性能目標達成していることを検証している。

ソフトウェアについての第三者検証では、ソースコードを対象とした静的解析ツールおよび動的解析ツールを用いた事例⁽¹¹⁾が報告されている。

上記の民生品の第三者検証に関する研究報告は、いずれも単体機能に関する第三者検証であり、システム製品全体に対する第三者検証はほとんど研究されていない。

IoT システムについては、Information-technology Promotion Agency(IPA)が「つながる世界の品質確保に向けた手引き」⁽¹²⁾により、IoT システムの開発および運用における妥当性確認と検証について考慮すべきポイントを網羅的に述べている。しかしながら、IoT システム製品を対象とした第三者検証の実施方式については、述べられていない。

2.3 民生品 IoT システム製品に対する第三者検証実施の課題

2.3.1 本研究の位置づけ

Internet of Things (IoT)システムは独自に製品化され管理されている複数の製品が、オープンなインターフェースのネットワークを介して繋がりシステムを構成するという特徴がある。IoT システムの適用分野が社会インフラから民生品まで広がり、民生品のデバイスもインターネットと繋がることで機能が広がり、制御が自動化されてきている。IoT システムは構成要素が多くそれらは独自に製品化されていることから、開発者がシステム全体を把握することは困難になっている。このため、IoT システムに障害が発生した場合、原因の切り分けや責任分解が難しく、障害解決に時間がかかることが予想される。このことから、民生品 IoT システム製品についても、異なる評価軸から検証を行い多角的に品質推定精度を上げる、第三者検証を実施する必要があると考える。そこで、図 2.5 に示すように、開発期間やコストが重要視される民生品 IoT システム製品に対して、開発者の視点で第三者検証を実施するという、新たな第三者検証方式を提案する。

第三者検証の実施は、システム製品に対する第三者検証とプロトタイプに対する第三者的検証に分けられる。図 2.6 に本研究の目的の位置づけを示す。

- (1) 民生品 IoT システム製品の不具合推定を含んだ第三者検証方式
- (2) IoT プロトタイプの開発者による第三者的検証方式
- (3) IoT システムのプロトタイプの第三者的検証による開発の方向付け

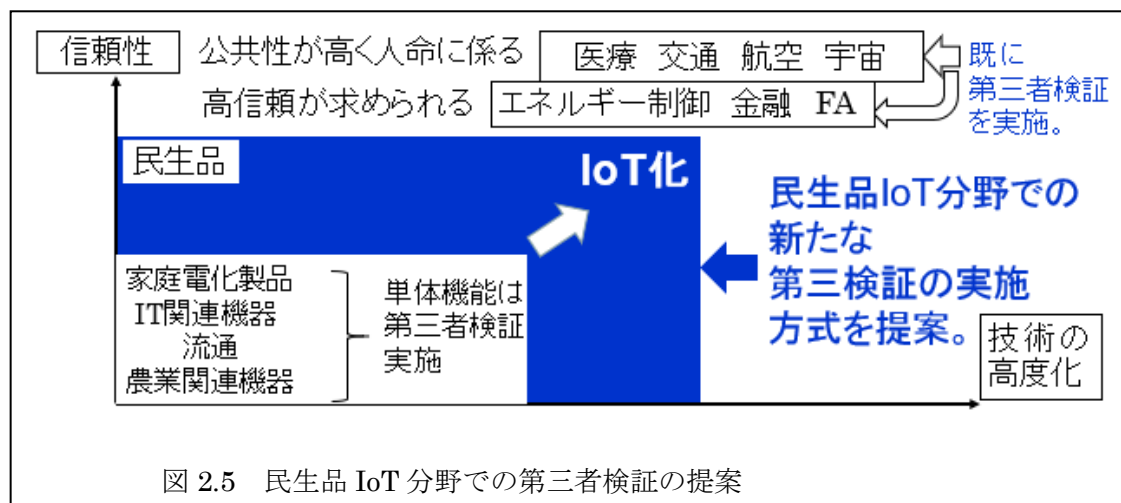


図 2.5 民生品 IoT 分野での第三者検証の提案

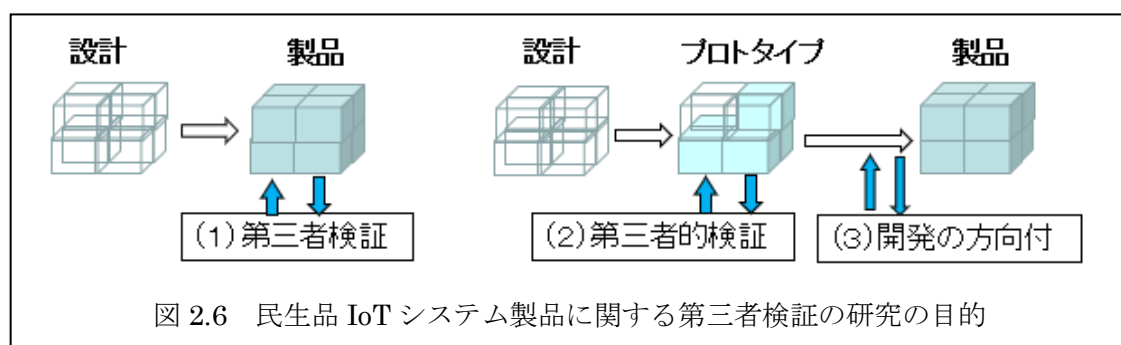


図 2.6 民生品 IoT システム製品に関する第三者検証の研究の目的

2.3.2 IoT システム構成の特徴と民生品 IoT システム

インターネットの普及は、大量の情報へのアクセスを可能として、IoT システムではあらゆる「機器」がインターネットへ接続される社会が生まれようとしている。機器がインターネットに組み込まれることで新しい価値が創出され、新しいビジネスが創出できると期待される。IoT システムでは、機器や人間やフィールドから集めたセンサーデータを、エリアネットワーク、ゲートウェイ、アクセスネットワークを介してクラウドに送信する。クラウド上では、集められたデータの可視化や分析などの処理を行い、その結果をアクチュエーターにフィードバックを行い、あるいは制御端末へ送信するという動作を、人手を介さずに行う。IoT システムは以上のような技術が集約されたものである。IoT システムの一般的な構成を図 2.7 に示す。図に示すように IoT システムは、(1)デバイス、(2)ゲートウェイ、(3)クラウド上のアプリケーション、(4)制御端末という複数のサブシステムで構成される。物理的(Physical)なデバイスと仮想的(Cyber)なクラウドの世界が、オープンネットワークインタフェースを用いて繋がって IoT システムが構成される⁽¹³⁾⁽¹⁴⁾ と言える。

IoT システムは、独自に製品化され管理されている複数の製品が、オープンなインタフェースでネットワークを介して繋がり一つのシステムを構成するという特徴がある。このため、開発者が IoT システムを構成する全てのサブシステムの設計思想を把握することは困難である。

IoT システムの適用分野は、エネルギー制御という社会インフラや工場の生産ラインの制御から、ウェアラブル端末・スマートハウス・アシスタントロボから農業や物流など、

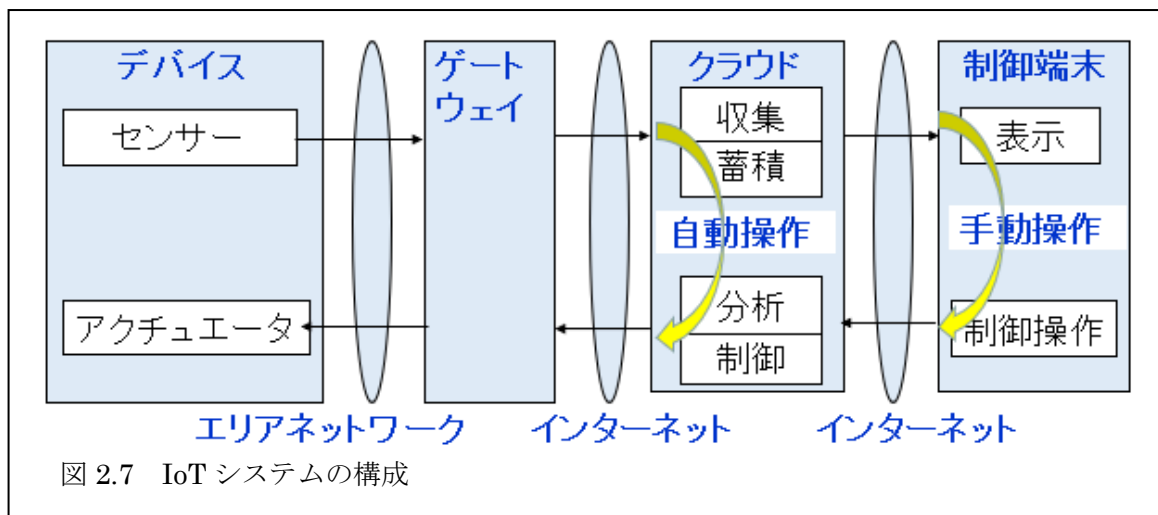


図 2.7 IoT システムの構成

ウェアラブル端末 子供位置情報 モニタリング 	スマートハウス 家電製品の 遠隔操作 	アシスタントロボ 会話しながら 生活を支援 	セキュリティ スマートキー 
小売り 店舗内販売 分析 	シェアリング バイクシェア 	農業 水耕栽培 	物流 配送荷物の 無人配達 

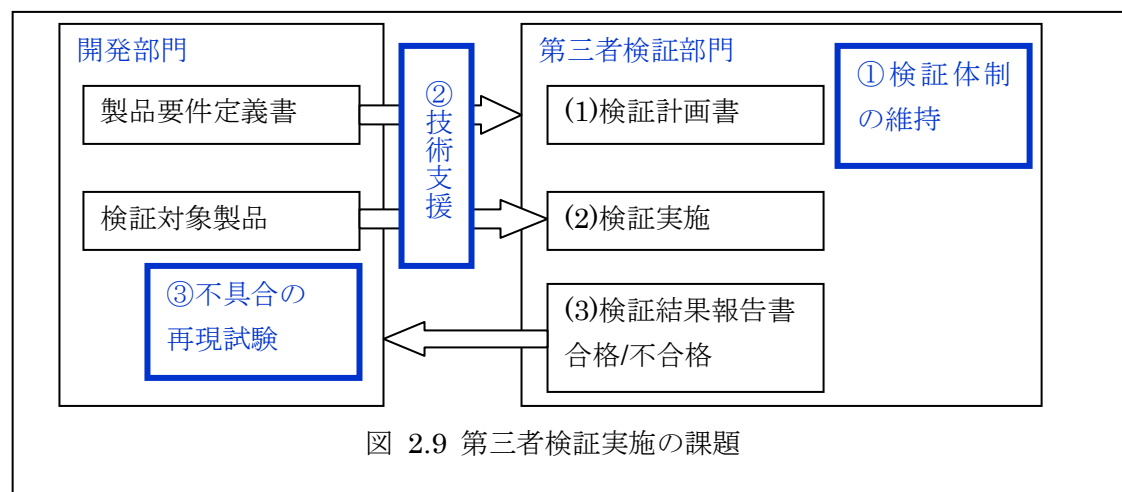
図 2.8 民生品 IoT システムの例

図 2.8 に示すように、民生品分野まで広がっている。このため、民生品 IoT システム製品分野では、第三者検証が必要であると考ええる。

2.3.3 民生品 IoT システム製品の第三者検証実施の課題

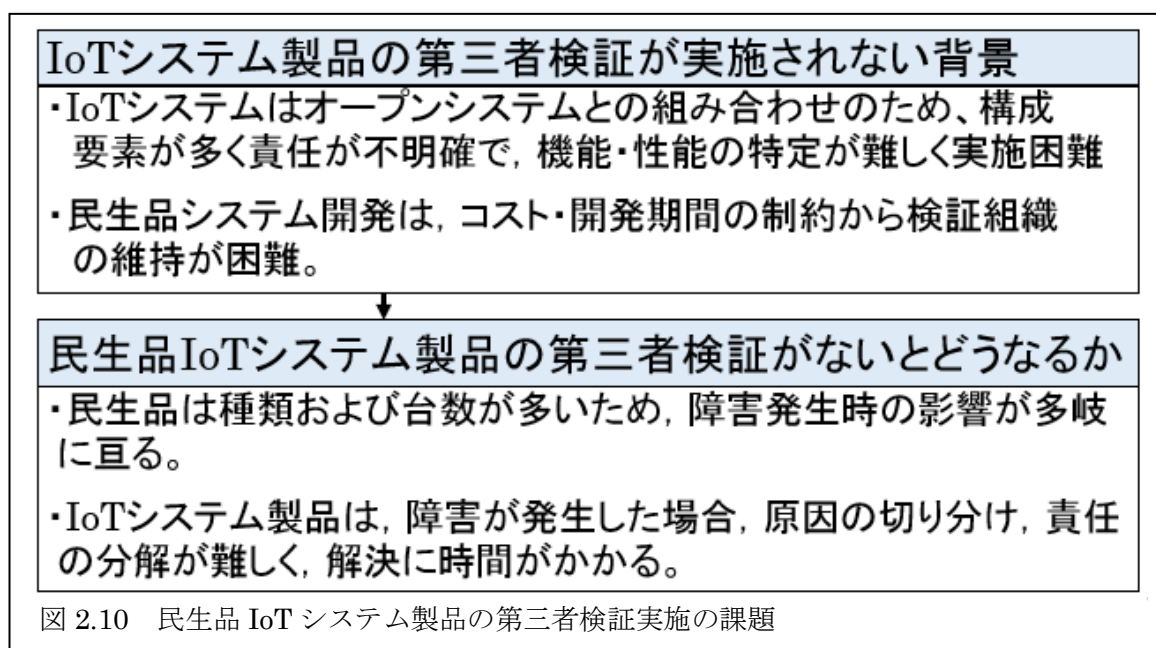
IoT システムは、独自に製品化され管理されている複数の製品が、オープンなインタフェースでネットワークを介して繋がり一つのシステムを構成するという特徴がある。このため、いわばブラックボックス製品として他社製品を利用し、ネットワークサービスやクラウド上でのアプリケーションの実行など管理責任が異なる製品も含めてシステムを構成している。開発部門は、開発製品の動作環境を規定しその範囲で開発製品が設計どおりに動作することのテストを実施するが、システム製品の導入環境・運用者・利用者が開発者の想定を超える可能性があることから、開発者とは異なる視点による第三者検証の実施が求められる。

民生品の分野のシステム製品は、検証対象のシステム構成・機能・性能・利用者が様々であることから、第三者という「他の組織」で検証を行うためには、テスト環境構築・テストケースの設定・利用者の明確化という検証時に必要とされる項目について、開発部門から検証部門へある程度の技術的介入や協力が必要となると考えられる。これは、開発部門に負担がかかることが予想され、第三者検証の導入への障壁ともいえる課題となる。図 2.9 に、民生品 IoT システムの第三者検証を実施する場合の課題を示す。図中①は第三者検証部門の維持のコストが発生することを示す。②は開発部門から検証部門への技術支援の



負担が発生することを示す。③は検証部門で検出した障害を解析するために、開発部門で再現試験が必要であることを示す。これらはコスト増となるとともに開発期間の延長にもなることから、第三者検証の導入時の課題と考えられる。

第三者検証で検出した障害について、検証部門で不具合の発生箇所と原因の推定を可能として、開発部門での不具合原因の究明作業の軽減を図ることが考えられる。しかしこの場合、第三者検証の基本は「技術面・管理面・財務面での独立組織による検証」であることから、不具合箇所の特定のためシステム製品の内部を意識することは、第三者検証の独立性や客観性の確保といった点が犠牲になり、また不具合箇所の特定に時間を使う、という難点がある。以上をまとめて、図 2.10 に、民生品 IoT システム製品の第三者検証実施の課題を示す。



IoTシステムは新規性の高い開発要素があり、新しい市場を開拓すると期待される。このため、プロトタイプを作成し、客観的に確認(検証)することが有効。

- (1) 主要機能の実現性の確認を目的とし、廃棄する。
- (2) 主要機能の実現性の確認後も、継続開発し製品化する。

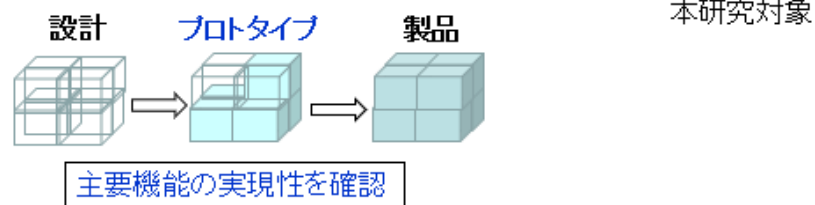


図 2.11 IoT システム開発におけるプロトタイプの目的

2.3.4 民生品 IoT システム製品開発におけるプロトタイプの目的

IoT システムは、新規性の高い開発要素があり、さらに自社の開発製品以外に、独自に製品化され内部処理をブラックボックスと見なす他のシステムを自社製品に繋いで一つのシステムとして機能させることから、プロトタイプを開発して開発システムの主要な仕様の実現性を確認することは有効である。 図 2.11 にプロトタイプの位置づけを示す。

プロトタイプを、開発の優先度やユーザ要求仕様の確認のためのマイルストーンとして位置づけ、客観的に評価を行うことで、製品化までに行うべき事柄が明確になり、製品化までの開発期間の短縮を図ることができる。また、プロトタイプの段階でプロトタイプをそのまま機能・性能を拡張して製品化することの難易度を数値化して表現できると、製品化計画の確度を上げることができると期待される。さらにプロトタイプを実際に操作して対象を限定した市場のユーザ評価を行い、評価結果を開発にフィードバックすることによりユーザ要求に合った製品仕様を実現できると期待される。

本論文では開発者が第三者検証を模して、プロトタイプに対して客観的に実施する検証を「第三者的検証」と表す。

なおプロトタイプには、基本機能の確認を目的として確認実施後は破棄するものと、全体機能の確認を目的として、継続開発を行い製品化するものの 2 種類がある。本研究で第三者検証の対象とするのは、後者の継続開発を行い製品化を目指すプロトタイプである

2.3.5 民生品 IoT プロトタイプシステムの第三者的検証実施の課題

プロトタイプを開発の進捗をモニタリングするマイルストーンとして位置づけ、機能達成度について客観的評価を行うことで、プロトタイプからそのまま開発を続けて製品版とす

IoTプロトタイプ^①の第三者的検証が実施されない背景

プロトタイプ^①の客観的評価の基準と手法がない。



IoTプロトタイプ^①の第三的検証がないとどうなるのか

- ・プロトタイプから開発を継続して製品化するための不足部分が不明で、製品完成に時間がかかる。
- ・継続開発して製品化できるか、難易度が把握できず、製品の投入タイミングが確定できない。
- ・設計に係るユーザ評価の指摘事項は十分反映できない。

図 2.12 IoT プロトタイプの第三者的検証実施の課題

るために必要な不足部分の指摘を行うことが可能となり、製品化までに行うべき開発項目が明確になり開発期間を短縮できるという効果が期待できる。しかし、プロトタイプをそのまま継続開発して製品化できるかという観点での、プロトタイプに対する第三者検証の方式や評価項目はない。

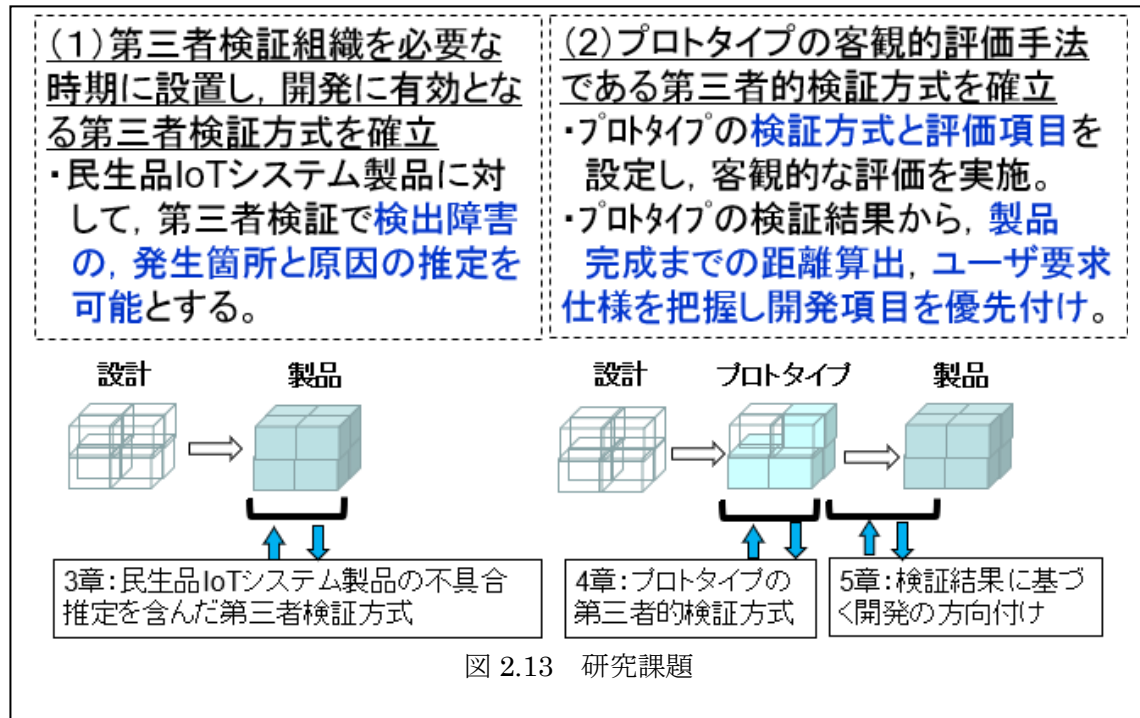
また、IoT システムは新しい利用分野を開拓すると期待されていることから、ユーザ要求の把握および市場に対する製品の投入のタイミングが重要と考えられる。しかし、ロトタイプから製品化までの開発の難易度の数値化、およびプロトタイプのユーザ評価の方式はない。

以上をまとめて、図 2.12 に、民生品 IoT プロトタイプの第三者的検証実施の課題を示す。

2.3.6 課題と研究の目的

民生品システム製品では、USB・Bluetooth・ECHONET Lite などの単体機能については第三者検証が実施されているが、システム製品全体の第三者検証は、開発期間とコストの制約から実施されていない。あらゆる「モノ」がインターネットへつながる社会を目指す IoT は、モバイル技術・無線通信・クラウド技術の発達により IoT 基盤が整い、社会インフラだけでなく一般消費者との係りが深い民生品分野での利用が始まっている。IoT システム製品は、自社の開発製品以外に一つ一つは独立して開発された他社製品をオープンインタフェースのインターネットで繋いでシステム化するという特徴がある。このため、IoT システム製品では第三者検証が必要と考えられるが、図 2.10 に示したような課題がある。また IoT システム開発にはプロトタイプの開発が有効であり、プロトタイプに対して第三者検証を実施して客観的評価に基づき、プロトタイプから製品完成までの実現性を確認し、さらに開発項目の優先付けとユーザ要求仕様の確定を行うという期待があるが、プロトタイプの第三者的検証実施には図 2.12 に示した課題がある。

本研究の課題は、図 2.13 に示すように、民生品 IoT システム製品に対する、第三者検証組織を必要な時期に設置し、開発に有効となる第三者検証方式の確立、およびプロトタイプの客観的評価手法である第三者的検証方式の確立であり、次章以降でこれらの方式の提案を行う。



3. 民生品 IoT システム製品の不具合推定を含んだ第三者検証方式

本章では、民生品分野の IoT システム製品に対する、開発者部門が主導する、検証部門での第三者検証の実施のための方式を提案する。民生品 IoT システム製品開発は、開発期間やコストが重視されることから開発部門とは異なる検証組織の維持は困難である。また IoT システムは自社製品だけでなく、独立して開発された他社製品をシステムを構成するサブシステムとして利用し、オープンインタフェースのネットワークで繋いでシステム化していることから、システムとしての機能・性能・責任範囲の特定が開発者以外の第三者では難しい。さらに開発部門が他部門へ検証を依頼する場合には、検証部門への技術支援のため開発部門の負荷が増加するという課題もある。

本研究では、開発者が IoT システムを構成するサブシステムを明らかにし、予めインタフェーストレースとログを採取し障害解析を可能とするテスト環境を規定する検証要求仕様書を作成することで、検証部門を必要時に設置して第三者検証の実施を可能とし、検証部門においては検出した障害の発生箇所と原因の推定を可能とする検証方式の提案を目的とする。具体的には、「検出障害の推定方式をもつ第三者検証方式」および「検出障害の発生箇所と原因の推定方式」を提案する。

3.1 はじめに

第三者検証は、第三者のテスト専門部門が潜在欠陥の検出を目的にテストを行い、特定の基準に基づいて結果判定を行う検証方法である。製品検証に対する客観的な検証結果の判定が行える他に、開発者とは異なる経験や試験対象への知識、異なる手法、製品横断的な見方、利用者の視点に基づいて検証を行うことから、開発者では見つけられない不具合を検出できるという利点がある。ソフトウェアの分野では **Independent Verification and Validation(IV&V)** として「開発組織から技術面、管理面、及び財務面で独立した組織が実施する検証と妥当性確認のことである」と定義されている⁽¹⁾（ここでのソフトウェアとはシステムエンジニアリングのプロセスを意味している）。現状の第三者検証は、USB などの規格認定やコンパイラソフトウェアの検証など単体製品を対象としたものが多い。また組み込みシステム製品についても、製品の高度化に伴いソフトウェアで実現される領域が増大したことから、国際基準に基づく認証による検証が求められるようになってきている⁽²⁾。しかし第三者検証においては、試験項目に対する試験結果の合否判定がなされるが、検出された不具合の発生箇所や原因の指摘はなされない。近年、複数のサブシステムをオープンなインタフェースを用いて組み合わせたシステム製品が注目されている。これらシステム製品はそれぞれ独自に製品化され管理されているサブシステムを組み合わせでシステム化することから、対象製品の全体の把握が難しく、第三者検証事例は殆ど報告されておらず、さらに第三者検証によりサブシステム単位で不具合箇所とその原因を推定する手法が必要とされている。

本論文では、システム製品を対象として、不具合の発生箇所や原因の推定を行う第三者検証手法を提案する。本提案の手法は、試験結果の異常値からサブシステム単位の不具合箇所と不具合原因の推定を行うものである。この推定は以下の2つのプロセスで構成される。

- ・サブシステム間のインタフェーストレースから不具合発生の子システムを切り分け
- ・サブシステムの動作環境ログとインタフェーストレースから不具合原因を推定

本提案の第三者検証手法を、多数の世帯においてスマートフォンを用いて家庭内の電化製品を屋外から操作するシステムを対象として適用し、不具合箇所と原因の推定手法の有効性を確認した。

3.2 現状の第三者検証とその課題

3.2.1 第三者検証の運用形態

第三者検証は検証部門が第三者検証は検証専門部門が、開発部門とは独立して開発部門から提供される要件定義書をもとに検証項目を設定し、開発された製品の検証目的に合った試験環境を構築して検証実施とテスト結果判定を行う検証形態を指す。日本では一般に開発部門と同一企業内の検証部門で実施されるケースが多いが、特定の通信規格に基づく製品などは、規格認定機関の企業が検証を実施し合否を判定する場合もあり、この場合は別会社での検証となる。典型的な第三者検証の運用形態を図 3.1 に示す。第三者検証部門はまず図 3.1 の(1)に示すように、検証対象となる製品の要件定義書をもとに、検証計画書を作成する。つぎに図中 (2)のように、試験対象製品に対して製品仕様および規格に合致していることを検証するテストを実施、得られた合格・不合格の検証結果を(3)のレポートにまとめる。

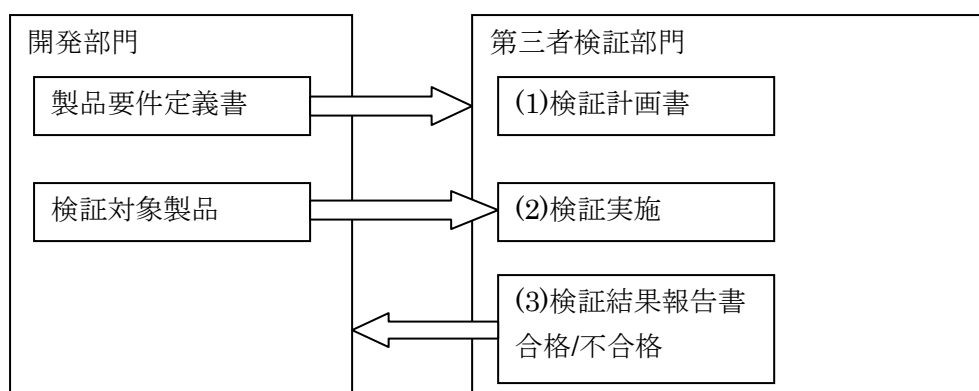


図 3.1 第三者検証の運用形態

3.2.2 第三者検証でシステム製品の不具合箇所と原因を推定するための課題

ハードウェアを対象とした第三者検証では、製品の設計検証に対応したプログラムTMP(Test and Maintenance Program)を準備し検証に適用している事例が報告されている⁽³⁾。ハードウェア仕様に基づいてテストプログラムを準備し、単体機能検証、組み合わせ機能検証、RAS(Reliability, Availability, Serviceability)機能検証、さらに性能目標達成していることを検証している。ソフトウェアについての第三者検証では、ソースコードを対象とした静的解析ツールおよび動的解析ツールを用いた事例⁽⁴⁾および第三者検証の定量的評価手法⁽⁵⁾が報告されている。要件定義・設計工程における標準プロセスに基づく設計書の点検、製造工程・テスト工程においてツールを用いた成果物の検証、検出した不具合の整理と分類を行っている。

しかしながら、システム製品を対象とした検証、さらに検出した不具合の推定手法については報告されていない。この理由として、システム製品はネットワークを含む複数のサブシステムを結合して構成されることから、テスト環境の作成およびテストケースの設定、さらに検出した不具合切り分けの一般化が難しいことが挙げられる。

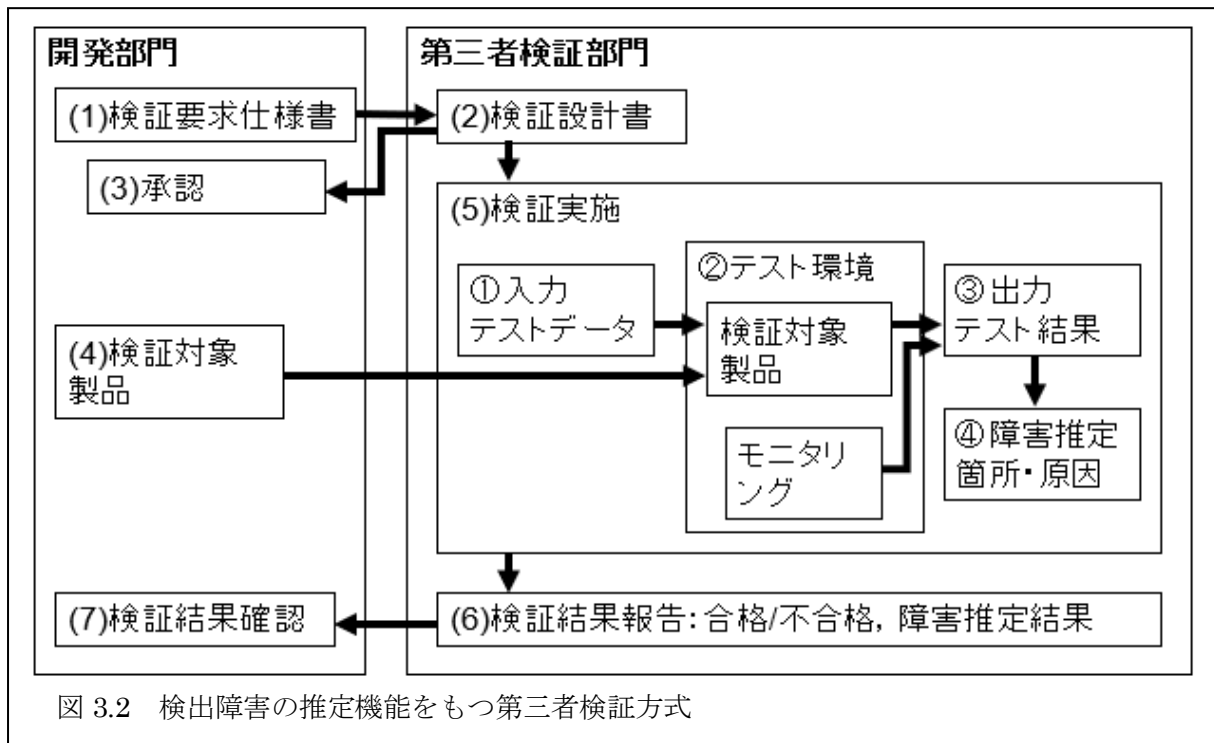
3.2.3 ミッションクリティカルな分野で適用される検証の手法

ミッションクリティカルなシステム製品に対して、次のような検証手法が適用されている。STAMP⁽⁶⁾(System Theoretic Accident Model and Processor)は、安全制御の視点から制御コンポーネント間の連携動作に障害原因を見つける、事故原因の解析手法である。STPA⁽⁶⁾(STAMP based Process analysis)は STAMP の手法に基づいて、設計時点で発生が予想される事故要因を解析する設計手法である。JAXA 出版社の IV&V プロセス⁽⁷⁾は、要求設定・設計・製造からシステム試験までの全ての段階でレビューあるいはテストを実施し、開発成果物を妥当性と有効性の観点で解析評価する手法である。このうち設計における IV&V に関しては、開発が作成する設計書を対象に、上位設計書に基づいて設計レビューが行われる。AADL⁽⁸⁾(Architecture Analysis & Design Language)エラーモデルは、システムの機能要素間の故障状態の波及性をシミュレーションし、機能要素やインタフェース上の不具合を検出する設計手法である。これらの手法と本方式の比較については 3.6 節で述べる。

3.3 システム製品の不具合の推定を含む第三者検証手法の提案

3.3.1 検出障害の推定機能をもつ第三者検証方式

システム製品の不具合箇所と原因の推定を可能とする第三者検証方式の流れを図 3.2 に示す。開発部門は、試験対象となるシステム製品が実現する機能とユーザインタフェースおよび準拠する規格や動作条件に加えて、システムの構成をサブシステム単位に機能とインタフェースを記載、さらにインタフェーストレースとログを採取できるテスト環境を記述した(1)検証要求仕様書を第三者検証部門へ提供する。第三者検証部門は (1)検証要求仕様



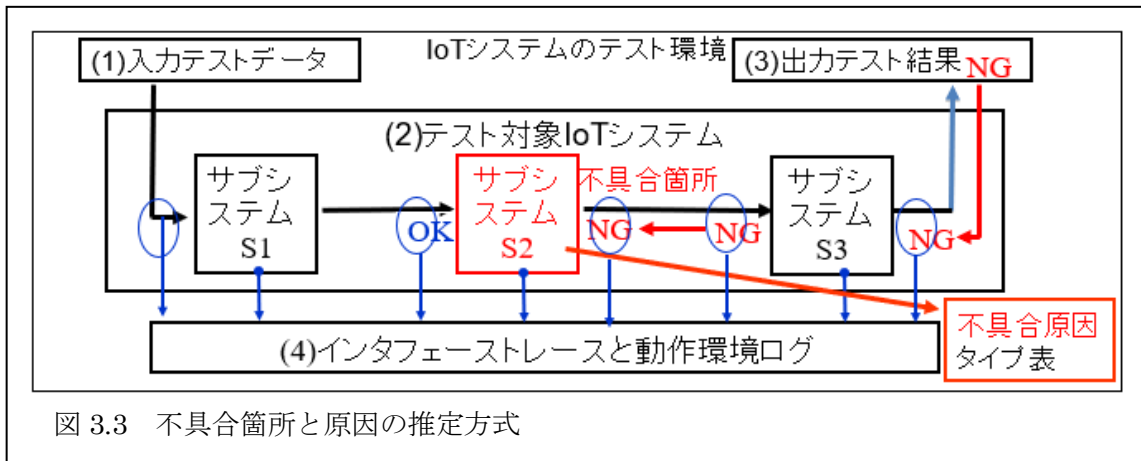
書をもとに、検証項目と合否判定基準を記載した(2)検証計画書を作成し開発部門へ承認依頼を行う。開発部門では(3)承認を行うとともに(4)検証対象製品を第三者検証部門に提供する。(5)第三者検証部門での検証実施の流れを示す、①テスト入力データを作成し、②検証対象製品をサブシステムに分けてインタフェーストレースとログをモニタリングする機能を備えたテスト環境へ投入、③出力されたテスト結果とモニタリングデータをもとに、④推論テーブルを用いて不具合箇所と不具合原因を推定する。第三者検証部門では(6)検証結果と検出した障害についての推論結果を合わせて開発部門へフィードバックを行い、開発部門では(7)検証結果の確認を行う。

3.3.2 検出障害の発生箇所と原因の推定方式

本節では、上記検証方式の流れにおいて、テスト環境、テストデータを用いて、検証結果から検出障害の発生箇所および不具合原因の推定を行う方式を以下に論じる。

3.3.2.1 異常値から不具合箇所と原因を推定

上記流れの図 3.2 (5) 検証実施の内容について、図 3.3 不具合箇所と原因の推定方式にその詳細を示す。図中のテスト環境では、(1) 入力テストデータとして、システム製品が仕様どおりに正しく動作するか、仕様を超える条件で不正な動作をしないか、仕様で想定しない環境においても期待通りに動作するかという 3 つの観点で試験データを準備し、これを(2)テスト対象 IoT システムへの入力データとする。(2)テスト対象 IoT システムを機能単位にサブシステムへ分けて、サブシステム単位に処理結果の判定が可能となるように入出



ラインタフェーストレースと動作環境ログ機能を設定する。(3)はテスト結果、(4)はサブシステム間に設定したインタフェーストレースとサブシステムの動作環境ログをモニタリングする機能を示す。

図中(2)(3)(4)のテスト環境および(1)の入力テストデータの詳細は後述する〈3.3.2.2〉および〈3.3.2.3〉で述べるが、これらを通して得られたテスト結果とモニタリングしたデータが推論の入力情報となる。テストケースの特定の入力データ(1)に対して、システムの出力結果(3)を判定し結果が期待値と異なる異常値であった場合、不具合箇所と不具合原因を推定する手順を(P1)、(P2)、(P3)、(P4)の順に示す。

(P1) 検出した異常値から不具合箇所を切り分ける

サブシステム間に設定したインタフェーストレースのモニタリング機能(4)を用いて、検出された異常値からサブシステム連携の連鎖を遡って異常原因となるサブシステムを推定する。これが不具合原因箇所の候補となる。図 3.3 では、サブシステム S3 の出力であるテスト結果が異常値(NG)を示している。サブシステム S3 の入力を調べると異常値(NG)を示している。インタフェーストレースを遡って、サブシステム S3 の入力となるサブシステム S2 の出力を調べる。するとサブシステム S2 の出力結果が異常(NG)であることがわかる。さらにサブシステム S2 の入力を調べると S2 の入力は正常(OK)である。サブシステム S2 の入力は正常であり、S2 の出力結果が異常であることから、サブシステム S2 の処理あるいは動作環境に不具合があると考えられる。この結果からサブシステム S2 が不具合箇所の候補となる。

(P2) 異常値を一般的な不具合現象へ変換

テストで検出される異常値・一般的な不具合現象・対応する不具合原因を、本方式では不具合原因タイプ表 3.1 として分類する。不具合原因箇所として推定したサブシステムにおいて発生する異常値は、テストケース毎に現れる形式が異なることから、この異常値を一般的な不具合現象に変換する。異常値と不具合現象の対応は表 3.1 において次のようになる。表中(1)結果異常の異常値では、一般的な不具合現象として(1-1)処理結果不具合が対応する。

表 3.1 不具合原因タイプ表

NG結果	不具合現象	不具合現象	参照データ
(1) 結果異常	(1-1) 処理結果不具合	(a) 処理誤り	トレース
		(b) インタフェース不一致	
(2) タイムアウト	(2-1) イベント不達	(c) 外乱	トレース
		(d) リトライタイミング不一致	
		(e) 応答遅れ	
	(2-2) エラーイベント通知	(f) インタフェース不一致	トレース, ログ
		(g) リソース不足(CPU, メモリ)	
(3) 処理遅れ	(3-1) イベント通知の遅延	(h) 他のサブシステム処理待	ログ
		(i) 非同期イベント発生	
		(j) 同時並行処理の誤り	
		(k) 外乱	トレース
	(3-2) 間欠的イベント通知	(l) リトライのタイミング不一致	
		(m) 応答遅れ	
	(4-1) 仕様外の処理	(n) イベントの削除忘れ	トレース

表中 (2) タイムアウトの異常値では、(2-1)イベント不達と(2-2)エラーイベント通知の2種類の不具合現象が対応する。(3)遅延の異常値では(3-1)イベント通知の遅延と(3-2)間欠的イベント通知の不具合現象が対応する。さらに (4) 仕様外の値という異常値では、(4-1)仕様外の処理という不具合現象が対応する。

(P3) 不具合現象から不具合原因を特定

表 3.1 に示す不具合原因タイプをもとに、インタフェーストレースおよび動作環境ログから、不具合現象に対応する不具合原因の候補を選択する。不具合現象と不具合原因の対応は表 3.1 において次のとおりとなる。表中(1-1)処理結果不具合現象に対応する原因候補は、(a)処理誤りと、(b)インタフェース不一致が対応する。(2-1)イベント不達不具合現象に対応する原因候補は、(c)外乱、(d)リトライタイミングの不一致、(e)応答遅れが対応する。(2-2)エラーイベント通知不具合現象に対する原因候補は、(f)インタフェース不一致が対応する。(3-1)イベント通知の遅延不具合現象に対応する原因候補は、(g)CPU・メモリなどの計算機リソース不足、(h)他のサブシステム処理結果待ち、(i)非同期イベントの発生、(j)同時並行処理の誤りが対応する。(3-2)間欠的イベント通知不具合現象に対応する不具合の原因候補は、(k)外乱、(l)リトライタイミングの不一致、(m)応答遅れが対応する。(4-1)仕様外の処理という不具合現象に対応する原因候補は、(n)イベント通知要求の削除忘れが対応する、というように不具合原因の候補が考えられる。インタフェーストレースと動作環境ログを

もとに、これらの候補から不具合原因を選択するという推論方式をとる。システムをサブシステムの連携として捉え、さらにサブシステムの連携は時間概念で連結されていると考えることができることから、不具合現象が発生した時刻をキーとして相当するトレースおよびログを参照することがポイントとなる。

(P4) 不具合原因箇所と不具合原因の推定の確認

ある特定の不具合原因を選択すると、こんどは選択した不具合原因の候補をもとにサブシステム連携の連鎖を順方向に辿って、不具合箇所の不具合原因から不具合現象さらに異常値の発生を確かめることで、不具合原因と不具合原因箇所の推定の正しさを確かめる。ここで想定した不具合原因から異常値を論理的に導けない場合は、再度表 3.1 の不具合原因候補の選択あるいは不具合現象の変換へ戻り、推論を繰り返す。

3.3.2.2 サブシステム間のトレースと動作環境ログのモニタリング機能をもつテスト環境

システムの動作はサブシステムの連携として表すことができる。サブシステム間に設定したインタフェーストレースと動作環境をモニタリングするテスト環境を図 3.4 に示す。図中サブシステム S2 は①入力、②処理、③出力、さらに④動作環境を加えて 4 つの要素をもつ機能単位として表すことができる。すると入力データ、処理および動作環境に問題がある場合、それぞれのサブシステム間のトレースとサブシステムの動作環境ログ機能をモニタリングすることで、サブシステム単位に処理結果の異常値を検出可能となるテスト環境が構築できる。

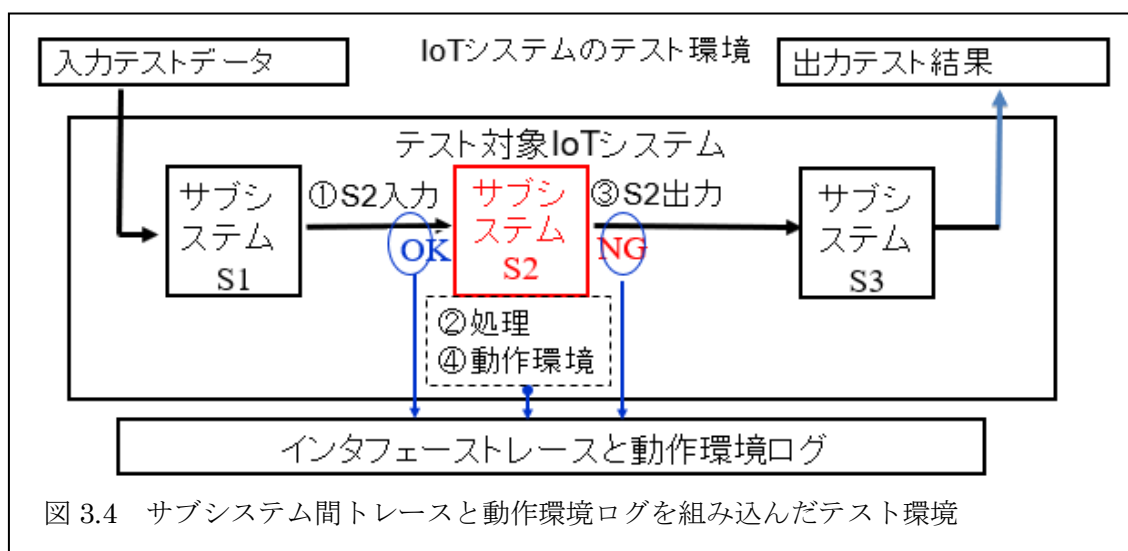


図 3.4 サブシステム間トレースと動作環境ログを組み込んだテスト環境

3.3.2.3 テストケース作成

製品の仕様範囲に着目した試験ケースを表 3.2 に示す。前述したようにシステムの動作は、サブシステムへのイベント情報の入力起点となっており、動作環境がサブシステムの

処理への影響のパラメータとなる。サブシステムの入力は製品仕様の範囲内の値と、製品仕様の範囲を超える値に分けられる。さらに製品仕様と関係なく運用操作というパラメータが考えられる。この分類をテストケースに当てはめる。**表 3.2**では、システム製品が仕様どおりに正しく動作するか、仕様を超える入力データに対しても不正な動作をしないか、仕様で想定しない環境においても期待通りに動作するかという、3つのテストケースを示している。また第三者検証部門での独立した試験の実施を考慮して、機能・性能・相互接続性・ユーザビリティ・セキュリティの5種類のテスト観点を設定している。**表 3.2**に示すテストケースを、具体的製品仕様の範囲内の値、製品仕様の範囲を超える値、製品仕様と関係ない運用操作の順に述べる。表中(1)機能試験では、操作性、弱点、異常回復試験ケース。続いて(2)性能試験では、目標値、限界値、長時間走行試験ケース。(3)相互接続性試験では、代表端末接続、多端末同時接続、実ネットワーク接続試験。(4)ユーザビリティ試験では、アンケート試験、ユーザビリティエンジニアによるヒューリスティック試験、一般ユーザのモニタによる知的ウォークスルー試験。(5)セキュリティ試験では、アタックツール試験、セキュリティ診断、ハッキング試験、をそれぞれ試験ケースとする。

表 3.2 製品の仕様範囲に着目した試験ケース

テストの観点	テストケース
	仕様範囲内
	仕様範囲を超える
	想定外の操作
(1) 機能試験	操作性
	弱点
	異常回復
(2) 性能試験	目標値
	限界値
	長時間走行
(3) 相互接続性試験	代表端末接続
	多端末同時接続
	実ネットワーク接続
(4) ユーザビリティ試験	アンケート
	ヒューリスティック
	知的ウォークスルー
(5) セキュリティ試験	アタックツール
	セキュリティ診断
	ハッキング

3.4 家庭電化製品の遠隔操作システムへの本手法の適用

3.4.1 家庭電化製品の遠隔操作システムのモデル

本第三者検証手法の対象とする家庭電化製品の遠隔操作システムの動作環境は、インターネット上のサーバおよび、サーバに接続された家庭内ゲートウェイと携帯電話、さらにゲートウェイに接続された家庭電化製品で構成される。本論文におけるテスト環境は、実機とシミュレータの両方で実現している。システムと環境の相互作用については、試験項目に合わせて実機とシミュレータの両方を使い分けて確認を行う。本第三者検証手法の対象とする家庭電化製品の遠隔操作システムを図 3.5 に示す。

家庭電化製品(以降デバイスと表現)およびスマートメータはゲートウェイコントローラに接続される。デバイスの動作状態および消費電力値はゲートウェイコントローラを経由してクラウドサーバに蓄積される。スマートホンを用いてインターネット経由でクラウドおよびゲートウェイコントローラを経由してデバイスの遠隔操作と消費電力値の表示を行うことができる。本システムを検証対象システムとして、表 3.2 に示すテストの観点のうち、代表的な機能テストと性能テストを選んで実施する。クラウドサーバに接続される家庭は 1 万世帯である。

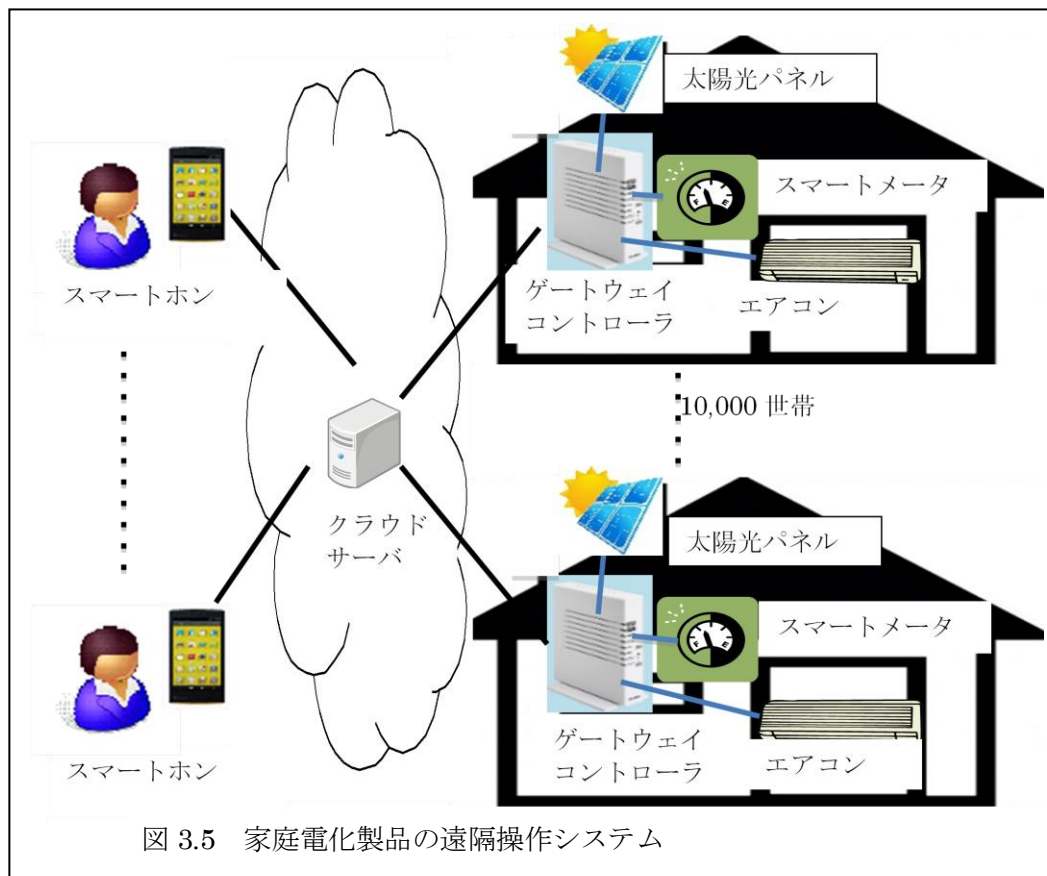


図 3.5 家庭電化製品の遠隔操作システム

3.4.2 検証対象 IoT システム

本手法の検証対象として、利用の効率性を考慮して上記デバイス遠隔操作システムと機能及びシステム構成が同等と考えられる IoT/M2M (Machine to Machine)システム⁽⁹⁾⁽¹⁰⁾を第三者検証の実証の対象とする。今回構築した家電遠隔操作テストシステム⁽¹¹⁾構成を図 3.6 に示す。

デバイスとそれを制御するゲートウェイ、制御結果データを集積・分析するクラウドサーバ上の Web アプリケーションサーバと、サーバに接続してデバイスの状態を見るスマートホンで構成され、デバイスとゲートウェイを結ぶエリアネットワークには ECHONET Lite⁽¹²⁾、ゲートウェイとサーバは TCP/IP で接続される。デバイスから送られたデータをゲートウェイ経由でクラウドサーバへ蓄積し、そのデータをスマートホンで表示する。またスマートホンからデバイスへの指示はサーバを経由してゲートウェイに接続されているデバイスへ伝える。このシステムにおける試験対象は、ゲートウェイ及びクラウドサーバ上の Web アプリケーションであり、その他のデバイスとスマートホンは独立したサブシステムと考える。検証を行って潜在欠陥を検出した場合、直接対処できる対象はこの 2 つのサ

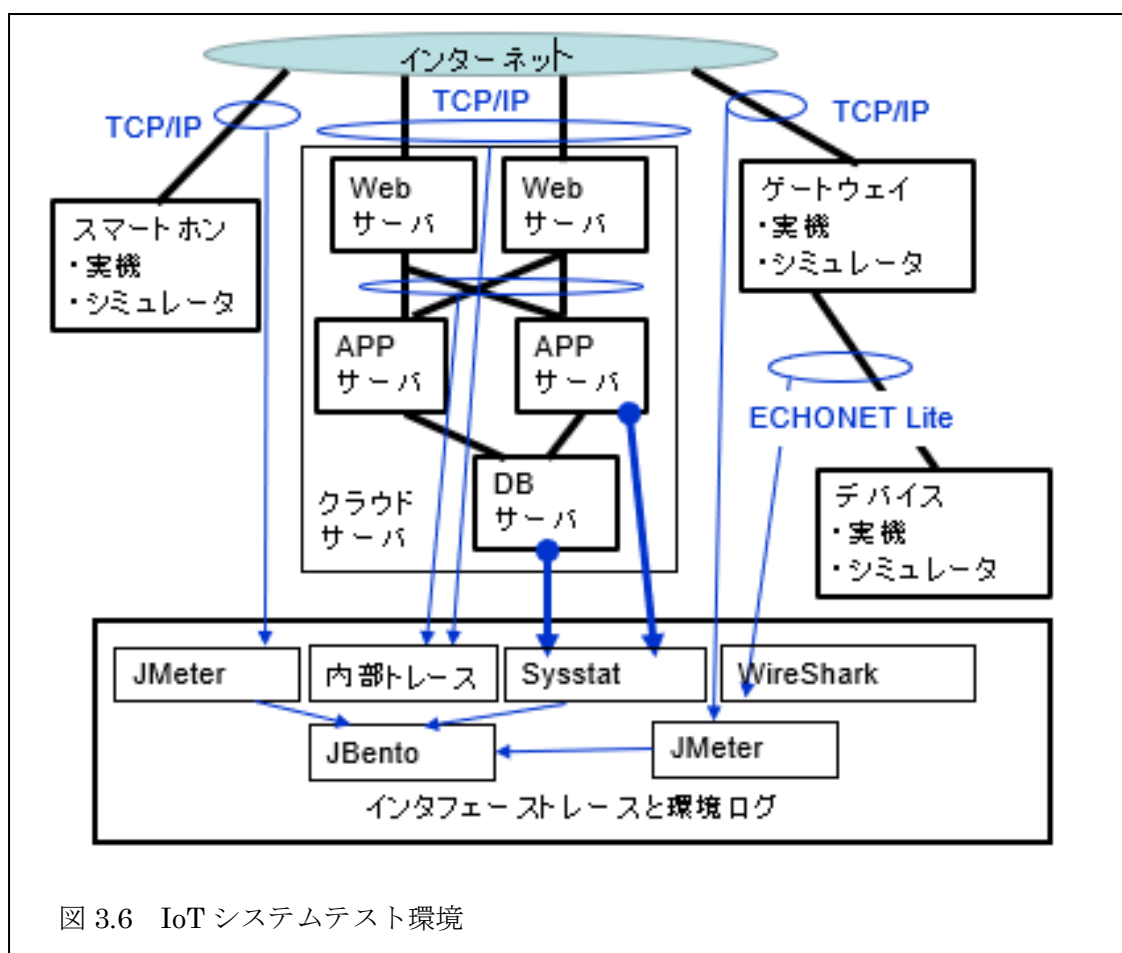


図 3.6 IoT システムテスト環境

ブシステムであり、デバイスとスマートホンは直接変更を加えることはできない独立したサブシステムである。テストシステム規模として、1万世帯の遠隔操作要求は全世帯数の1%が同時動作すると仮定して、100台のデバイスに対して同時に100台のスマートホンの接続を想定した環境を準備する。ゲートウェイおよびスマートホンはPCを用いてそれぞれ100台のシミュレートを行う。

3.4.3 検証手順

図 3.6 に示す家電遠隔操作テストシステムを対象に、表 3.2 に示すテストの観点のうち、代表的な機能テストと性能テストを選んで実施する。

3.4.3.1 サブシステムへ分解した試験環境構築

図 3.6 に示す家電遠隔操作テストシステムを構成する、デバイス、ゲートウェイ、サーバアプリケーションおよびデバイス操作を行うスマートホンの機能単位にサブシステムとして把握する。家電遠隔操作テストシステムにおいて、機能テストの ECHOENT Lite 必須コマンド確認試験ではゲートウェイとエアコンを実機同士で接続して、さらにタイムアウト発生試験はデバイス側をシミュレータで実施している。性能テストでは1万台のゲートウェイとスマートホンの接続について同時動作する台数を1%の100台と想定して、ゲートウェイとスマートホンそれぞれ100台の同時動作をシミュレータで実現し、実機とシミュレータを組み合わせて環境との相互作用について検証する。デバイス本体に加えて、デバイスのシミュレータとしてPC上で動作する「えねっとくん」⁽¹³⁾（リコーITソリューションズ製）を利用。ゲートウェイとデバイス間の入出力インタフェーストレースを採取するツールとしてオープンソースソフトウェア（以降 OSS と記載）である Wireshark⁽¹⁴⁾を利用。ゲートウェイシミュレータとしてPC上で動作する OSS の JMeter⁽¹⁵⁾を利用。サーバ動作環境について CPU・メモリ・LAN の負荷を CentOS 上の sysstat 機能を用いてログを採取し、スマートホンシミュレータとしてPC上で動作する OSS の JMeter を利用する。収集したトレースおよびログデータを解析するツールとして OSS の JBento⁽¹⁶⁾を用いる。

3.4.3.2 テストケースの設定

M2M プロトタイプテストで実施する機能試験と性能試験について、家電遠隔操作テストシステムのテストケース表 3.3 に示す。

（1）機能試験 製品仕様の範囲内の値に基づくテストケースとして、表中の① ECHOENT Lite の必須コマンドを選択する。製品仕様の範囲を超える値のテストケースとして、② ECHOENT Lite で定義されていないプロトコル上の応答タイムアウトを設定する。製品仕様と関係ない運用操作のテストケースとして、③ WiFi のリンクロス後の復旧テストを設定する。

機能試験で考慮すべき仕様は、スマートホンとゲートウェイ間の TCP/IP(HTTP/HTTPS) プロトコル、およびゲートウェイとデバイス間の ECHONET Lite プロトコルと WiFi プロトコルである。

表 3.3 テストケース

テストの観点	テストケース
	仕様範囲内
	仕様範囲を超える
	想定外の操作
(1) 機能試験 スマートホンから家庭内エアコン を操作	①ECHONET Lite必須コマンド操作
	②応答タイムアウト(規定なし)
	③WiFiリンクロス後の回復
(2) 性能試験	①50台同時動作時の応答性能
	②100台同時動作による限界性能
	③3日間の電源ON/OFF連続操作

(2) 性能試験 製品仕様の範囲内の値に基づくテストケースとして、表 3.3 の表中 ①50 台のデバイスを同時動作させ、目標応答性能を 5 秒とする。製品仕様の範囲を超える 値のテストケースとして、②100 台のデバイスまで同時動作させた場合の応答性能を測定す る。製品仕様と関係ない運用操作のテストケースとして、③3 日間連続でデバイスの電源 ON/OFF 操作を繰り返すテストを設定する。

性能試験で考慮すべき仕様は、サーバおよびゲートウェイの OS（オペレーティングシス テム）の CPU、メモリ、LAN のリソース負荷と、アプリケーションのマルチスレッド（複 数同時動作）処理である。

この試験実施のためのテストデータとして、スマートホンあるいはスマートホンシミュレ ータから、つぎの(a)から(e)の操作シー ケンスを繰り返し入力する。

- (a)デバイスの電源 ON の操作
- (b)デバイスのステータスを入手し表示
- (c)デバイスの電源 OFF の操作
- (d)デバイスのステータスを入手し表示
- (e)(a)へ戻り操作を繰り返す。

スマートホンシミュレータである JMeter の設定例として上記(c)デバイスの電源 OFF のパラメータを、JMeter 入力 データ図 3.7 に示す。HTTP プロトコル上 でパワーオフのメッセージを送り、応答 コードの正常（200）を確認している。

```
<HTTPSamplerProxy guiclass="HttpTestSampleGui"
testclass="HTTPSamplerProxy" testname="パワー
OFF" enabled="true">
    <elementProp
name="HTTPSampler.Arguments"
elementType="Arguments"
guiclass="HTTPArgumentsPanel"
testclass="Arguments" enabled="true">
        . . .
    </elementProp>
    <ResponseAssertion guiclass="AssertionGui"
testclass="ResponseAssertion" testname="アサーショ
ン/応答コード=200" enabled="true">
```

図 3.7 JMeter による入力データ

3.5 実施結果と評価

3.5.1 不具合箇所と原因の推定

機能試験と性能試験として実施した 6 種類の試験ケースとモニタリングした試験結果およびトレースとログの対応を表 3.4 テスト結果とモニタリングデータに示す。どちらの試験においても①製品仕様の範囲内の値に基づく試験ケースと②製品仕様の範囲を超える値の試験ケースで異常値(NG)を検出している。これら 4 種類の NG の試験ケースについて、図 3.3 に示した不具合推定手順と表 3.1 に示した不具合原因のタイプ分けを用いた推論方式の実施結果を、表 3.5 不具合サブシステムと不具合原因の推定のプロセスに示す。異常値(NG)が検出された表中の 4 種類の試験ケース NG(1), (2), (3), (4)それぞれについて、不具合箇所と原因の推定プロセス(P1), (P2), (P3)を以下に述べる。

表 3.4 テスト結果とモニタリングデータ

テストの観点	テストケース	テスト結果	ログとトレース
(1)機能試験 スマートホンから家庭内エアコンを操作	①ECHONET Lite必須コマンド操作	NG(2)	図3.9
	②応答タイムアウト(規定なし)	NG(1)	図3.8
	③WiFiリンクロス後の回復	OK	
(2)性能試験	①50台同時動作時の応答性能	NG(4)	図3.13 図3.14
	②100台同時動作による限界性能	NG(3)	図3.10 図3.11 図3.12
	③3日間の電源ON/OFF連続操作	OK	

3.5.1.1 NG(1)機能試験の ECHONET Lite で未定義のタイムアウト値設定時の不具合

この試験ケースでは、インタフェースの弱点試験として ECHOENT Lite で規定されていないデバイスからの応答タイムアウトの扱いに着目し、デバイスシミュレータの応答時間を 5 秒に設定した。不具合箇所の推定手順を表 3.5 の行 NG(1)に示す。

(1) (P1) インタフェーストレース上の不具合箇所検出

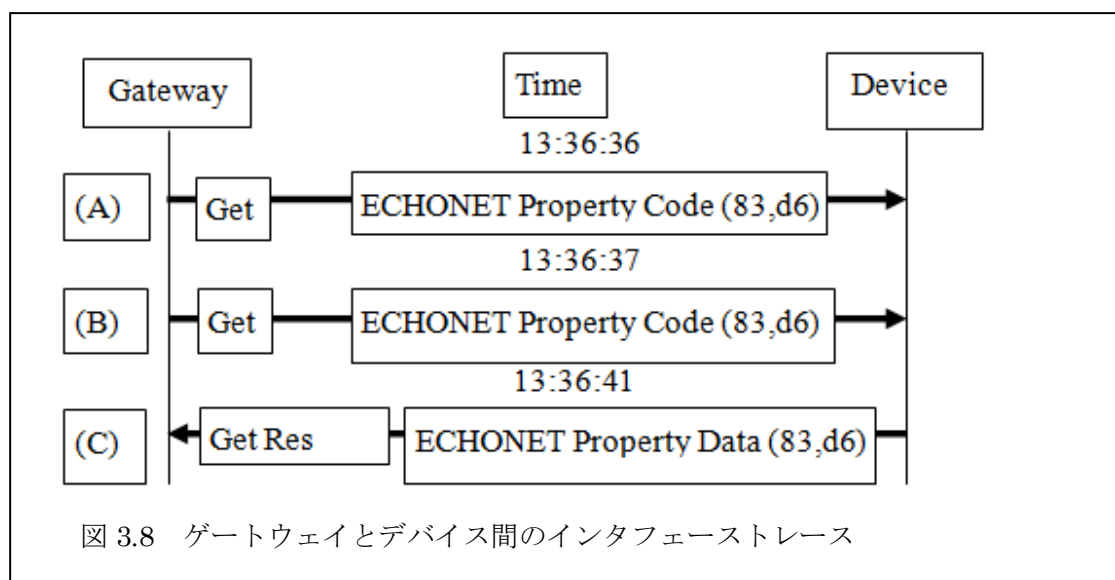
ゲートウェイでデバイスが認識できないという試験結果が検出された。ゲートウェイとデバイス間の入出力インタフェーストレースを Wireshark で採取し、ECHONET Lite プロト

表 3.5 不具合発生箇所と原因の推定プロセス

NG No.	(P1)NG結果から不具合箇所を推測		(P2)NG結果を不具合現象へ変換	(P3)障害現象から原因を推定	
	NG結果	発生箇所	不具合現象	不具合原因	不具合箇所
NG(1)	タイムアウト	ゲートウェイとデバイス間	イベント不達	(d)リトライタイミング不一致	ゲートウェイ
NG(2)	デバイス認識できない	ゲートウェイとデバイス間	処理結果不具合	(a)処理誤り or (b)インタフェース不一致	ゲートウェイ
NG(3)	処理遅延	スマホシミュレータとクラウドサーバ間	イベント処理遅れ	(g)リソース不足(CPU)	APPアプリ(ゲートウェイ)
NG(4)	処理遅延	スマホシミュレータとクラウドサーバ間	イベント処理遅れ	(h)他のサブシステム処理待ち or (j)同時並行処理の誤り	不明

コルシーケンスに表した内容を，ゲートウェイとデバイス間のインタフェーストレース図 3.8 に示す。

このインタフェーストレースから，プロトコル規約上リトライしている箇所を探し出す。図 3.8 からゲートウェイは，(A)ポーリング送信後，デバイスシミュレータの応答を待たず 1 秒後に，(B)ポーリングのリトライをおこなっていることから異常値として，タイムアウトがトレース上発生していることがわかる。今回インタフェース上の外乱は発生していないことから，異常値発生の原因の箇所はゲートウェイ装置とデバイス間のインタフェースにあると推測する。



(2) (P2)障害現象を推定

インタフェーストレースから、ゲートウェイからデバイスへ送信されているデータ内容は正しいと判断する。この結果表中のタイムアウトという異常値は、イベントが伝わらないという障害現象であるに対応付ける。

(3) (P3)不具合原因を推定

図 3.8 のインタフェーストレースから、デバイスシミュレータからの応答(C)は(A)の 5 秒後に送信されている。この現象をもとにリトライ原因を推定する。ゲートウェイのポーリングのタイムアウト値は「1 秒」であり、デバイスシミュレータからの応答に「5 秒」かかる場合は、ゲートウェイは応答を受け取らずにポーリングの再送を行うと推定できる。つまり不具合原因はゲートウェイのポーリングタイムアウト時間に依存する、表中の(d)リトライタイミングの不一致にあると推定される。

インタフェースの弱点試験では、デバイスの応答時間が 5 秒以上かかる場合、ゲートウェイからのポーリングに応答できないことがわかる。デバイスは直接変更を加えることができないサブシステムであることから、不具合箇所はゲートウェイ装置にあると推定する。

3.5.1.2 NG(2)機能試験の ECHONET Lite 必須コマンド操作時の不具合

この試験ケースでは実際のデバイスを接続して、フマートホンからゲートウェイ経由でデバイスに対して必須コマンドの操作ができることを検証した。不具合箇所の推定手順を表 3.5 の行 NG(2)に記述する。

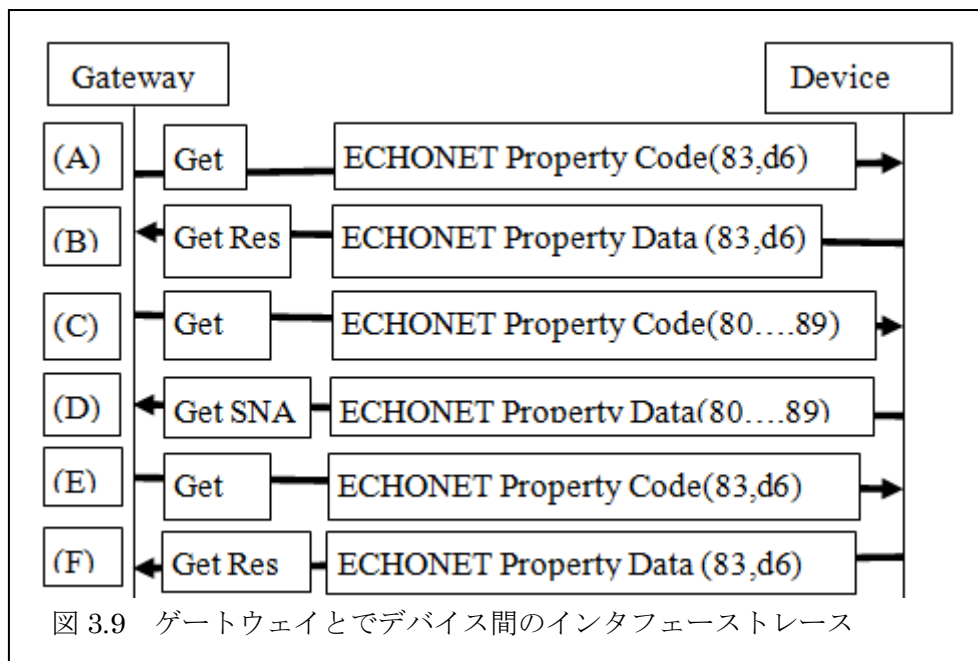
(1) (P1)インタフェーストレース上の不具合箇所検出

ゲートウェイでデバイスが認識できないという試験結果が検出され異常値として、結果誤りが対応付けられる。デバイスに一番近い、ゲートウェイとデバイス間の入出力インタフェーストレースを Wireshark で採取し ECHONET Lite プロトコルシーケンスに表した図を、ゲートウェイとデバイス間のインタフェーストレース図 3.9 に示す。

図 3.9 (A)では、ゲートウェイからデバイスへ接続開始時のデバイス情報 Get が送付されこれに対して、(B)でデバイスが応答データが返信されている。次に(C)でゲートウェイからデバイスに対してデバイス情報 Get の要求が出されているが、(D)ではデバイスから否定の応答が返信されている。このためゲートウェイは(E)でデバイスへ接続開始時のデバイス情報 Get が送付され、これに対して、(F)でデバイスが応答データが返信されている。(E)(F) は(A)(B)と同じシーケンスであり、以降このシーケンスを繰り返し先へ進まない。結果異常という異常値発生の原因の箇所はゲートウェイ装置とデバイス間のインタフェースにあると推測する。

(2) (P2)障害現象を推定

インタフェーストレースから、ゲートウェイからデバイスへ送信されているデータ内容は正しいと判断する。表中の結果誤りという異常値は、処理結果誤りという障害現象であるに対応付ける。



(3) (P3)不具合原因を推定 処理結果誤りという不具合現象に対する不具合原因候補は、表中の(a)処理誤りと(b)インタフェース不一致の 2 つが考えられるが、この時点ではどちらか特定できない。

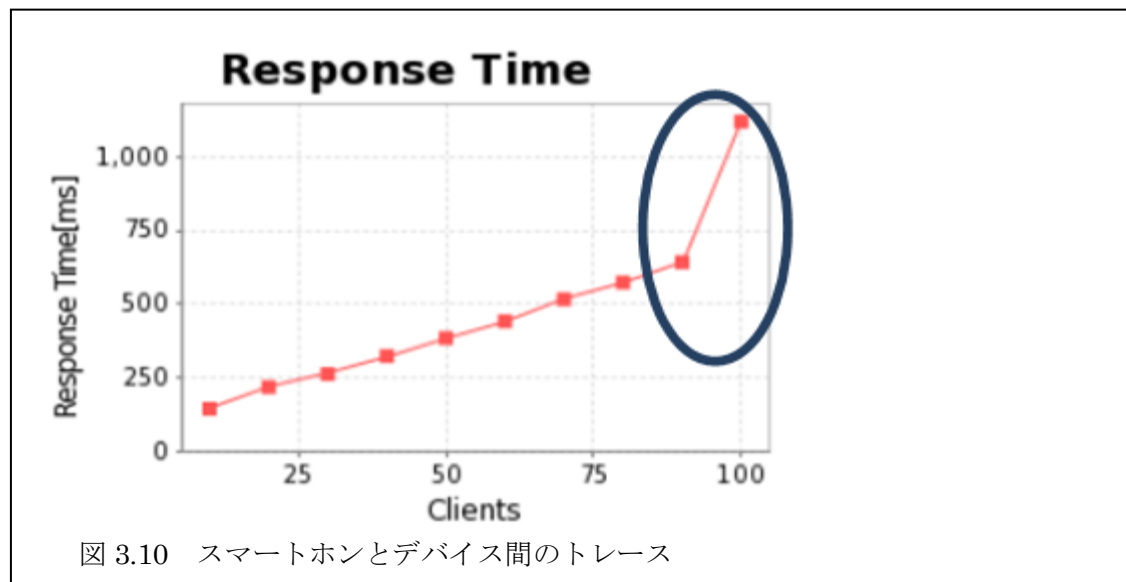
3.5.1.3 NG(3)性能試験の限界性能確認時の不具合

この試験ケースでは同時接続スマートホンの台数の限界値を検出するために、同時接続数の設計値 50 を超えて 100 台まで接続し測定する。不具合箇所の推定手順を表 3.5 の行 NG(3)に記述する。

(1) (P1)インタフェーストレース上の不具合箇所検出

スマートホンシミュレータ(JMeter)とインターネット間の入出力インタフェーストレースをもとに、オープンソースプログラムの解析ツール(JBento)を用いて応答性能をグラフ化し、スマートホンとデバイス間のインタフェーストレース図 3.10 に示す。

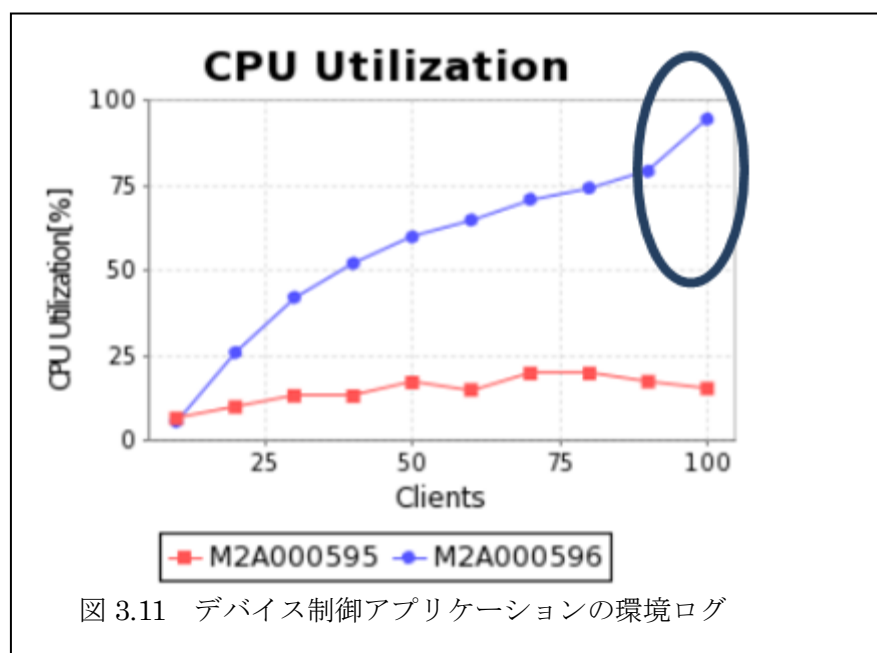
グラフの横軸に同時接続デバイス数を、縦軸にデバイスからスマートホンへの応答時間を示している。このグラフからレスポンスタイムについて、スマートホンによるデバイスの同時制御で 50 台を超えて 100 台までの同時動作時の応答性能を把握する。その結果としてすべて目標値「5 秒」をクリアしているが、デバイス数が 90 を超えるとレスポンス時間が急に 1 秒をオーバーして応答が遅くなるという異常値として、処理遅れが検出されることがわかる。異常値発生の原因の箇所はスマートホンとクラウド上のサーバ間にあると推測する。



(2) (P2)障害現象を推定

応答性能が目標値を超えるという表中の処理遅れという異常値は，イベント処理の遅延という障害現象であるに対応付ける。

(3) (P3)不具合原因を推定 スマートホン制御アプリケーションとデバイス制御アプリケーションの動作環境ログ(sysstat)をもとに，性能に関するリソースとしてCPU，Javaメモリ管理，LANの各負荷状況について解析ツール(JBento)を用いてグラフ化し，図 3.11 アプリケーション制御環境ログ情報に示す。図 3.11 は，横軸は同時接続デバイス数を，縦



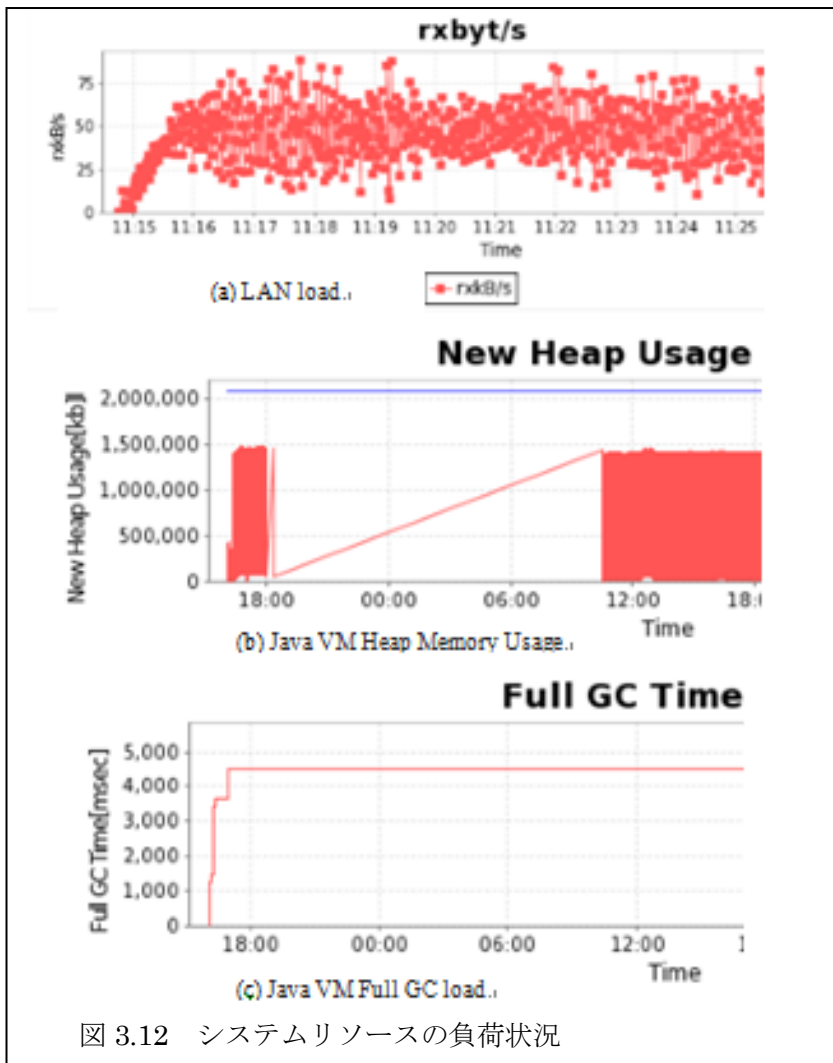


図 3.12 システムリソースの負荷状況

軸はサーバ CPU 負荷率を示している。このグラフからデバイス数 90 台を超えた時点で M2A000596(デバイス制御アプリケーションサーバ)の CPU 負荷が 75%を超えてアプリケーションの動作が遅くなることが推定できる。

サーバの性能に関係する値として CPU 負荷の他に LAN 負荷と Java メモリ使用量があり、これらを、図 3.12 システムリソース負荷情報に示す。

図 3.12 (a)は横軸に時刻縦軸に単位秒当りの受信データのバイト数を示している。サーバの LAN の負荷は最大で 80KByte/sec であり、LAN の最大通信速度性能は 100Mbps であることから LAN の負荷は 0.8%であり問題ないことがわかる。図 3.12 (b)は横軸に時刻を、縦軸に JavaVM のヒープメモリの使用量を示している。グラフからヒープメモリは取得と開放を繰り返しており問題ないと判断できる。図 3.12 (c)は横軸に時刻を縦軸に JavaVM のフル GC の発生によるフル GC に要した時間を示している。グラフから JavaVM の FullGC の発生は、

システムが安定稼働した計測時間範囲の 24 時間に 1 回 1 秒発生している程度であることから、システム性能への影響は問題ないと言える。以上の結果から限界値性能となるボトルネックの原因はデバイス制御アプリケーションサーバ(M2A000596)の表中の(g)CPU リソース不足にあると判断できる。

3.5.1.4 NG(4)性能試験の目標性能確認時の不具合

この試験ケースでは同時接続スマートホン 50 台の応答目標性能 5 秒の確認を行う。不具合箇所の推定手順を表 3.15 の行 NG(4)に記述する。

(1) (P1)インタフェーストレース上の不具合箇所検出

スマートホンシミュレータ(JMeter)とインターネット間の入出力インタフェーストレースをもとに、オープンソースプログラムの解析ツール(JBento)を用いてに応答性能をグラフ化し、図 3.13 スマートホンとデバイス間のインタフェーストレースに示す。

グラフの横軸に同時接続デバイス数を、縦軸にデバイスからスマートホンへの応答時間を示している。このグラフからデバイス接続数 50 の応答性能は目標値 5 秒を超える異常値として、処理遅れが検出されていることがわかる。異常値発生の原因の箇所はスマートホンとクラウド上のサーバ間にあると推測する。

(2) (P2)障害現象を推定

応答性能が目標値を超えるという表中の処理遅れという異常値は、イベント通知の遅延という障害現象であるとは対応付ける。

(3) (P3)不具合原因を推定

スマートホン制御アプリケーションとデバイス制御アプリケーションの動作環境ログ(sysstat)をもとに、性能に関係するリソースとして CPU、Java メモリ管理、LAN の各負荷状況について解析ツール(JBento)を用いてグラフ化し、デバイスコントロールサーバ環境ログ図 3.14 に示す。



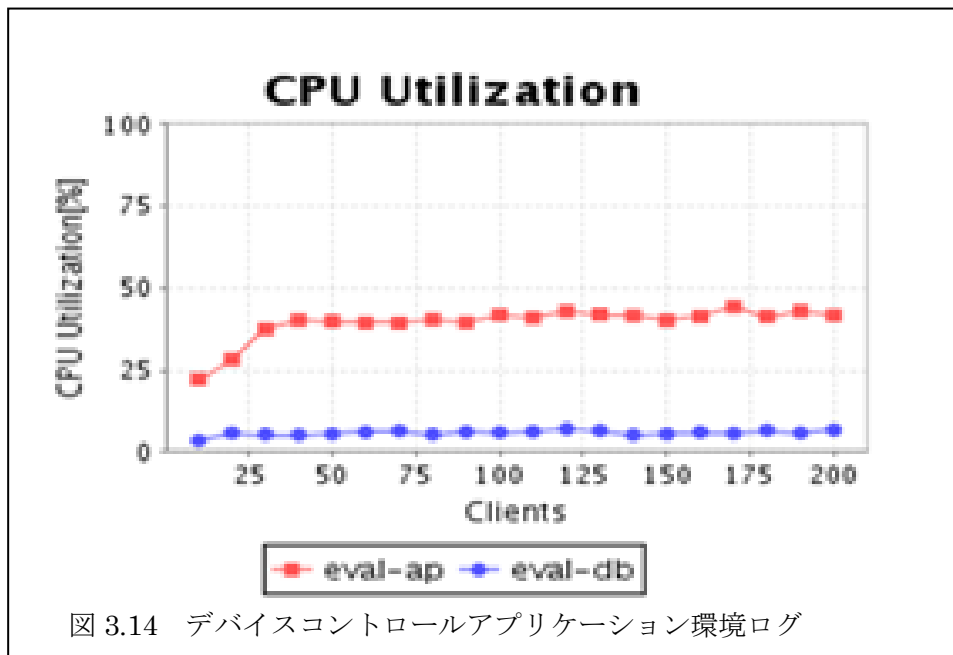


図 3.14 は、横軸は同時接続デバイス数を、縦軸はサーバ CPU 負荷率を示している。グラフからデバイスが 30 台を超えた時点から、デバイスの増加によらず CPU の負荷は一定であることがわかる。LAN および Java のヒープメモリも問題箇所は検出できていない。イベント通知の遅延という不具合現象に対応する原因候補は、表中の(g)CPU・メモリなどの計算機リソース不足、(h)他のサブシステム処理結果待ち、(i)非同期イベントの発生、(j)同時平行処理の誤りが対応する。今回は(g)計算機リソース不足は発生していない。また(i)非同期イベントも発生していない。従ってイベント通知の遅延という不具合現象に対応する原因候補は、(h)他のサブシステム(DB)へのアクセス応答待ち、あるいは(j)同時並行処理の誤りの 2 つが考えられるが、この時点ではどちらか特定できない。

3.5.2 不具合推定手法の評価

不具合推定手法の評価を、表 3.6 評価結果概要に示す。

表中の(1)②ECHONET Lite タイムアウト値設定(5 秒)テストと、(2)②限界性能確認テストにおいては、不具合箇所と原因の推定ができた結果となっている。一方、(1)①ECHOENT Lite 必須コマンド操作テストにおいては、不具合原因は 2 ケース挙げられるが特定できていない。さらに表中の(2)①目標性能確認テストにおいては、不具合箇所は予想できておらず、かつ不具原因が 2 ケース挙げられるが特定できていないという結果となっており、不具合箇所と原因の推定は不十分である。

表 3.6 評価結果概要

実施したテスト内容		不具合の発生箇所と原因の推定結果	
テストの観点	テストケース	不具合の推定箇所	推定原因
(1)機能試験 スマートフォンから 家庭内エアコン を操作	①ECHONET Lite必須 コマンド操作	○:ゲートウェイ	△:(a)処理誤り or (b)インタフェース不一致
	②応答タイムアウト (規定なし)	○:ゲートウェイ	○:(d)リトライタイミング不 一致
(2)性能試験	①50台同時動作時の 応答性能	×:推定不可	△:(h)他のサブシステム 処理待ち or (j)同時並行処理の誤り
	②100台同時動作によ る限界性能	○:ゲートウェイ制御アプリ ケーション	○:(g)リソース不足(CPU)

3.6 考察

3.6.1 不具合推定が不十分になる理由

不具合推定結果が不十分である結果となった、表 3.6 中の(1)①ECHOENT Lite 必須コマンド操作テストおよび(2)①目標性能確認テストについて、その原因を考察する。

(1) ECHOENT Lite 必須コマンド操作テストの不具合原因推定

図 3.9 のトレースデータの内容を解析すると、デバイスはゲートウェイが要求する特定のプロパティに対して応答していないことがわかる。このプロパティはオプションでありデバイスではサポートしていないことが初期処理の応答データからわかる。ゲートウェイにてデバイスがサポートしていないプロパティに対して Get 要求を行い、これに対してデバイスが否定応答を行うことはプロトコル上正しいシーケンスである。不具合箇所はゲートウェイにあると推定できるが、ゲートウェイが図 3.9 の図中(A)のシーケンスを繰り返すという不具合原因は、表 3.6 の表中(a)処理誤りと(b)インタフェース不一致のどちらか、この時点では特定できない。このため試験結果をもとに開発部門と不具合箇所と原因の解析を共同で行った結果、ゲートウェイが Get 要求している ECHONET Lite のプロパティ定義の一つであるデバイス識別子に対して、デバイスが否定応答していることが判明した。デバイス識別子は必須プロパティでないため、不具合原因は表中(b)インタフェース不一致と確定し、ゲートウェイから Get 要求を発行しないよう修正後テストを継続した。

(2) 目標性能確認テストにおける、不具合箇所と原因の推定

イベント通知の遅延不具合現象に対応する原因を、表 3.6 の表中(h)他のサブシステムへのアクセス応答待ちあるいは(j)同時並行処理の誤りのどちらか特定するため、開発部門へ解析を依頼した。その結果、デバイス制御アプリケーションにおいて同時平行（マルチスレッド）処理に関する実装上の誤りがあることがわかり、表中 (j)同時平行処理を行っていない誤りであることが確定し、修正後確認テストを実施した。

今回の検証手法の適用結果から、不具合箇所と原因がサブモジュール内部のロジックに深く関わっている場合は不具合推定が不十分となるが、第三者検証で行う推定の限界と言える。

3.6.2 システム運用へのフィードバック

今回の〈3.5.1.3〉限界値試験の観点では、スマートホン 100 台の同時接続環境において、90 台の同時接続が限界であるという結果を得た。同時接続台数は全体の 1%と想定するとこの結果から、クラウドサーバに接続されているゲートウェイコントローラの接続台数が 9,000 台を超えると、デバイス制御サーバの CPU 負荷が 75%を超えて不安定になる可能性があると予想される。運用にあたって同時接続台数を監視して、50 台を超えた時点でサーバ性能の増強を行うなどの対応を取るなど運用面へのフィードバックが可能である。

3.6.3 第三者検証を適用する際の工夫点および限界点

本方式の適用に際しては、開発部門である自部門への説得以外に、他の組織である第三者検証側へ次のような介入や協力伺いが生じる。

- ・ 検証方法への要請、技術指導、検証項目の優先付け、検証結果の報告方法

これらは開発部門が第三者検証を実施する際の障壁の一つになり得る。このため、検証部門が要件定義書に基づいて作成する検証計画書を、開発部門で承認するという手続きが重要になる。ここでは、試験環境の構築や試験結果データの採取方法、試験進捗と不具合の報告タイミングなどについての記述が必要と考えている。本論文の実施例である家庭電化製品の遠隔操作システムの検証において、筆者らは検証環境に組み入れるトレースやシミュレータのツール選択、および検証期間中のデイリーレポートの報告様式や障害発生時に採取するデータ形式について検証部門が検証計画書に記載し、開発部門が承認するという手続を行った。これにより開発部門が第三者検証を適用する際に感じる障壁の軽減を図った。

本提案の手法は、試験結果の異常値から、要件定義書で提示されるサブシステム単位に不具合箇所と不具合原因の推定を行うものである。しかしながら、本手法の限界点として次の事項が考えられる。

- ・ 不具合箇所の推定は、要件定義書に記載されたサブシステムの単位に留まる。サブシステム内に存在する不具合の箇所はわからない。
- ・ 仕様の実装方法に依存する不具合の場合、発生箇所と原因の推定ができない。
- ・ 検証実施の力を不具合推定の活動にある程度割かれるため第三者検証部門に期待される、気づきといわれるテスト項目以外の不具合指摘の件数が低下するという影響はある程度避けられない。

3.6.4 ミッションクリティカルな分野に適用される検証の手法との比較

(1) STAMP, STPA STAMP は、〈3・2・3〉で述べたようにシステムの制御コンポーネント間の連携動作に障害原因を見つける事故原因の解析手法であり、STPA は STAMP の手法に基づいて、発生が予想される事故要因を解析する設計手法と言える。一方、本方式は第三者によるシステム試験において、開発部門から提示される要件定義書に基づいてシステム製品を対象にテストを実施して、不具合発生箇所と原因を検証部門がサブシステム単位に推定するテスト方式である。本方式においては、〈3・3・2・3〉に記述した製品の仕様範囲に着目したテストケースを入力として、システムテスト環境から出力されたテスト結果とトレース、およびログから、不具合発生箇所と原因を推定する手法に特徴がある。

(2) NASA や JAXA で行われている包括的な IV&V 〈3・2・3〉で述べた包括的な IV&V と比較して、本方式は第三者によるシステム試験の段階で、開発部門から提示される要件定義書に基づいてシステム製品を対象にテストを行い、不具合発生箇所と原因を検証部門がサブシステム単位に推定するテスト方式である。なお、システム試験における第三者検証方式としては、製品仕様範囲に着目したテストケースを入力として、システムテスト環境から出力されたテスト結果とトレース、およびログから、サブシステム単位に不具合発生箇所と原因を推定する手順自体に特徴があると考えている。

(3) 設計 IV&V IV&V に基づく設計レビューは、〈3・2・3〉で述べたように開発が作成する設計書を対象に、上位設計書に基づいて、妥当性と有効性の評価を行うものと言える。

一方、本方式は、第三者によるシステム試験において、開発部門から提示される要件定義書に基づいてシステム製品を対象にテストを実施して、不具合発生箇所と原因を検証部門がサブシステム単位に推定するテスト方式である。〈3・3・2・3〉に記述した製品の仕様範囲に着目したテストケースを入力として、システムテスト環境から出力されたテスト結果とトレースおよびログから、不具合発生箇所と原因を推定する手法に特徴がある。この要件定義書には、システム製品が実現する機能とサブシステム構成は記述されるが、サブシステムの内部の仕様は記述されていない。本方式では、この状況において、テスト結果から不具合の指摘を可能とすることを狙いとしている。

本方式を適用した例において、ゲートウェイの接続相手であるエアコンは、ECHONET Lite 規格に適合しているが、サポートしているプロパティおよびタイムアウト値については、実装に依存する仕様として公開されていない。このため実際に接続テストを行うことで、ゲートウェイの不具合を検出している。

(4) AADL 〈3・2・3〉で述べた AADL のエラーモデルと比較して、本方式は、第三者によるシステム試験において、開発部門から提示される要件定義書に基づいてシステム製品を対象にテストを実施して、不具合発生箇所と原因を検証部門がサブシステム単位に推定するテスト方式である。〈3・3・2・3〉に述べた製品の仕様範囲に着目したテストケースを入力として、テストを実施する点に特徴がある。

3.6.5 テストの観点についての考察

本手法を適用するテストでは、開発部門が提示する要件定義書に記載された機能・サブシステム構成・ユーザインタフェース・規格・動作環境に対応したテスト項目を、機能・性能・相互接続性・セキュリティの観点と、〈3・3・2・3〉に記述した製品の仕様範囲に着目したテストケースを用いて作成している。実際のシステム製品を対象としたシステム試験においては、検証の目的は果たしていると考えている。本手法を適用して実施した、4章の家庭電化製品の遠隔操作システムのテストにおいて、不具合を検出している。なお、テスト観点は、ISO9126⁽¹⁷⁾の品質特性の中から機能・性能・相互接続性・ユーザビリティ・セキュリティを選択している。しかしながら、ISO9126の品質特性では、さらに保守性・可搬性について定義していることから、保守や再利用の視点での試験については不足しており、今後の課題であると考ええる。

3.6.6 本手法の価値と限界

高度な品質と安全性が求められるミッションクリティカルな宇宙・航空分野のようなシステム製品に対しては、要求設定からシステム試験までの全ての段階において、包括的なIV&Vプロセスを用いた、第三者によるレビューあるいはテストが実施されている。本方式は、第三者によるシステム検証の段階で、システムの機能・準拠する規格・ユーザインタフェース・動作環境が記載された要件定義書に基づき、製品仕様範囲に着目したテストケースを用いて、テスト環境とテストデータを作成し、テストターゲットであるシステム製品そのものを対象にテストを実施する。収集したテスト結果は、要件定義書に照らして妥当性と有効性を評価し、不具合と判定した事象は、サブシステムの連携を遡る方式で不具合発生箇所と原因を検証部門がサブシステム単位に推定する、という点に価値があると考ええる。この特徴は民生品の製造サービス分野等への適用に効果があると考ええる。本手法は、システムの実装仕様を参照せずに、テストターゲットであるシステム製品そのものを対象にテストを実施することから、詳細仕様書および実装仕様書に基づくテストは範囲外であり、それらに依存する不具合の推定はできない。このように宇宙・航空分野などのクリティカルな製品分野で求められる品質レベルの試験に適用するには限界点がある。

3.6.7 今後の課題

今回は機能試験および性能試験に対して提案手法を実装し確認したが、さらに提案手法を充実させるための課題を以下に述べる。

(1) **テストケースの実績の拡大** 今回製品仕様の限度に着目した3つのテストケースを、表3.2に示すテストの観点に対して適用し、機能試験と性能試験について実施したが、未実施のテスト観点についてもシステム製品として必要な試験項目であり実証が必要である。これらテスト観点についても今回提案するテストケースを適用し、テストケースの実績の拡大を行う所存である。

(2) システム障害事例に基づく本手法の有効性の評価

情報処理推進機構ではシステム障害の事例に基づく教訓集⁽¹⁸⁾を発行し、障害事例の共有により同一障害の発生を予防する活動を行っている。この障害事例を第三者検証で実施するテスト項目に適用すると、既存のシステム障害を検証の段階で検出することができると思われ。障害事例を検出できるテストケースを作成し本手法の評価を行う所存である。

3.7 まとめ

本研究では、開発者が IoT システムを構成するサブシステムを明らかにし、予めインタフェーストレースとログを採取し障害解析を可能とするテスト環境を規定する検証要求仕様書を作成することで、検証部門を必要時に設置して第三者検証の実施を可能とし、検証部門においては検出した障害の発生箇所と原因の推定を可能とすることを目標とした。その結果、図 3.15 に示すように、開発部門と検証部門を連携させる「検証要求仕様書」および検証部門での「検出障害の発生箇所と原因の推定方式」を提案し、家庭電化製品の遠隔操作システムへ適用し、本手法が有効であるという成果を得た。

本章では、民生品IoTシステム製品に対する検出障害の推定機能をもつ第三者検証方式を提案し、実装評価を行った。

成果 : ・必要な時期に第三者検証部門を編成可能とした。
・検証部門において検出した障害の、発生箇所と原因を推定可能とし、開発部門の障害解析負荷を軽減した。

成果物: ・開発部門と検証部門を連携させる「**検証要求仕様書**」
・検証部門での「**検出障害の発生箇所と原因の推定方式**」

図 3.15 研究の成果

4. IoT プロトタイプの開発者による第三者的検証方式

本章では、民生品 IoT システム開発において作成するプロトタイプに対する、開発者による第三者的検証方式を提案する。プロトタイプを客観的に評価することで、製品完成までに行うべきことを明らかにして指摘することができれば、開発期間の短縮を図ることができる。このためには、プロトタイプで実現している機能・性能の評価だけでなく、プロトタイプをそのまま機能・性能を拡張開発して製品化するという観点での評価が必要であるが、これには製品とプロトタイプの両方の仕様を理解することが必要であることから、開発者以外の第三者部門で実施することは困難である。

本研究では、開発部門でプロトタイプについて製品と同程度のプロトタイプ機能仕様書を作成することで製品との比較を可能とし、さらに検証環境および評価項目と判定基準を明らかにした検証計画書を作成することで、開発者による第三者的検証を可能とする検証方式の提案を目的とする。具体的には、「開発者によるプロトタイプの第三者的検証方式」および「開発不足項目の指摘機能をもつプロトタイプ第三者検証方式」を提案する。

本論文では開発者が第三者検証を模して、プロトタイプに対して客観的に実施する検証を「第三者的検証」と表す。

4.1 はじめに

IoT(Internet of Things)システムでは、データ発生現場に置かれたセンサーやアクチュエータとクラウドシステムがネットワークで接続され、データの伝達、分析、結果のフィードバック動作などが、人手を介さずに行われる。この機能を用いて製造業、サービス業、農業など適用範囲を広げて新しい市場が開拓されると期待されている。IoT システムは構成要素が多様であり、またセンサーやアクチュエータという物理的な(Physical)ものとクラウド上の仮想的な(Cyber)もので構成されるという特質がある。このように技術および適用の範囲が広いと、製品化には通常長い期間が必要になる。しかしながら、プロトタイプを作成して早い時点で主要仕様の達成度を検証すれば、製品化までに開発すべき項目が明確になり、開発期間の短縮を図ることが可能になる。

製品に対する客観的評価の手法として、第三者のテスト専門部門がテストを行い、特定の基準に基づいて結果判定を行う第三者検証がある。単体製品や組込みシステム製品に対して実施⁽¹⁾されており、ミッションクリティカルなシステム製品に対しては、IV&V プロセス⁽²⁾により要求設定・設計・製造からシステム試験までの全ての段階でレビューあるいはテストを実施し、開発成果物を妥当性と有効性の観点で解析評価する第三者検証手法がとられている。民生品のシステムに対する第三者検証として、いままでに筆者らは「不具合の推定を含んだシステム製品における第三者検証」⁽³⁾において、IoT 機能を持った製品の要件定義書に基づくシステム製品に対する第三者検証を提案した。

プロトタイプにおいて、追加開発を行うことにより製品化できることの判断には客観的評価が求められることから、IoT プロトタイプに対する第三者的検証方式が必要とされたと考える。

プロトタイプには、つぎの2パターンがあると考えられるが、本研究では、継続開発を行い製品化を目指すプロトタイプを対象とする。

- ・基本機能の確認を目的とし、確認後は破棄する。
- ・システム全体の機能確認を目的とし、確認後も継続開発を行い製品化を目指す。

本論文では、IoT プロトタイプシステムの第三者的検証方式を提案する。本方式の骨子はつぎのとおりである。

(1) 仕様達成度： プロトタイプにおいても製品と同じくプロトタイプ機能仕様書を作成し、これに基づいて機能・性能・セキュリティについて仕様達成度を検証する。

(2) ハザード対策： 想定されるハザードへの対策が取られているか否かを検証する。

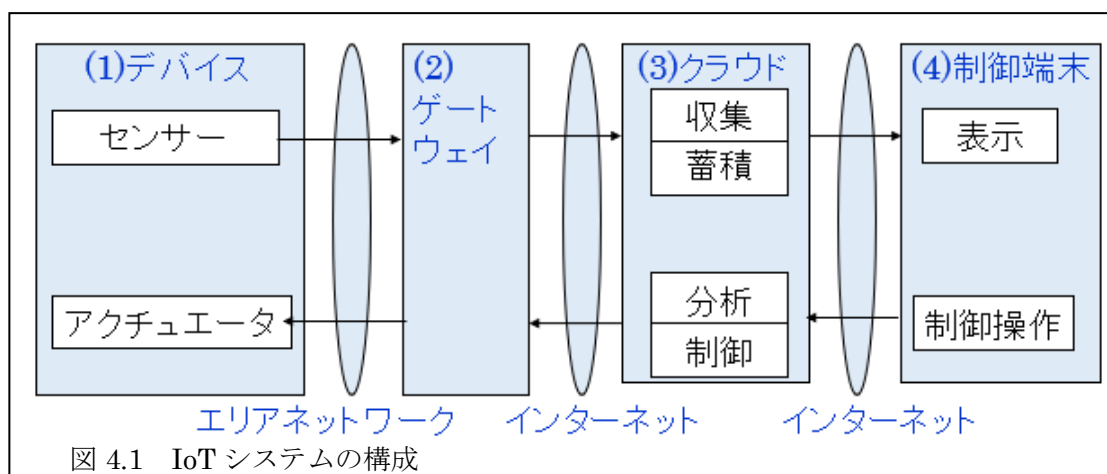
(3) 拡張容易性： プロトタイプから製品版への追加開発における、拡張の容易性を評価する。

本方式を、水耕栽培と害鳥駆除 UAV(無人飛行機：ドローン)の2種類のプロトタイプに適用し、第三者的検証を実施、製品化までに行うべき開発項目の明確化を行い、プロトタイプ検証の有効性を確認した。

4.2 IoT プロトタイプの第三者的検証の意義

4.2.1 IoT システムの構成

IoT システムは、機器や人間などフィールドからのフィジカルなセンサーデータを、エリアネットワーク、ゲートウェイ、アクセスネットワークを介してクラウドに送信する。クラウド上ではサイバーな情報システムによって、集めたデータの可視化、分析などの処理を行い、その結果をアクチュエーターにフィードバックするという動作を、人手を介さずに行う技術が集約されたものである。IoT システムの一般的な構成を図 4.1 に示す。図に示すように IoT システムは、(1)デバイス、(2)ゲートウェイ、(3)クラウドあるいはサーバ上のアプリケーション、(4)制御端末という、複数のサブシステムをエリアネットワークおよびインターネット上のオープンなインターフェースを用いて組み合わせて構築される⁽⁴⁾⁽⁵⁾。



4.2.2 プロトタイプの第三者的検証

1 章で述べたように IoT システムは早期の市場投入を行うために、プロトタイプシステムに対して機能・性能の追加開発を行い、早期に市場へリリースしてユーザの評価を受けることが期待される。プロトタイプを客観的に評価することで、製品完成までに行うべきことを明らかにして指摘することができれば、開発期間の短縮を図ることができる。

製品の第三者検証においては、開発組織に対して技術的独立性、管理的独立性、および経済的独立性をもった組織によって行われる検証および認証と定義されている⁽⁶⁾。プロトタイプでは、通常このような検証および認証機関は存在しないため、つぎの 2 とおりの検証のパターンを前提とする。まず開発者が検証についても実施し、この際に第三者の視点を用いるパターン。つぎは第三者の視点をもつ開発者以外のメンバが、開発者の協力を得て検証を実施するパターンがある。

4.2.3 ハザード検証の必要性

IoT システムはフィールドのフィジカルな世界とクラウド上の情報システムによるデータの可視化、分析などのサイバーシステムとの組み合わせになる。そのため、単体での品質検証だけでは予想できない、個々のサブシステムのインタフェースに起因するハザードが発生する可能性がある。

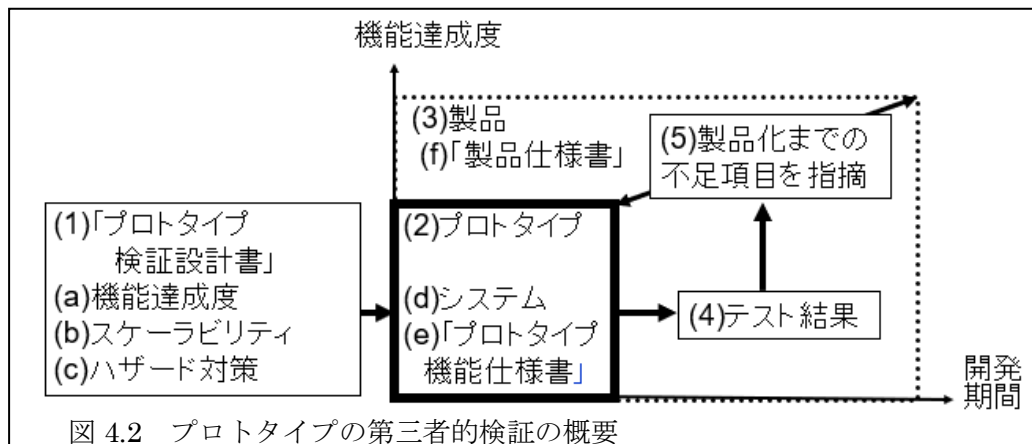
例えば、センサーからのアラーム情報が一斉に通信され、負荷が特定の箇所に集中して通信トラフィックの輻輳が発生することによりデータ送信のリトライが発生する状況では、Web ミドルウェア、データベース、ブラウザのタイマ値がそれぞれ異なるため、通信のリトライ中にタイムアウトと判断され、結果として通信エラーとなるケースが考えられる。またシステムの緊急操作として、操作員の動作が加えられるケースでは、緊急通知に操作員が気付かず緊急処置操作をしないケースや、通知を誤って解釈して操作するケースが考えられる。これらは結果としてハザードの原因となる可能性があるが、IoT の特質から、ハザード対策について製品ができてから手当すると変更箇所が多岐にわたる可能性があり、手戻りが大きすぎる。このためプロトタイプの段階から、システム上考えられる状況の悪化やシステムの変化などによりハザードが発生する条件を抽出して試験ケースを作り、ハザード対策の問題点を検出することが有効である。

4.3 IoT プロトタイプの第三者的検証方式の提案

4.3.1 プロトタイプ検証方式

4.3.1.1 検証方式の概要

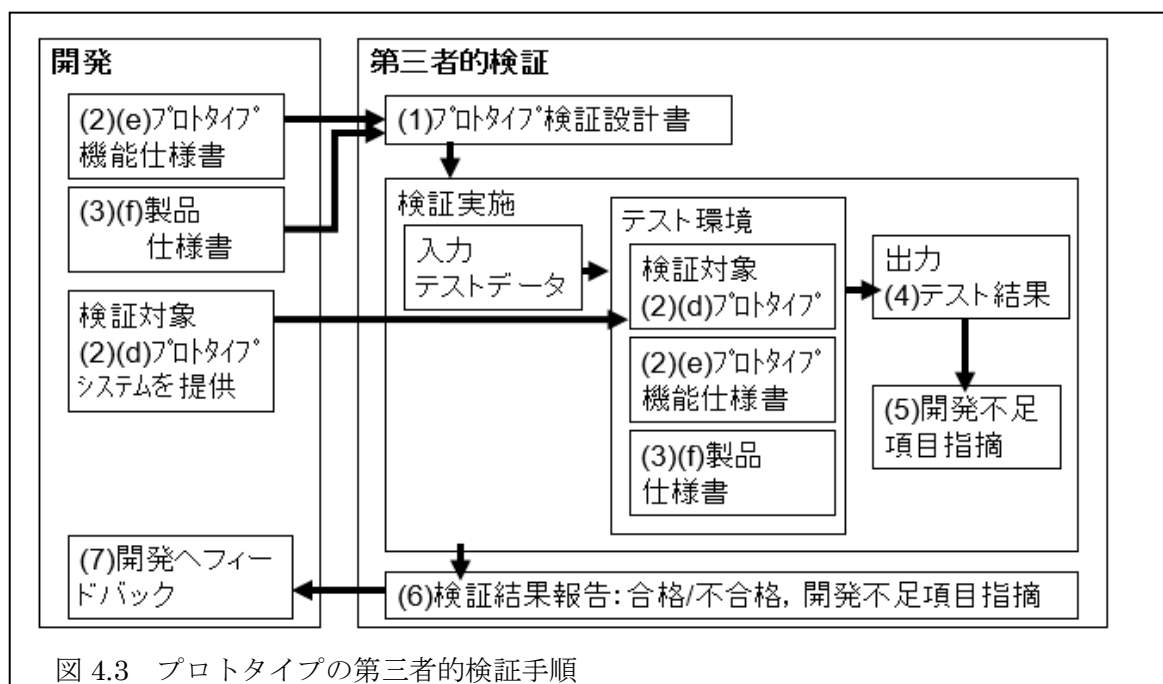
IoT プロトタイプに対して第三者的検証を行い、製品完成までに行うべき開発項目を検出する検証方式の概要を図 4.2 に示す。図中の縦軸は機能達成度、横軸は開発プロセスの進捗を表している。図中(1)はプロトタイプに対するテスト項目と判定基準を記載した「プ



「プロトタイプ検証設計書」であり、(a)仕様達成度、(b)ハザード対策、(c)拡張容易性の評価項目で構成される。この検証設計書に基づいて(2)プロトタイプの(d)システム本体と(e)プロトタイプの機能を記載した「プロトタイプ機能仕様書」というドキュメントを対象としてテストを実施する。(c)拡張容易性については、(3)製品の仕様要件を記載した(f)「製品仕様書」との比較を行う。この結果(4)テスト結果が得られる。この(4)テスト結果をもとに、(2)プロトタイプから製品化までの(5)不足項目の指摘結果を得る。

4.3.1.2 検証方式の実施手順

IoT プロトタイプに対して第三者的検証を行い、製品完成までに行うべき開発項目を検出



するプロトタイプの第三者的検証方式の流れを図 4.3 に示す。(番号は図 4.2 に対応。)

開発部門は、試験対象となるプロトタイプが実現する機能とユーザインタフェースおよび準拠する規格や動作条件に加えて、システムの構成をサブシステム単位に機能とインタフェースを記載、さらにインタフェーストレースとログを採取できるテスト環境を記述した(2)(e)プロトタイプ機能仕様書、および製品の機能仕様を記述した(3)(f)製品仕様書、さらに検証対象となる(2)(d)プロトタイプシステムを第三者検証部門へ提供する。第三者検証部門では、機能・ハザード対策・拡張性を試験の観点とした 検証項目と合否判定基準を記載した(1)プロトタイプ検証設計書を作成する。次に、テスト入力データを作成し、(2)(d)検証対象プロトタイプをサブシステムに分けてインタフェーストレースとログをモニタリングする機能を備えたテスト環境へ入力して、機能・ハザード対策の検証を行う。製品拡張性の検証は、(2)(e)プロトタイプ機能仕様書と(3)(f)製品仕様書のドキュメント上で比較を行う。続いて、出力された(4)テスト結果をもとに、(5)製品完成までに行う開発の不足項目の検出を行う。検証部門では(6)検証結果と検出した製品完成までに行う開発の不足項目を合わせて開発部門へ報告を行い、開発部門では(7)検証結果の開発へのフィードバックを行う。

4.3.1.3 検証方式の環境

プロトタイプ検証では、テストデータを入力として、IoT プロトタイプで処理を行った出力結果を期待値と比較してテストの合否判定を行う。IoT の特性として通信障害やクラウド上のアプリケーションの相互作用などがテスト結果に影響することが想定されるため、テスト実行時に想定した試験条件であることを確かめる必要がある。このために筆者らが行った「不具合の推定を含んだシステム製品における第三者検証」⁽³⁾方式のテスト環境をプロトタイプにおいても適用する。IoT プロトタイプのテスト環境を図 4.4 に示す。

テストの流れは、(1)テストデータを入力として、(2)プロトタイプシステムのテスト環境

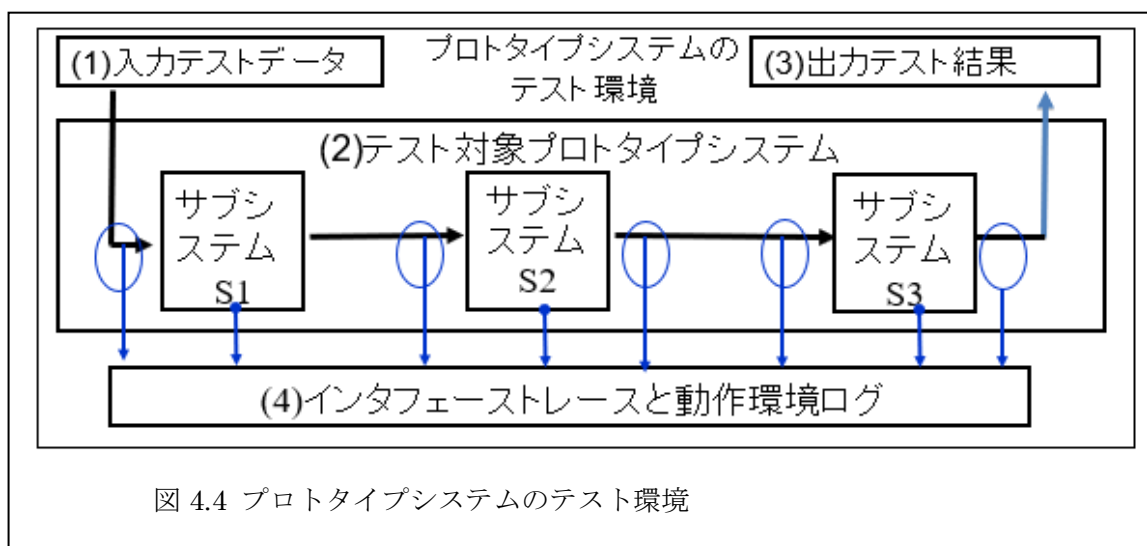


図 4.4 プロトタイプシステムのテスト環境

で処理を行い、(3)出力テスト結果を判定する。通信障害やクラウド上のアプリケーションの相互作用の発生の有無を確認するために、通信トレースとクラウド上の動作環境ログを採取する。図 4.4 の図中(4)はインタフェーストレースと動作環境ログ採取の箇所を示している。このような検証環境を設けることで、検証結果を判定できると考える。

4.3.2 プロトタイプ検証の検証項目

4.3.2.1 仕様達成度検証項目

仕様達成度の検証項目は、問題点を早めに検出するというプロトタイプの目的に合わせて基本的な実装アルゴリズムの確認項目である、機能と性能およびセキュリティ対策とする。仕様達成度の検証の観点とテストケースを表 4.1 に示す。

表中(a)機能テストの観点に対して、機能操作・インタフェース上の弱点・障害回復の 3 種類のテストケースを設定する。(b)性能評価は、目標性能達成、限界性能測定、長時間走行の 3 種類のテストケースを設定する。(c)セキュリティ対策については、IoT の一般的構成に対応したセキュリティ⁷⁾の脅威とその対策をテストケースとする。IoT の一般的構成に対応したセキュリティの脅威は、デバイスの盗難からネットワーク上のデータの盗聴・改竄、さらにゲートウェイやサーバへの不正アクセスや DDoS 攻撃(Distributed Denial of Service attack)によるネットワーク回線飽和などが考えられる。検証においては対策が考慮されていることの設計ドキュメント上での確認だけでなく、セキュリティ攻撃された結果の影響の大きさから、サーバ・ゲートウェイおよび制御端末について実装上の検証が必要と考える。商用のサーバに対して検証ツールを適用した場合、攻撃されたと認識される恐れがあるため、意図的な実施は困難であるが、サーバ・ゲートウェイ・制御端末に対してテストを実施しセキュリティ対策を確認することが有効である。

表 4.1 機能達成度検証の観点

テストの観点	テストケース
(a)機能試験	操作性
	弱点
	障害回復
(b)性能試験	目標値
	限界値
	長時間走行
(c)セキュリティ試験	通信路暗号化
	データ暗号化
	アンチウィルス
	ユーザ認証
	Dos攻撃

4.3.2.2 拡張容易性評価項目

プロトタイプから製品版への拡張容易性の評価項目は、表 4.2 に示す(a)機能と(b)性能とする。

表 4.2 拡張容易性の評価項目

テストの観点	評価項目
(a) 機能試験	基本設計
	実装の考慮
	変更範囲
(b) 性能試験	基本設計
	実装の考慮
	変更範囲

表中(a)機能の拡張容易性については、製品の仕様書である要件定義書とプロトタイプの仕様を記載した機能仕様書を比較する。拡張が必要な機能について、プロトタイプ機能仕様書に機能追加のための基本設計が記述されていること、および実装手段が考慮されていること、さらに機能追加修正による変更の影響がシステム全体に及ばず限定的であることを評価項目とする。(b)性能の拡張容易性については、<4.3.2.1>仕様達成度検証で測定したプロトタイプの限界性能結果をもとに製品の目標性能を達成するための、実装手段と変更範囲が限定的であることを評価項目とする。

4.3.2.3 ハザード対策検証項目

ミッションクリティカルな分野における事故モデルおよび安全解析手法として

STAMP(Systems Theoretic Accidental Model and Processes)/STPA (System-Theoretic Process Analysis) ⁽⁸⁾⁽⁹⁾⁽¹⁰⁾ が、注目されている。STPA は設計時のレビュー手法であるが、障害原因の解析手法としても利用されている⁽¹¹⁾ことから、IoT プロトタイプの検証において、STPA を利用しハザード発生条件を抽出してこれを検証項目とする。IoT システムの構成要素をコンポーネントとして、コンポーネントおよびコンポーネント間の相互作用を図 4.5 に示す。

図 4.5 の(a)から(i)に対応する、コンポーネント間の相互作用が正常に働かなくなるハザードの誘発要因を表 4.3 に示す。

図 4.5 と表 4.3 をもとに STPA の手法に従い、制御作用の発生タイミングごとに、想定されるハザードの誘発要因を洗い出すことができる。ハザード対策の検証項目として、このハザードの誘発要因を起こす動作条件を試験項目として設定する。

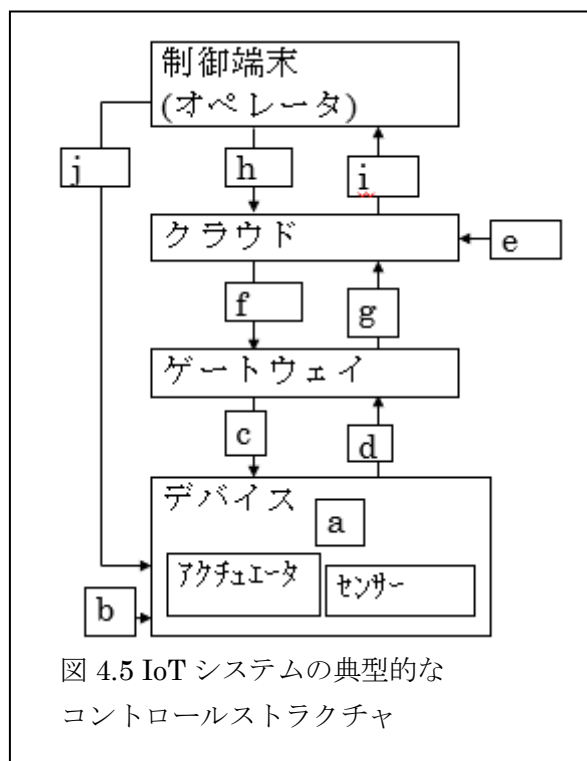


表 4.3 IoT システムの典型的な事故要因

事故要因
(a) デバイス故障
(b) 入力誤り
(c) 指示の誤り
(d) 不十分な動作
(e) 意図しない外乱
(f) 指示の誤り
(g) 不十分な動作
(h) 指示の誤り
(i) 不十分な動作
(j) 不適切な操作

4.4 実装

本検証方式を評価するための実装は、固形培地を必要としない植物の育成方法である水耕栽培と、ブドウ園に飛来するカラスを追い払う害鳥駆除 UAV(無人飛行機：ドローン) プロトタイプについて実施した。前者はデバイスが移動しないものであり、後者はデバイスが移動するものである。検証は、前者については開発者自身が行い、後者は第三者の視点をもった開発者以外のメンバが、開発者の協力を得て検証を実施した。

4.4.1 水耕栽培プロトタイプ

4.4.1.1 プロトタイプの開発環境

水耕栽培システムは、レタスなどすでに大規模なシステムが製品化されているが、葉物以外の植物についても、青・赤の光の照度と照射時間、温湿度と培養液をコントロールして生育期間を短縮し安定的な栽培を行う IoT システムが報告されている⁽¹²⁾。筆者らは様々な植物の生育データを蓄積する目的をもった実用システムの開発を目的として、デバイス・ゲートウェイおよびクラウドを用いてプロトタイプを開発した。クラウドはデバイス制御のアルゴリズムの確認と、長期間に渡り発生する植物の育成データの蓄積および分析するためのアプリケーションの追加という拡張容易性に着目して商用クラウドシステムである AWSIoT(Amazon Web Service IoT)⁽¹³⁾を利用した。

4.4.1.2 プロトタイプの機能

今回開発した水耕栽培プロトタイプのシステム構成を図 4.6 に示す。(1)デバイスは温湿度センサーと照度センサーおよびカラーLED という簡単なセンサーとアクチュエータで照度制御デバイスを作成し、育成対象となる植物と培養液のケースの上に設置する。培養液の灌水は人手で実施する。センサーで検知された温湿度および照度データは、(2)ゲートウェイ経由で(3)クラウドへ送信され、アプリケーションを用いて操作員が持つ(4)制御端末へデバイスの植物育成データとして表示される。操作員は育成計画に従って(4)制御端末からLEDの照度の変更指示を行う。LED照度の指示データは(3)クラウドへ送信され、アプリケーションを用いて(2)ゲートウェイ経由で(1)デバイスへカラーLEDの照度の値を送信する。温度管理は外部に設置されたクーラで制御する前提としているが、(3)クラウドアプリケーションでは、(1)デバイスで測定された温度が一定の閾値を超えると(4)制御端末へアラーム表示して、緊急クーラを起動させて温度を下げる操作の必要性を操作員に伝える。その後デバイスで測定された温度が閾値を下回ると(3)クラウドアプリケーションでは緊急クーラの停止指示を(4)制御端末へ通知する。

次にシステムのデータ処理方法を示す。(1) デバイスは Arduino⁽¹⁴⁾をベースに温湿度センサー・フルカラーセンサーおよびフルカラーLED と、ZigBee⁽¹⁵⁾通信モジュールで構成する。(2)ゲートウェイはPCをベースに Processing⁽¹⁶⁾上で動作する Java アプリケーションを用いて ZigBee 通信を介して 10 秒間隔でデバイスのセンサーデータを採取する。採取したデータは WiFi 通信上の MQTT プロトコルでインターネットを介してクラウドへ送信する。(3)クラウドは商用の AWSIoT を採用して実用システムにおけるプラットフォーム化を目指している。ゲートウェイからのセンサーデータ情報をデータベースへ登録し、インターネットを介し

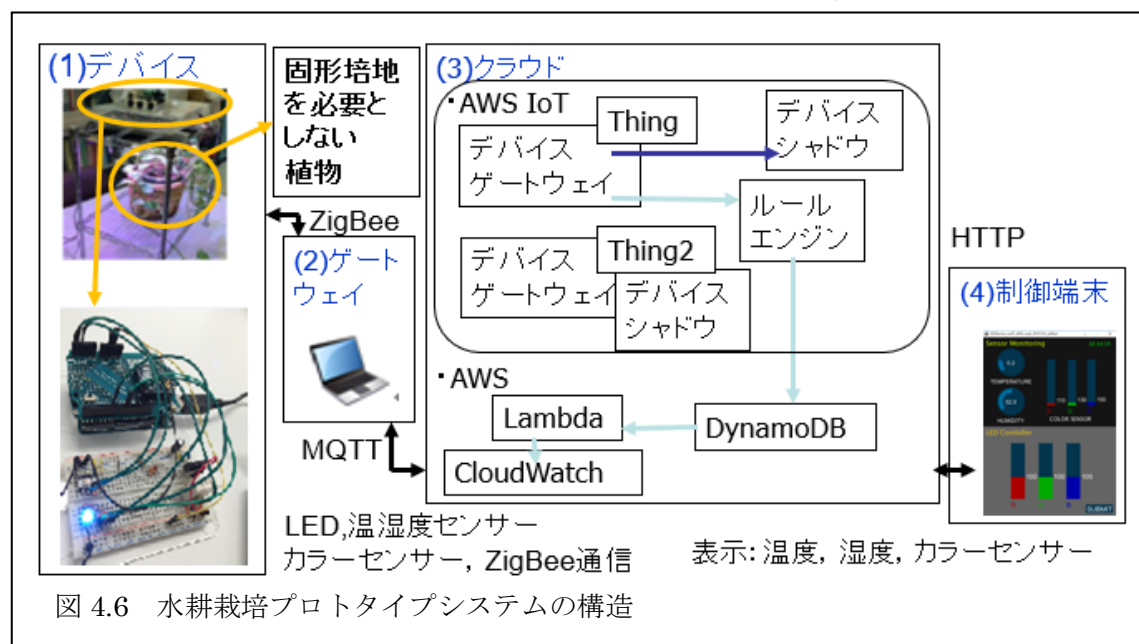


表 4.4 製品とプロトタイプ仕様比較（水耕栽培）

仕様		(A)製品	(B)プロトタイプ
デバイス 制御	光源	○	○
	温度	○	△
	湿度	○	△
	風力	○	—
	CO2	○	—
	養液温度	○	—
	養液pH	○	—
	養液EC	○	—
表示	センサーデータ表示	○	—
	カメラ映像表示	○	—
	育成制御	○	—
	保守	○	—
性能	接続台数	10台	1台
	送信データ量	200GByte	2MByte
セキュリ ティ	デバイス	専用機	Arduino
	エリアネットワーク	ZigBee	ZigBee
	ゲートウェイ	専用機	Windows PC
	インターネット	MQTT	MQTT
	クラウド	AWSIoT	AWSIoT
	制御端末	タブレット	Windows PC

- : 未実装
 ○ : 実装
 △ : 一部実装（センサーのみ）

て MQTT プロトコルで(4)制御端末へ送信するとともに、温度センサーの値が閾値を超えると緊急クーラ起動の指示を(4)制御端末へ送信する。制御端末は PC をベースに構成し、Processing 上の Java アプリケーションを用いて(3)クラウドから通知されたデバイスのセンサー情報を画面に表示する。画面から操作員が入力する RGB 別カラーセンサーの照度の値は(3)クラウドへ送信され、(2)ゲートウェイ経由で(1)デバイスのフルカラーLED を制御する。水耕栽培プロトタイプの仕様を製品仕様と対比させて、内容を表 4.4 に示す。表中(A)は製品の要件定義仕様、(B)はプロトタイプで実現している機能を示す。

4.4.2 害鳥駆除 UAV プロトタイプ

4.4.2.1 プロトタイプの開発環境

上空から飛来するカラスやムクドリなどによる果実の被害は大きく、対策として花火や防護ネットの設置などを行ってもすぐに慣れてしまい、効果が少ない。そこでこれら害鳥が恐れる鷹や鷲などの天敵を、UAV(無人飛行機：ドローン)で模擬するシステムが報告されている⁽¹⁷⁾。筆者らは特定の果樹園に対応して、害鳥駆除への有効性を評価する目的でプロトタイプを開発した。まず安定飛行するドローンを作成し、次に監視カメラと組み合わせて、害鳥の侵入を監視して検出するとドローンを発進させて予め設定されたルートを飛行して戻ってくる、という機能を実装した。

4.4.2.2 プロトタイプの機能

害鳥監視・防除 UAV プロトタイプのシステムの構成を図 4.7 に示す。(1-2)は固定設置された監視カメラで、害鳥の出現を検出する。検出した画像は(3)サーバへ送信され、アプリケーションを用いて操作員が持つ(4)制御端末へ害鳥の検出を通知する。通知を受け取ると操作員が(1-3)プロポ(送信機)を用いて (1-1)デバイス UAV を発進させる。(1-1)UAV は予め設定されたルートを周回飛行して戻ってくる。次にシステムの詳細構成とデータの処理方法を Fig.7(b)に示す。(1-2)監視カメラでは、5 秒間隔で画面をキャプチャーし背景差分法を用いて害鳥の出現を検出し、検出した結果をゲートウェイを介して HTTP プロトコルで (3)サーバへ送信する。サーバでは害鳥侵入通知をもとに、インターネットに接続された (4)制御端末 (スマートホン) へメールで害鳥侵入通知情報を送信する。そこで、操作員は (1-3)UAV 発進装置を用いて ISM (Industry-Science-Medical)Band で接続した (1-1)UAV を発進する。UAV は搭載したセンサーで、高度・GPS・温度気圧、さらに UAV ロータ停

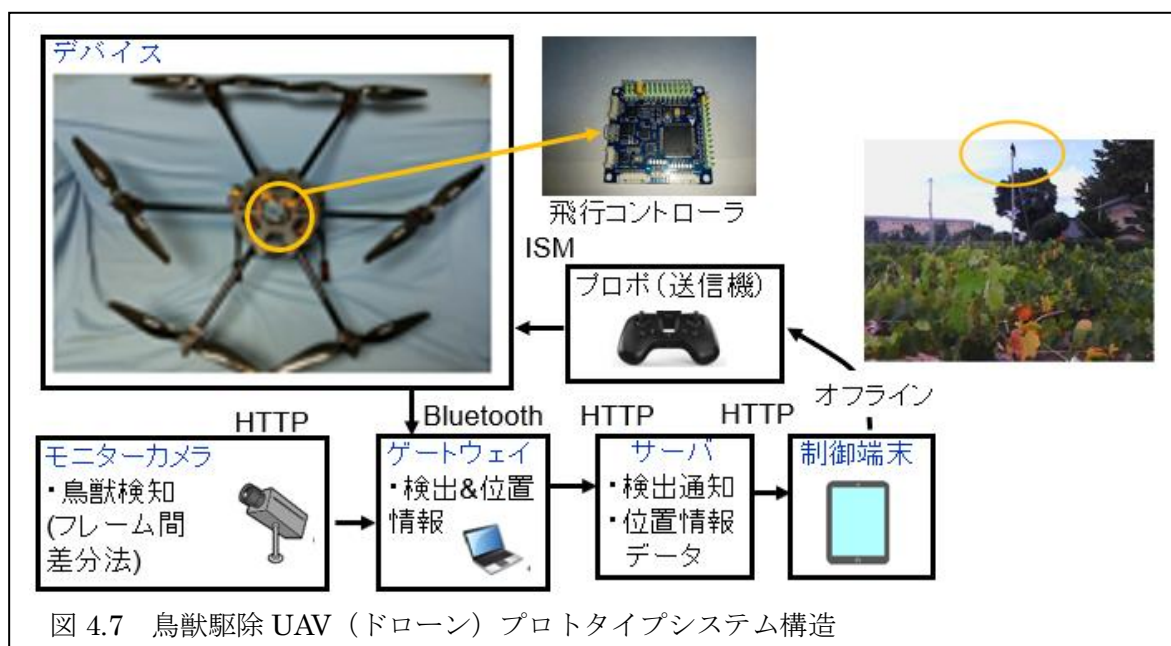


図 4.7 鳥獣駆除 UAV (ドローン) プロトタイプシステム構成

表 4.5 製品とプロトタイプの仕様比較 (害鳥駆除 UAV)

仕様		(A)製品	(B)プロトタイプ
機能 (UAV)	安定飛行	○	○
	荷重	○	○
(コントローラ)	飛行制御	○	○
	3軸ジャイロ	○	○
	3軸角度センサ	○	○
	3軸磁気センサ	○	○
	気圧センサ	○	○
	ロータ障害検知	○	—
(カメラ)	鳥侵入検知	○	○
サーバ	鳥侵入通知	○	○
	飛行位置表示	○	—
	部品交換通知	○	—
	自動飛行経路制御	○	—
性能 (UAV)	飛行時間	30分	8分
	同時飛行台数	1台	1台
(カメラ)	連続稼働	1ヶ月	1週間
セキュリティ	UAV	DJI/RCTimer	DJI/RCTimer
	カメラ	Buffalo	Buffalo
	エリアネットワーク	Bluetooth	Bluetooth
	ゲートウェイ	専用機	Windows PC
	インターネット	HTTP	HTTP
	クラウド	AWSIoT	サーバ
	制御端末	スマートホン	スマートホン

— : 未実装
 ○ : 実装
 △ : 一部実装

止検出を行い、Bluetooth で接続された (2)ゲートウェイにこれらセンサーデータを送信する。ゲートウェイでは UAV で収集したデータをインターネットを介して HTTP プロトコルで (3)サーバへ送信する。サーバでは、UAV が収集した飛行位置座標および日時の天候情報をデータベースへ保管するとともに、UAV ロータ動作情報を故障診断情報として保管する。今回開発した害鳥駆除 UAV プロトタイプの仕様を製品仕様と対比させて表 4.5 に示す。表中 (A)は製品の要件定義仕様、(B)はプロトタイプで実現している機能を示す。

4.5 テスト結果と評価および考察

4.5.1 水耕栽培プロトタイプの実験結果と評価

4.5.1.1 仕様達成度および拡張容易性評価項目

仕様達成度検証を行うため、プロトタイプの仕様要件である<4.4.1.2>表 4.4 に基づいて、<4.3.2.1>表 4.1 に示した機能・性能・セキュリティのテストの観点に従いテスト項目を設定してテストを実施した。セキュリティについては、ゲートウェイと制御端末に対して Nmap⁽¹⁸⁾を用いてポートスキャンを行い、設計ドキュメントどおりにポートの設定が行われていることを確認した。サーバについては商用クラウドサーバを採用していることから、検証目的が攻撃とみなされる恐れがあるため実施しなかったが、ゲートウェイプログラムの実装誤りにより、諸評価中に予期せずサーバへの MQTT 通信が時間間隔を空けずに繰り返される状況が発生し、サーバの管理者から問い合わせを受けた。実際に異常に多いサーバのアクセスランザクション数が報告され、商用サーバは問題なく動作したことが確認された。

プロトタイプから製品への拡張容易性について、表 4.4 に基づいて評価の観点を設定した。デバイス機能の拡張要素としては、植物育成要素に対応するセンサーおよびアクチュエータの増加、監視カメラの設置、植物の育成環境である温度・湿度・風・培養液の供給の自動化、密封された植物生育棚などが想定される。性能の拡張要素としては、植物育成データ量と解析アプリケーション本数の増加が想定され、これらについては採用しているクラウドのインフラ機能および処理性能について<4.3.2.2>表 4.2 に示した評価項目に基づいてテストを行った。

4.5.1.2 ハザード対策検証項目

安全性の損失につながるハザードとしては、水溶液の溢れによるデバイス機能損失事故が考えられるが、今回のプロトタイプはデバイス培養液の灌水は自動制御の機能は実装せず、人手で実施している。このため温度センサーの値が閾値を超えた場合に緊急冷房機能が期待どおり機能しないケースをハザードとして設定した。水耕栽培のコントロールループを図 4.8 に示す。

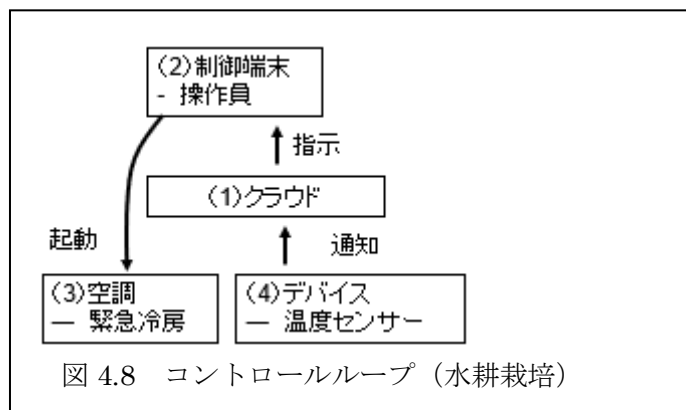


表 4.6 緊急冷房が期待通りに機能しない UCA 条件

4つのガイドワード	安全でないコントロールアクション	
1.安全のためのコントロールアクションが設置されていない	UCA1-(1)	クラウドは温度が上限閾値を超えても、制御端末へ冷却通知しない。
	UCA1-(2)	制御端末に冷却通知がきても、操作員は緊急冷房を起動しない。
	UCA1-(3)	緊急空調の冷却機能が働かない。
2.安全でないコントロールアクションが設置されている。	UCA2-(1)	クラウドは温度が上限閾値を超えていないのに、制御端末へ通知する。
	UCA2-(2)	操作員が、冷却通知がないのに緊急冷房を起動する。
3.コントロールアクションのタイミングが、早すぎる/遅すぎる。	UCA3-(1)	クラウドは温度が上限閾値を超えても、すぐには制御端末へ冷却通知しない。
	UCA3-(2)	制御端末に冷却通知がきってから、操作員が緊急冷房を起動するのが遅い。
4.コントロールアクションがすぐ止まる/適用が長すぎる。	UCA4-(1)	十分温度が下がる前に操作員が緊急冷房を停止する。

(4) デバイスのセンサーから温度情報が(1)クラウドに送信される。クラウドでは温度と閾値を比較し、閾値を超えた場合(2)制御端末へ緊急クーラ起動通知が送られる。この情報に基づいて操作が(3)外部のクーラを起動し温度を低下させるというコントロールアクションの流れとなる。次にコントロールアクションのタイミングを4つの条件に分けて、緊急冷房が期待どおり機能しないというハザードが発生するUCA(Unsafe Control Action)条件を抽出する。表4.6にUCA1-(1)からUCA4-(1)までの8種類のハザードが発生するコントロールアクションを示す。

抽出した安全でないコントロールアクションによりもたらされる、緊急冷房機能が期待どおり動作しないというハザードの発生原因(HCF: Hazard Causal Factor)と、発生させるテストケースを表4.7に示す。

表中の通信障害/回復テスト、および閾値付近で連続した温度の上下動の2項目についてテストを実施、操作が員緊急冷房起動通知に気付かない、および緊急クーラの故障という2項目のハザード発生原因については、設計上の誤り防止の対策をドキュメント上で確認した。

表 4.7 HCF(ハザード発生要因)テストケース（水耕栽培）

HCF	UCA 1-(1)	UCA 2-(1) 3-(1)	UCA 1-(2) 2-(2) 3-(2) 4-(1)	UCA 1-(3)	テストケース 実施項目, あるいは (ドキュメントチェック)
1.冷却通知が実行されない	○				通信障害と回復
2.不正なタイミングで冷却通知される。		○			温度が閾値の前後で上下する
3.操作員が冷却通知を誤解する。			○		(操作誤りの防止策)
4.緊急冷却装置故障				○	(緊急冷却装置の点検策)

4.5.1.3 検証結果の評価

仕様達成度評価テストと拡張容易性評価, およびハザード対策テストケースを実施した結果を, 合格件数/試験項目件数として表 4.8 に示す。テストはシステム全体を対象に実施するが, 評価は構成要素毎に示しハザード対策はシステム全体で表す。

製品化へ向けての拡張容易性については, デバイスおよびゲートウェイの拡張容易性の評価結果値が低いことがわかる。デバイスはブレッドボードに部品を搭載したものであり実装面での評価が低い。ゲートウェイは PC ベースであることから, 製品版にするには CPU および OS を変更する必要がある実装面での評価が低い。クラウドは製品版の AWSIoT を用いていることから, 蓄積データやアプリケーション数の増加に対する拡張容易性については問題ない。制御端末については製品版も Windows ベースのタブレットを用いることが可能であり拡張容易性について問題ない。

ハザード対策については, 操作員が緊急クーラ起動の指示に気付かないケース, および緊急クーラが起動されないケースで対策に問題がある。デバイスの温度が閾値付近で頻繁

表 4.8 テスト結果（水耕栽培）

テスト対象	機能達成度	機能拡張性	ハザード対策
デバイス	10/15	1/5	---
ゲートウェイ	20/20	5/10	---
クラウド	20/20	10/10	---
制御端末	5/5	5/5	---
全体システム	55/60	11/30	7/9

に上下するというテストケースについて問題点は検出されていないが、温度が頻繁に上下するケースに対する設計上の考慮が必要であると評価した。

4.5.2 害鳥駆除 UAV プロトタイプのテスト結果と評価

4.5.2.1 仕様達成度および拡張容易性評価項目

仕様達成度検証を行うため、プロトタイプの仕様要件である<4.4.2.2>表 4.5 に基づいて、<4.3.2.1>表 4.1 に示した機能・性能・セキュリティのテストの観点に従いテスト項目を設定してテストを実施した。セキュリティについては、ゲートウェイとサーバに対してポートスキャンツールの Nmap⁽¹⁸⁾を用いてポートスキャンを実施し、設計ドキュメントどおりの設定が行われていることを確認した。ユーザ評価は、技能検証の受講経験者が操作を行い、飛行に関する安全監視者を 1 名設置し、法令で定められている 30m の距離に人が入らないブドウ園という条件で実施した。

プロトタイプから製品への拡張容易性について、表 4.5 に基づいて評価の観点を設定する。クラウド機能の拡張要素としては、クラウド上のアプリケーションの数とデータベース容量の増加が想定される。性能の拡張要素としては、UAV と監視カメラの接続台数の増加、および UAV の飛行時間の延長が想定され、これらについては採用しているクラウドのインフラ機能および処理性能、さらに UAV の充電機能について<4.3.2.2>表 4.2 に示した評価項目に基づいてテストを行い、不明点は開発者へインタビューを実施し補った。

4.5.2.2 ハザード対策検証項目

安全性の損失につながるハザードとして UAV の落下事故が考えられる。UAV 本体に、自己診断を行い故障検出時あるいは外部からの指示で、緊急着陸機能が実装されていることを想定する。サブシステム単体の障害は単体試験で検出されるという前提で、今回はサブシステムの故障がなくてもシステムとして障害が発生するケースをハザードとして設定した。害鳥駆除 UAV の緊急着陸制御のコントロールループを図 4.9 に示す。

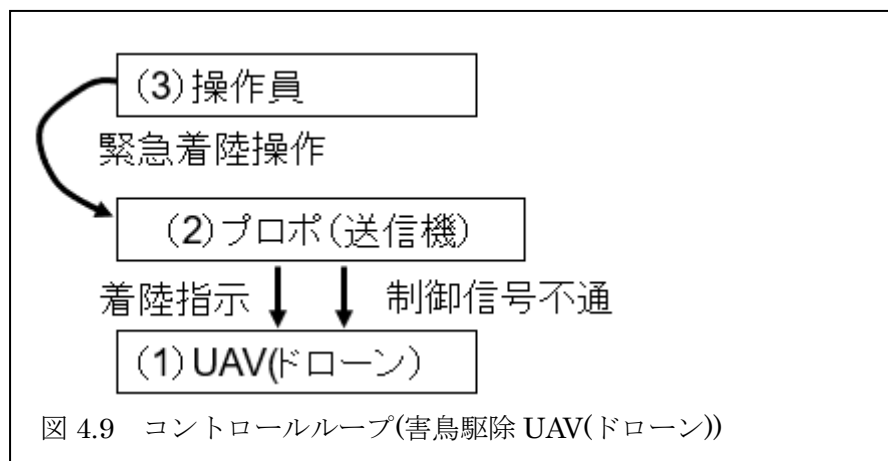


表 4.9 UCA 条件（ドローンの緊急着陸）

4つのガイドワード	安全でないコントロールアクション	
1.安全のためのコントロールアクションが設置されていない	UCA1-(1)	ドローンに緊急着陸機能が実装されていない。
	UCA1-(2)	プロポ(送信機)から緊急着陸指示が出力されない。
	UCA1-(3)	なし
2.安全でないコントロールアクションが設置されている。	UCA2-(1)	なし
	UCA2-(2)	制御信号不達を不正に検知する。
	UCA2-(3)	通常飛行中に操作員が不正に緊急着陸指示を行う。
3.コントロールアクションのタイミングが、早すぎる/遅すぎる。	UCA3-(1)	なし
	UCA3-(2)	なし
	UCA3-(3)	操作員の緊急着陸指示が遅れる。
4.コントロールアクションがすぐ止まる/適用が長すぎる。	UCA4-(1)	なし
	UCA4-(2)	なし
	UCA4-(3)	なし

(3)操作員が雷雨発生など天候の急変時に緊急に UAV を着陸させる必要を認識した場合、
 (2)UAV 発進装置へ緊急着陸指示を出し (1)UAV が緊急着陸する、というコントロールアクションの流れとなる。次に制御アクションのタイミングを4つの条件に分けて、UAV が期待どおり動作しないというハザードが発生する UCA 条件を抽出する。表 4.9 に安全でないコントロールアクションUCA1-(1)からUCA3-(3)までの4種類のハザードが発生するコント

表 4.10 HCF(ハザード発生原因：ドローン)

HCF	UCA 1-(1)	UCA 1-(2)	UCA 2-(3) 3-(3)	UCA 2-(2)	テストケース 実施項目、あるいは (ドキュメントチェック)
1.ドローンに緊急着陸機能が実装されていない。	○				(実装の確認)
2.プロポ(送信機)から緊急着陸指示が出力されない。		○			通信障害
3.操作員が誤って緊急着陸操作を行う。			○		(操作誤りの防止策)
4.プロポ(送信機)から緊急着陸指示が誤って送信される。				○	通信障害と回復

表 4.11 テスト結果（ドローン）

テスト対象	機能達成度	機能拡張性	ハザード対策
デバイス	15/15	4/5	---
ゲートウェイ	20/20	9/10	---
サーバ	20/20	9/10	---
制御端末	5/5	5/5	---
全体システム	60/60	27/30	4/5

ロールアクションを示す。

抽出した安全でないコントロールアクションによりもたらされる UAV が期待どおり動作しない、というハザードの発生原因(HCF) と、発生させるテストケースを表 4.10 に示す。

表中の通信障害/回復テストの 2 項目についてテストを実施, UAV の緊急着陸機能を実装, および操作員の誤操作と操作が遅れるという 2 項目のハザード発生原因について機能実装および誤り防止の設計上の対策についてドキュメント確認を実施した。

4.5.2.3 検証結果の評価と製品完成までの距離算出

仕様達成度評価テストと拡張容易性, およびハザード対策テストケースを実施した結果を, 合格件数/試験項目件数として表 4.11 に示す。

テストはシステム全体を対象に実施するが, 評価は構成要素毎に示しハザード対策はシステム全体で表す。

拡張容易性については, 監視カメラで行っている害鳥検出の精度向上について設計上の対応が必要である。

ハザード対策については, UAV 本体に, 自己診断を行い故障検出時あるいは外部からの指示で緊急着陸および通信障害時の緊急着陸指示機能が実装されていた。操作員の誤操作による緊急着陸の防止手段が必要であると評価した。

4.5.3 考察

本論文で提案した IoT プロトタイプシステムの第三者的検証方式を, 水耕栽培プロトタイプと害鳥駆除 UAV プロトタイプに対して実施した評価結果を表 4.12 に示し, 各検証項目と開発不足項目指摘の結果について考察する。

(1) 仕様達成度 機能・性能・セキュリティについて試験要領書を作成して試験を実施した。またテスト実施が困難な項目についてはプロトタイプ機能仕様書を確認し, 有効な試験結果を得ることができたと考える。これはプロトタイプ製品に対して製品の要件

表 4.12 本方式の総合評価

本手法の総合評価		
評価項目	水耕栽培	ドローン
(1) 機能達成度	55/60	60/60
(2) 拡張容易性	11/30	27/30
(3) ハザード対策	7/9	3/5
(4) 不足項目の指摘	<ul style="list-style-type: none"> ・自動灌水, Webカメラ, 送風機の追加時の機構設計不足。 ・商用クラウドの課金制限について, 設計時の考慮。 	<ul style="list-style-type: none"> ・飛行時間拡大 ・UAV通信障害時の緊急着陸機能, オペレータの操作誤り防止機能について, 設計時の考慮。

定義書相当の機能仕様書を開発者が作成したことから, 開発者とは異なるメンバによる検証が可能となった効果であると言える。

(2) 拡張容易性 水耕栽培プロトタイプでは表 4.4 に示したように実装される機能が少なく, <4.1.2>に記載したように密封された植物生育棚などが実装されていないことから, デバイスの評価結果は表 4.8 に示すように低い。またゲートウェイはプロトタイプでは汎用製品である PC ベースで作成されているが製品版では専用製品となることから, 製品版との差という意味で拡張容易性は低いという検証結果を得た。デバイスおよびゲートウェイについてさらに製品版に近いプロトタイプを作成する必要があるという結論を得た。

害鳥駆除 UAV プロトタイプでは, 表 4.5 に示したように製品版に近い機能が実装され, プロトタイプをもとに監視カメラ 1 台を複数台に拡張して製品開発を継続し, 製品版にできるという結論を得た。

一方, 飛行の安定性・飛行コースなどの, 機能追加手順および実装手順についてドキュメント評価では検証できない試験項目が残った。このため開発者へ試験項目をヒヤリングすることで評価を補ったことは, 開発者と同程度の技術を保有していない第三者の検証実施の限界と考える。

(3) ハザード対策 今回のハザード対策評価において, 水耕栽培では, 温度閾値の前後の値が連続して検出される場合にデバイスの温度制御がうまく動作しないケースについて検証した。問題点は検出されなかったがテスト条件が十分でない可能性もある。但し設計時に考慮すべき項目であると考え。

害鳥駆除 UAV では, 表 4.10 に示したように, 通信障害により緊急着陸制御がうまく動作しないケースについて試験項目を設定した。問題点は検出されなかったが, UAV 本体に故障検出時あるいは外部からの指示で緊急着陸および通信障害時の緊急着陸指示が実装されているため, どのような条件で動作したかの切り分けができなかった。

水耕栽培および害鳥駆除 UAV とともに、テストで障害を検出することは困難であり、設計レベルで考慮すべき項目を指摘するに留まったが、設計時に考慮すべき項目の抽出ができたことは有効であると考える。

(4) 不足項目の指摘 プロトタイプをそのまま開発を継続して製品化するにあたり、プロトタイプの段階で不足する項目としてつぎの指摘事項を得た。

水耕栽培プロトタイプでは、自動灌水装置および Web カメラさらに送風機の各装置をデバイスに追加するための、デバイス装置の筐体の機構設計が必要。商用クラウドを利用する際に課金される利用料について考慮したクラウドアクセス制御の設計が必要。

害鳥駆除 UAV プロトタイプでは、バッテリー重量を考慮した飛行時間拡大の設計が必要。着陸時にオペレータの誤操作を防止する機能について、設計時の考慮が必要。

4.5.4 従来の第三者検証との違い

従来のシステムに対する第三者検証方式は、検証項目に対する製品の合格/不合格を判定し、製品全体が検証基準を満足しているかどうか、判定することを目的としている。本論文で提案しているプロトタイプに対する第三者的検証は、プロトタイプから開発を続けて製品版とするために必要な不足部分を指摘することを目的としており、この点が製品に対する従来の第三者検証と異なる。

4.6 まとめ

本研究では、開発部門でプロトタイプについて製品と同程度のプロトタイプ機能仕様書を作成することで製品との比較を可能とし、さらに検証環境および評価項目と判定基準を明らかにした検証計画書を作成することで、開発者による第三者的検証を可能とすることを目標とした。その結果、図 4.10 に示すように、民生品 IoT プロトタイプに対する製品完成までの「不足項目の指摘機能をもつ第三者的検証方式」を提案し、水耕栽培プロトタイプおよび、害鳥駆除 UAV（ドローン）プロトタイプへ適用し、本手法が有効であるという成果を得た。

本章では、民生品IoTプロトタイプに対する製品完成までの不足項目の指摘機能をもつ第三者的検証方式を提案し、実装評価を行った。

成果 : ・プロトタイプの第三者的検証の実施を可能とした。

成果物: ・製品への**拡張性とハザード対策**を評価項目とする,
「プロトタイプ検証設計書」

図 4.10 研究の成果

5. IoT システムのプロトタイプの第三者的検証による開発の方向付け

本章では、IoT システム開発におけるプロトタイプの第三者的検証による開発の方向付けの方式を提案する。IoT システムは新たな利用分野を開拓すると期待されることから、市場への製品投入のタイミングとユーザ要求仕様の把握が重要と考えられる。このため、プロトタイプを客観的に評価した結果に基づいて、プロトタイプから製品完成までの開発の難易度を数値化して距離として表し、さらにユーザ要件定義と対応する開発項目を確定し、プロトタイプから製品への方向付けを行う方式の確立が求められる。

本研究では、IoT プロトタイプシステムの第三者的検証の結果に基づいて、プロトタイプをそのまま継続開発して製品化するまでの難易度を数値化する方式、および想定するユーザの人数・性別・年齢などを限った限定ユーザによるプロトタイプの操作結果からユーザ要件定義と開発項目を確定する方式の提案を目的とする。具体的には、「プロトタイプから製品完成までの距離の算出方式」および「プロトタイプの限定ユーザ評価方式」を提案する。

5.1 初めに

宇宙・航空・自動車分野などのミッションクリティカルな分野の製品に対しては、開発者は設計段階から機能要素やインタフェース上の不具合を検出する設計手法を用いている。例えば、AADL (Architecture Analysis & Design Language) ⁽¹⁾や、事故原因の解析手法である STAMP(System Theoretical Accident Model and Processor) ⁽²⁾⁽³⁾の手法に基づいて、設計時点で発生が予想される事故要因を解析する設計手法である STPA ⁽²⁾⁽³⁾などが適用されている。JAXA 出版社の IV&V プロセス ⁽⁴⁾⁽⁵⁾による第三者検証では、開発部門が作成する設計書を対象に、上位設計書に基づいて第三者が設計レビューを行う具体的手法が報告されている。

民生品に対して、筆者らは、第三者検証で検出した不具合の推定を含んだシステム製品における第三者検証方式を発表した ⁽⁶⁾。本方式は、IoT システムの動作はサブシステムの連携として表すことができる点に着目して、第三者検証で検出した障害結果からサブシステムの連携を遡ることで、障害発生箇所であるサブシステムとその原因を推定するものである。第三者検証で検出された障害について、障害修正を行うために開発者は障害発生の再現と解析を実施する必要があるが、サブシステム単位で障害の発生箇所と原因が第三者検証の結果として示すことができると、開発者の作業が容易になると予想される。

IoT(Internet of Things)システムでは、データ発生現場にあるセンサーやアクチュエータとクラウドがネットワークで接続され、データの伝達、分析、結果のフィードバック動作などが人手を介さずに行われる。IoT システムは構成要素が多様であり、またセンサーやアクチュエータという物理的な(Physical)ものとクラウド上の仮想的な(Cyber)もので構成されるという特質がある。このように技術および適用の範囲が広いため、IoT システムの製品化には、製品の狙いと基本となる技術基盤を決定し、解決すべき課題と目標を定めた方向づけが必要になる。IoT システムのプロトタイプを作成して、早い段階で主要仕様の達成度を検証すれば、完成までにやるべきことが明確になり、製品化までの期間の短縮が可能になる。

そこで、プロトタイプに対してそのまま機能・性能の追加開発を行うことで製品化できるか否かを客観的に評価することが求められる。筆者らは、IoT プロトタイプシステムを対象に第三者的検証を行い、検証結果に基づいてプロトタイプから製品完成までの開発の難易度を距離として算出する手法を発表した⁽⁷⁾⁽⁸⁾。

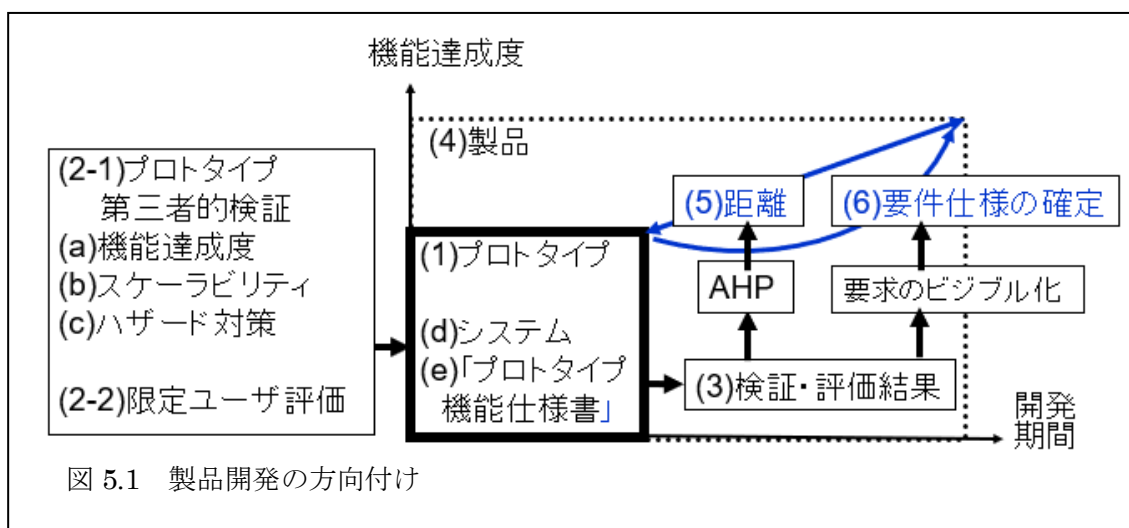
しかしながら、製品仕様は開発当初に定められたものであり、ユーザ要求を満たしているか、アピール力ある特徴となっているか明確でない。そのため本研究ではプロトタイプ検証を行った段階で、プロトタイプを対象としてユーザを限定した市場の反応の確認を行い、プロトタイプ開発の過程で得られた開発者の気づきと合わせて、利用目的と実現方式を明らかにして、製品仕様の確定を行う方式を提案する。

本研究では、IoT システムのプロトタイプに対する第三者的検証による製品仕様の確定手法を、水耕栽培とドローンによる害鳥駆除プロトタイプシステムに対して適用し、本手法の有効性を確認した。

5.2 IoT プロトタイプに対する第三者的検証による製品開発への方向付け

5.2.1 プロトタイプから製品への方向付け

IoT システムの製品化には、製品の狙いと基本となる技術基盤を決定し、解決すべき課題と目標を定めた開発の方向づけが必要である。プロトタイプを第三者的に検証し、プロトタイプをそのまま機能・性能を追加開発して完成品にできるか否か開発の容易性を把握するとともに、限定ユーザによるプロトタイプ検証により製品仕様を確定することで、開発の方向付けを行うことができる。この開発の方向付けの手法を図 5.1 に示す。図中(1)プロトタイプに対して、(2-1)第三者的検証および(2-2)限定ユーザによるユーザ評価を実施し、(3)検証・評価結果を得る。この検証結果から、(1)プロトタイプから(4)製品完成までの(5)距離と、ユーザ要求と実現方式が明確な(6)要件仕様を得る。(2-1)プロトタイプの第三者的検証は、4 章で実施したプロトタイプ検証であり、テストの観点(a)機能達成度、(b)スケーラビリティ



ビリティ, (c)ハザード対策である。

5.2.2 プロトタイプに対する第三者的検証による製品完成までの距離の算出手法

プロトタイプ製品をそのまま追加開発を行い製品完成までの距離の算出を, 客観的に合否判定されたテスト結果および, 拡張容易性についてのドキュメント評価という主観的判断に基づいて行う。これには, 問題解決型意思決定手法の一つである AHP(Analytic Hierarchy Process)⁽⁹⁾⁽¹⁰⁾を用いる。AHP は製品の品質分析手法としても利用が報告されている⁽¹¹⁾。プロトタイプ第三者的検証で用いた, つぎの3つの検証の観点を AHP の評価基準とする。

- (1) プロトタイプとしての機能の達成度
- (2) 製品への拡張容易性
- (3) ハザード対策

上記3項目の評価基準に対して IoT 製品の検証という観点から, AHP の一対比較を行った結果の重みベクトルを表 5.1 に示す。表中 λ_{\max} は一対比較マトリックスの最大固有値を示す。C.I.(Consistency Index)は整合度指数を示し, 0.1 以下であることから, AHP 行列の整合性に問題はないことを示している。

次に, 3項目の試験の観点に基づいて行った検証結果を, 評価基準に対する評価水準として「良い, 普通, 不十分」という3段階に分け, AHP の一対比較を行った結果の重みベクトルを表 5.2 に示す。

3項目の各評価基準に対する評価水準を全て同じとした。評価基準に基づいて実施する検証結果は, 合格件数/全検証項目数として客観的に表し, 評価水準の3段階に従って評価する。3段階の評価水準は評価対象となる製品の使用環境に応じて設定する。

表 5.1 評価基準の一対比較の重みベクトル

評価基準	機能達成度	拡張容易性	ハザード対策	重みベクトル
機能達成度	1	3	5	0.637
拡張容易性	1/3	1	3	0.258
ハザード対策	1/5	1/3	1	0.105
$\lambda_{\max} = 3.307$, C.I. = 0.018 < 0.1 (有効)				

表 5.2 評価水準の一対比較の重みベクトル

評価水準	良い	不通	不十分	重みベクトル
良い	1	3	5	0.627
普通	1/3	1	4	0.280
不十分	1/5	1/4	1	0.094
$\lambda_{\max} = 3.084$, C.I. = 0.042 < 0.1 (有効)				

AHP で用いる代替案を、製品とプロトタイプの 2 種類として、代替案間での比較に関して絶対比較を用いる。製品とプロトタイプの評価を、3 つの評価基準について実施した検証結果を表 5.2 の評価水準を用いて評価し、総合評価値を求める。

この手順を表 5.3 を用いて説明する。表 5.3(a)(1)に示すように、製品は全ての評価基準に対して「良い」が当てはまる。プロトタイプについては、各評価基準に対するプロトタイプ試験結果を評価水準に従い判定し「良い、普通、不十分」の結果を求める。判定した評価結果から表 5.2 の評価水準に対応した「良い、普通、不十分」の重みベクトル値 $ELev(x)$, $Elev(y)$, $ELev(z)$ を得る。表 5.3 (a)(2)プロトタイプの行には「良い」の重みベクトル値(0.627)で割り正規化した値(x),(y),(z)を入れる。この結果に表中(3)に示す各評価基準の重みを掛け合わせ表 5.3 (b)(2)プロトタイプの行に記入し、さらに合計値を求めて表 5.3 (b)(2)の総合評価値を得る。「製品」の総合評価値は「1」で表されることから、「プロトタイプ」の総合評価値を 100 倍すると、プロトタイプ検証結果を「製品」に対するパーセンテージで表すことができこの値を製品完成までの距離とする。

上記手順によりプロトタイプ作成時点で、プロトタイプから製品完成までの「距離」という概念を導入して、開発の容易性を表した。しかしながら、この時点の製品仕様は開発開始時点で定義した要件定義であることから、ユーザの利用目的に合っているか、開発者

表 5.3 絶対評価の手順

(a) 絶対比較表

比較代替案	機能達成度	拡張容易性	ハザード対策
(1)製品	良い= 0.627 /0.627	良い= 0.627 /0.627	良い= 0.627 /0.627
(2)プロトタイプ	(x)= $ELev(x)$ /0.627	(y)= $ELev(y)$ /0.627	(z)= $ELev(z)$ /0.627(3)
(3)評価基準の 重みづけ	0.687	0.258	0.105

(b) 評価点

比較代替案	機能達成度	拡張容易性	ハザード対策	総合評価値
(1)製品	0.687	0.258	0.105	1
(2)プロトタイプ	(x)*0.687	(y)*0.258	(z)*0.105	合計値

の意図が反映されているかを評価し、さらに実現方式を対応させた製品仕様の確定が必要である。

5.2.3 ベータ版のユーザ評価

プロトタイプを開発途中のベータ版として、プロトタイプのユーザに適した人数、性別、経験などを限定したユーザに評価してもらおうと、ユーザの評価結果を製品版に取り入れることが可能となり、製品開発の方向付けに有効な効果をもたらすと言える。一方、プロトタイプをユーザにベータリリースするには、使用するユーザや動作する環境への安全性への配慮が必要となる。このためベータ版のユーザ評価の仕方には次の方法をとる。

- (1) 評価ユーザが検証環境の場に来て、ベータ版を評価する。
 - (2) ベータ版の使用条件を限定しユーザの利用する環境に、ベータ版を持込み評価する。
- 上記(1)の場合は試験実施時と同じ環境で評価される。

上記(2)の場合はハードウェアを含むシステムの安全性の確保に加えて、使用するエリアおよび使用方法に関して法令に基づいた許可・承認申請など、実施にあたり対応する法令を考慮する必要がある。

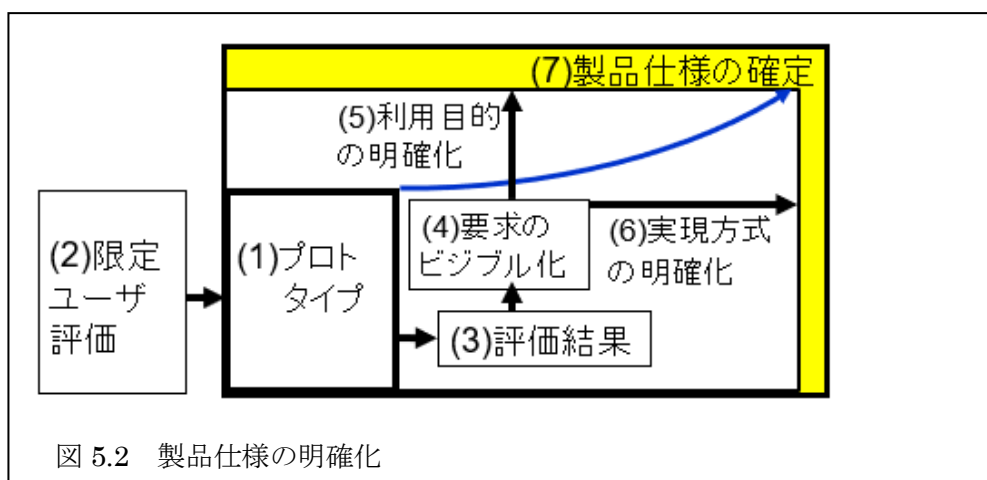
5.2.4 プロトタイプに対する第三者的検証による製品仕様の確定

IoT システムのプロトタイプに対して限定ユーザによる試使用評価を行い、評価結果をもとに、開発者がユーザ要求のビジブル化と開発課題の抽出を行い、製品仕様を明確化する。明確化する項目はつぎの2項目である。

- (a)利用目的の明確化
- (b)実現方式の明確化

この方式の概要を図 5.2 に示す。

図中(1)のプロトタイプに対して、(2)限定ユーザによるユーザ評価を行い、(3)評価結果を得る。これをもとに限定ユーザの特徴票、ユーザ要求のユーザストーリー票を作成し (4)ユー



ザ要求をビジブル化する。ユーザストーリー票から(5)利用目的の明確化および(6)実現方式の明確化を行い、(7)確定製品仕様を得る。

ユーザ評価は、プロトタイプを限定ユーザが一定期間操作した後に、開発者がユーザへインタビューする形で行う。インタビューから製品仕様の確定までの手法はつぎの 3 ステップで構成する。

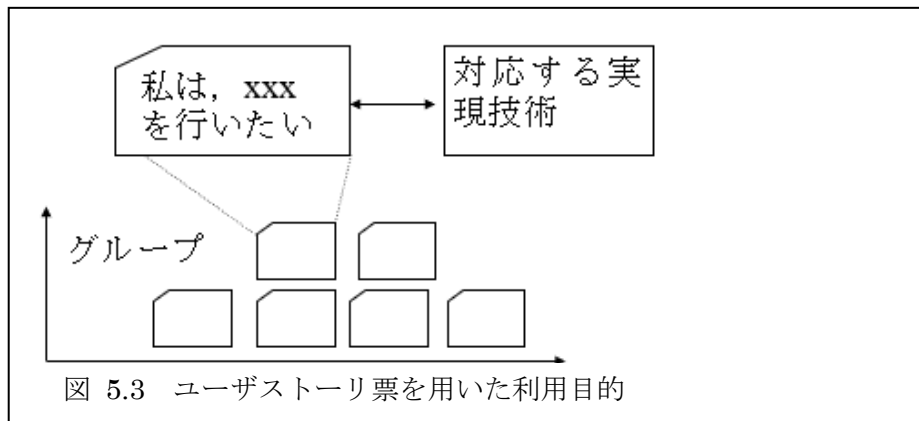
1. 限定ユーザの評価項目（操作記録作成）： ユーザが気付いていない暗黙の要求までを抽出する必要があるため、コンテクスチュアル・インクワリ手法⁽¹²⁾⁽¹³⁾を用いて開発者がユーザへ質問を行い、ユーザの回答の文脈からユーザ体験を把握する。限定ユーザが意識するのは操作であることから、質問を実施するにあたり、後から操作を思い出すことができるように、限定ユーザに時間・操作が明確になる項目で構成した、プロトタイプ使用時の操作記録の作成を依頼する。操作記録の項目を表 5.4 に示す。

2. 評価結果（インタビュー記録作成）：まず「何を教えて欲しいか」テーマをユーザに伝える。このため、表 5.4 の操作記録から、繰り返し操作を行っている、処置結果が NG となっている、作業周期と合わない時期に実施しているなどの特異点と思われる作業項目を見つけ、日付をもとにユーザの行動を理解するために「何をしたのか」「その時のことを教えて下さい」とインタビューを開始し、ユーザ体験を質問形式で教えてもらう。インタビューで「なぜ」その行動をしたのかという質問は行わず、「この時のことを教えて下さい」「どの程度の頻度で、あるいはどの程度の時間行っているか」「その前は何をしたか」「そのあとは何をしたか」など「くわしく」質問することで、ユーザから具体的な経験談を聴き出す。開発者がユーザの体験を把握することが目的である。約 30 分程度のインタビューでユーザが発言した言葉（発話）を評価結果として全て記録する。

3. ビジブル化（ユーザストーリー票を作成）：ユーザのプロフィール、テーマに対するエピソードとなる行動、象徴的な発言や関係する写真などユーザの特徴を表す項目をインタ

表 5.4 操作記録項目

操作記録項目
(1)操作項目名称
(2)操作のサイクル
(3)操作内容
(4)管理基準
(5)処置内容
(6)処置結果
(7)作業実施日と時刻



ビュー記録から抽出する。次にこうなったら良いというユーザ要求として、1件1葉に誰が何の操作をするという形のユーザストーリー票を作成する。ユーザストーリー票を、時系列化やグループ化など並べ替え操作を繰り返し、利用目的を抽出する。この抽出された利用目的に、実現する開発技術を対応させる。図 5.3 ユーザストーリー票から利用目的を抽出し実現方式を対応させる操作例を示す。この結果ユーザ要求と実現方式が明確になり、確定製品仕様を得る。

5.3 実装

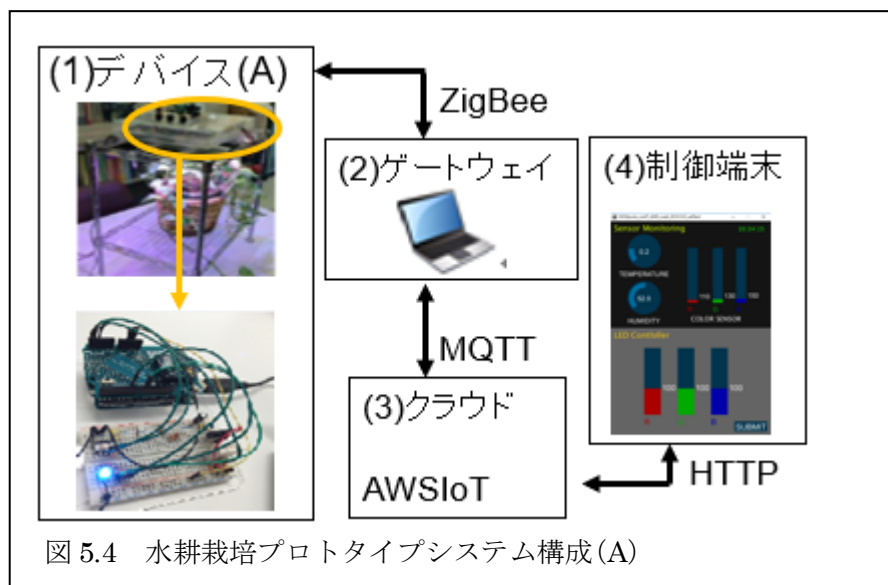
本検証方式を評価するための実装は、水耕栽培と害鳥駆除(ドローン)について実施した。前者はデバイスが移動しないものであり、後者はデバイスが移動するものである。

5.3.1 水耕栽培プロトタイプ

5.3.1.1 水耕栽培プロトタイプの機能

水耕栽培システムは、レタスなどすでに大規模なシステムが製品化されているが、葉物以外の植物についても、青・赤の光の照度と照射時間、温湿度と培養液をコントロールして生育期間を短縮し安定的な栽培を行うための育成データを収集する目的の IoT システムが報告されている⁽¹⁴⁾。筆者らは様々な植物の生育データを蓄積する目的をもった実用システムの開発を目的として、デバイス・ゲートウェイおよび商用クラウドである AWSIoT(Amazon Web Service IoT) を用いてプロトタイプを開発した。

今回開発した水耕栽培プロトタイプのシステム構成を図 5.4 に示す。(1)デバイスは温湿度センサー、照度センサー、およびカラーLED というセンサーとアクチュエータ群で水耕栽培装置を構成する。培養液の灌水は人手で実施する。センサーで検知された温湿度は、(2)ゲートウェイ経由で(3)クラウドへ送信される。クラウド上のアプリケーションを用いて操作員が持つ(4)制御端末へデバイスの植物育成データとして表示される。操作員は育成計画に従って(4)制御端末から LED の照度の変更指示を行う。LED 照度の指示データは(3)クラウドへ送信され、アプリケーションを用いて(2)ゲートウェイ経由で(1)デバイスへカラーLED の照度の値を送信する。温度管理は外部に設置されたクーラで制御する。水耕栽培プロト



タイプの仕様を製品仕様と対比させて、表 5.5 に示す。表中(A)は製品の要件定義仕様、(B)はプロトタイプで実現している機能を示す。

表 5.5 製品とプロトタイプの仕様比較（水耕栽培 A）

仕様		(A)製品	(B)プロトタイプ
機能	光源	○	○
	送風	○	—
	CO2	○	—
	養液温度	○	—
	養液pH	○	—
	養液EC	○	—
表示	センサーデータ表示	○	○
	カメラ映像表示	○	—
	育成記録	○	—
	保守	○	—

— : 未実装

○ : 実装

5.3.1.2 検証結果に基づく製品完成までの距離の算出

仕様達成度評価テストと拡張容易性評価，およびハザード対策テストケースを実施した結果を，合格件数/試験項目件数として表 5.6 に示す。テストはシステム全体を対象に実施するが，評価は構成要素毎に示しハザード対策はシステム全体で表す。

テスト結果の数値評価を行うために，表 5.6 のテスト結果を，表 5.7(1)に示すテスト結果の欄に記入し，(2)に評価水準に従った評価結果を記入する。水耕栽培の場合の評価水準を，評価者の経験値から合格率が 95%以上のものを良好 Good，75%以上を普通 Average，それ以下を不足 Ugly と設定した。(1)の機能達成度の試験合格率は $55/60=0.917$ であるので，(2)の評価結果は普通と判定される。(2)の評価結果に重み付けを行い「良い」の値で割り正規化した結果を(3)に示す。この値に(4)評価基準の重みづけを掛けた結果を(5)評価点に示す。(5)の各項目の合計である総合評価の値は 0.370 であることから，製品完成までの距離は 37%という評価値を得た。この結果から、プロトタイプを継続開発して製品化することは困難と判断した。

表 5.6 テスト結果（水耕栽培 デバイス A）

テスト対象	機能達成度	機能拡張性	ハザード対策
デバイス	10/15	1/5	---
ゲートウェイ	20/30	5/10	---
クラウド	20/20	10/10	---
制御端末	5/5	5/5	---
全体システム	55/60	11/30	7/9

表 5.7 総合評価結果（水耕栽培 デバイス A）

評価基準	機能達成度	拡張容易性	ハザード対策	総合評価
(1)テスト結果 (合格/テスト件数)	55/60 =0.917	11/30 =0.367	7/9 =0.778	—
(2)評価結果	普通	不十分	普通	—
(3)絶対評価と 正規化	0.280/ 0.627	0.094/ 0.627	0.280/ 0.627	—
(4)重みづけ	0.637	0.258	0.105	—
(5)評価点	0.284	0.039	0.047	0.370

5.3.1.3 水耕栽培プロトタイプに対する第三者的検証による製品仕様の確定

プロトタイプの第三者的検証結果から、製品化までの距離は 0.37 という数値を得た。この結果、プロトタイプを継続開発して製品化することは困難と判断し、デバイスのプロトタイプを作り直しデバイス (B)とした。このデバイス (B)を図 5.5 に示す。水耕栽培プロトタイプのデバイス (B)の仕様を製品仕様と対比させて、表 5.8 に示す。表中(A)は製品の要件定義仕様, (B)はプロトタイプデバイス (B)で実現している機能を示す。

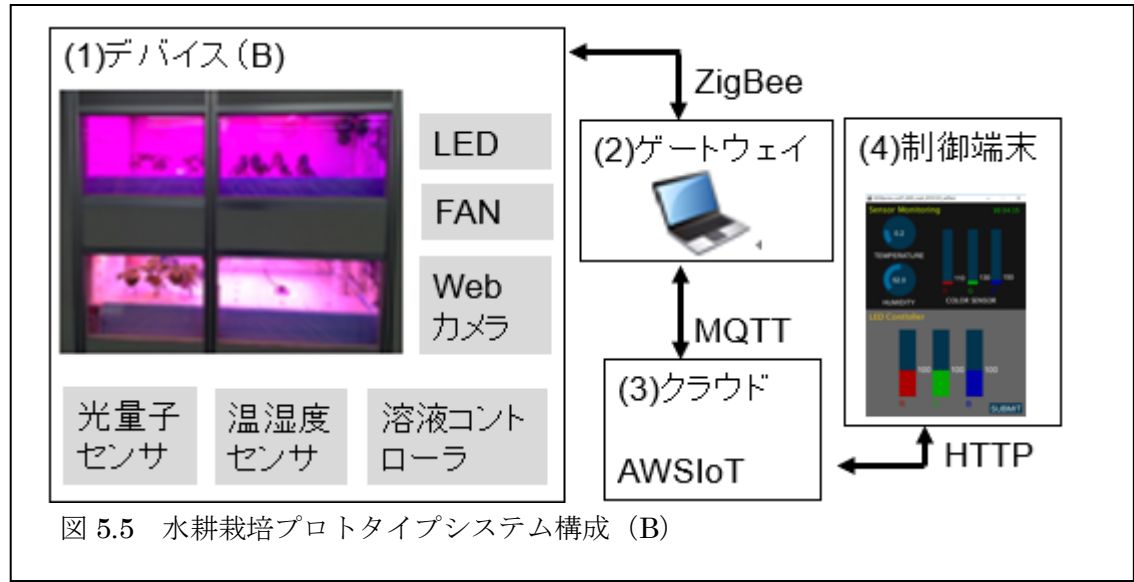


図 5.5 水耕栽培プロトタイプシステム構成 (B)

表 5.8 製品とプロトタイプの仕様比較 (水耕栽培 B)

仕様		(A)製品	(B)プロトタイプ
機能	光源	○	○
	送風	○	—
	CO2	○	○
	養液温度	○	○
	養液pH	○	○
	養液EC	○	○
	養液EC	○	○
表示	センサーデータ表示	○	○
	カメラ映像表示	○	○
	育成記録	○	—
	保守	○	—

— : 未実装

○ : 実装

表 5.9 プロトタイプ of 操作記録 (水耕栽培 デバイス (B))

操作内容	操作方法	操作サイクル	操作目的
(1)温度の監視	Web画面	毎日	室内温度を確認
(2)培養液の水量監視	目視	毎日	水位を調整
(3)葉の摘花時期の把握	目視	適宜実施	4枚葉を摘花
(4)培養液の入れ替え	目視	適宜実施	水位を調整

表 5.10 ユーザ評価結果から得られた利用目的と開発機能

ユーザ評価結果から得られた利用目的と実現方式 (デバイス (B))		
ユーザ操作のエピソード	利用目的	開発機能
(1)水位の誤認識	適正な量の溶液の投入	自動灌水
(2)何度も室内温度を確認	均一な室内温度設定	室内の送風
(3)成長記録を覚えていない	適切な摘花時期の把握	Webカメラの設置とデータ保続

デバイス B のタイプのプロトタイプを対象に限定ユーザ評価を実施する。

限定ユーザが毎日記録した水耕栽培装置の操作記録表を表 5.9 に示す。この表をもとに、不定期であるが回数の多い項目、目視で実施した項目、手動操作の項目について、実施日を指定して、何を行ったかを最初の質問項目として選択し、インタビューを実施した。

この結果、表 5.10 に示す次の 3 つの利用目的とそれに対応した実現方式を明確にした。

(1)水位自動調整機能をもった自動灌水機能により、ユーザの水位観測誤りを防止できる。

(2) 送風 FAN により室温を均一にすることができる。(3)Web カメラのデータを日単位で保存することで、過去の生育状態の確認ができる。

5.3.2 害鳥駆除 UAV プロトタイプ

5.3.2.1 害鳥駆除 UAV プロトタイプの機能

上空から飛来するカラスやムクドリなどによる果実の被害は大きく、対策として花火や防護ネットの設置などを行ってもすぐに慣れてしまい、効果が少ない。そこでこれら害鳥が恐れる鷹や鷲などの天敵を、UAV(無人飛行機：ドローン)で模擬するシステムが報告されている⁽¹⁵⁾。筆者らは特定の果樹園に対応して、害鳥駆除への有効性を評価する目的でプロ

表 5.12 テスト結果（害鳥駆除 UAV(ドローン)）

テスト対象	機能達成度	機能拡張性	ハザード対策
デバイス	15/15	4/5	---
ゲートウェイ	20/20	9/10	---
クラウド	20/20	9/10	---
制御端末	5/5	5/5	---
全体システム	60/60	27/30	4/5

表 5.13 総合評価結果（害鳥駆除 UAV(ドローン)）

評価基準	機能達成度	拡張容易性	ハザード対策	総合評価
(1)テスト結果 (合格/テスト件数)	60/60 =1.000	27/30 =0.900	4/5 =0.800	—
(2)評価結果	良好	普通	普通	—
(3)絶対評価と 正規化	0.627/ 0.627	0.280/ 0.627	0.280/ 0.627	—
(4)重みづけ	0.637	0.258	0.105	—
(5)評価点	0.637	0.115	0.047	0.799

5.3.2.2 検証結果に基づく製品完成までの距離の算出

仕様達成度評価テストと拡張容易性、およびハザード対策テストケースを実施した結果を、合格件数/試験項目件数として表 5.12 に示す。

テストはシステム全体を対象に実施するが、評価は構成要素毎に示しハザード対策はシステム全体で表す。

テスト結果の数値評価を行うために、表 5.12 のテスト結果を、表 5.13 (1)に示すテスト結果の欄に記入し、(2)に評価水準に従った評価結果を記入する。

UAV の場合の評価水準を、評価者の経験値から合格率が 95%以上のものを良好 Good, 80%以上を普通 Average, それ以下を不足 Ugly と設定した。(1)の仕様達成度の試験合格率は 60/60=1.000 であるので、(2)の評価結果は良いと判定される。(2)の評価結果に重み付けを行い「良い」の値で割り正規化した結果を(3)に示す。この値に(4)評価基準の重みづけを掛けた結果を(5)評価点に示す。(5)の各項目の合計である総合評価の値は 0.798 であることから、製品完成までの距離は 80%という値を得た。

表 5.14 プロトタイプ の操作記録（水耕栽培 デバイス（B））

操作内容	操作方法	操作サイクル	操作目的
(1)カラス検知	スマホへ通知がある	適宜実施	UAV(ドローン)発信
(2)カラス検知	目視	適宜実施	スマホ通知時にカラスの侵入を目で確認する。
(3)UAVを充電器に接続	手動	適宜実施	UAVのバッテリー充電
(4)飛行ルートを設定する。	手動	適宜実施	UAVの飛行データをUAVへ設定する。

表 5.15 ユーザ評価結果から得られた利用目的と開発機能

ユーザ 評価結果から得られた利用目的と実現方式		
ユーザ 操作のエピソード	利用目的	実現機能
(1)風によるカラスの誤認識	カラスの認識率の向上	カメラの設置台数増加
(2)逃げたカラスが戻ってくる	繰り返し飛行	GPS搭載による自動充電機能実現
(3)逃げたカラスの方向へドローンを飛ばす	カラスの追跡	Webカメラの搭載
(4)梅雨時のカラスの出現	雨の中の飛行	なし

5.3.2.3 害鳥駆除 UAV プロトタイプに対する第三者的検証による製品仕様の確定

限定ユーザが UAV の操作実施時に記録した害鳥駆除 UAV の操作記録表を表 5.14 に示す。

この表をもとに、操作回数の多い項目、目視項目、手動操作、害鳥駆除の結果について、実施日を指定して、何を行ったかを最初の質問項目として選択しインタビューを実施した。この結果、表 5.15 に示す次の 4 つの利用目的とそれに対応した実現方式を明確にした。

(1)カラスの認識率に合わせて監視カメラを増設することで、カラスの誤認識を減少させることができる。(2) GPS による着地地点精度向上と自動充電を行い繰り返し飛行することで、戻ってきたカラスに対してドローンを飛行させることができる。(3)Web カメラをドローンに搭載させることで、カラスの追跡機能の実現データを収集する。(4)雨天時での飛行には防水機能が必要となるが、現状は実現できない。

5.4. 評価と考察

5.4.1 評価

5.4.1.1 製品完成までの距離の算出

(4) 製品完成までの距離 検証結果の数値化に用いた AHP は複数の候補(代替案) に対して優先付けを行う手法であるが、今回は評価基準と評価水準の重みづけを IoT 製品の検証という観点から行い、検証対象のプロトタイプと製品の 2 つを挙げ、AHP 絶対評価を行った。<5.3.1>水耕栽培プロトタイプ(A)では 37%の距離、<5.3.2>害鳥駆除 UAV では 80%の距離という数値が得られた。害鳥駆除 UAV プロトタイプはこのまま製品開発の継続が可能という検証結果となり、水耕栽培プロトタイプ(A)ではこのまま製品版への開発を継続することは困難という検証結果を得た。

今回は、各評価基準に対する評価水準を同一としたことから、AHP 評価結果の数値として得た距離は、評価水準として設定した 3 つの重みベクトルの数値の組み合わせの数値となる。

5.4.1.2 製品仕様の確定

プロトタイプの限定ユーザによる評価結果から得られたエピソードをもとに、開発者がユーザ要求をビジブルし、利用目的と実現方式を明確にして開発仕様を確定する手法を、水耕栽培プロトタイプと害鳥駆除 UAV プロトタイプに適用し、製品として実現した。当初の開発仕様と、ユーザ評価から得た利用目的と対応する実現方式をもとに実現した製品仕様との比較結果を示す。

1. 水耕栽培プロトタイプ(B)

- (1) 自動灌水については、ユーザ操作性向上のための水位調整機能がある。
- (2) 送風については、発生した O₂ が葉を覆い CO₂ が吸収できなくなるのを防ぐ目的の他に、室温を均一にする目的を実現している。
- (3) Web 表示として植物の画像が必要であることは要件定義と同様であるが、ユーザ要求である当日データだけでなく、過去の画像データ表示機能も実現している。

2. 害鳥駆除 UAV プロトタイプ

- (1) 監視カメラの 10 台制御について、カラスの認識率に合わせてカメラを増設可能とする機能を実現している。
- (2) 飛行時間について要件定義の仕様は 30 分であるが、カラスを駆除する目的から、飛行時間延長よりも、GPS による着地地点精度向上と自動充電を行い、繰り返し飛行可能としている。
- (3) 当初仕様に定義されていないが、カラスの駆除効果を上げるため、カラス追跡機能の追加を開発者が発案した。このための開発データ収集のため、Web カメラのドローンへの搭載を製品として実現した。

(4) カラスの出現が梅雨時から始まるため、雨天時での飛行が求められた。実現には防水機能が必要となるが、当初要求仕様の前提から離れるため、今回は実施を行わないこととした。

上記のように、当初の開発仕様と比較して、利用目的と対応する開発方式が明確になったことから、本手法が製品仕様の明確化に有効であると判断する。

5.4.2 考察

本論文で提案したプロトタイプ限定ユーザによる評価手法について考察する。

1. 限定ユーザと実ユーザの違いを考慮する必要がある。ユーザ評価の実施人数については、5人で全ユーザの85%の要求を検出できるというヤコブ・ニールセンの説がある⁽¹²⁾。しかしIoT分野は応用範囲が広いと、システムを利用するユーザ層を絞れない場合は、ユーザ層毎にユーザ評価の実施が必要であると言える。

2. コンテクスチュアル・インクワリー手法（質問形式）の実施方法。本手法はユーザ体験を語り出すきっかけを作るために質問を行う手法であることから、予め質問票は作成しない。しかし何も準備しないと、ユーザによっては何も語らないという場合も予想されるため、ユーザの操作記録など操作を思い出すツールの準備や、あるいはその場で操作してもらいながら質問するなど、プロトタイプシステムに合わせたヒヤリングの準備が必要である。

3. プロトタイプを対象とした限定ユーザによる評価実施のタイミングは、対象となるIoTシステムにより異なる。農業分野や家庭を対象としたIoTシステムでは、機能が全ては実装されていない開発の早い段階で限定ユーザに絞ってプロトタイプをリリースする可能性はあるが、製造業や自動車などミッションクリティカルに近い分野では、ほぼ完成した段階でテストドライバーなど専門の評価者による評価が必要と考える。

5.5. まとめ

本研究では、プロトタイプを客観的に評価した結果に基づいて、プロトタイプから製品完成までの開発の難易度を数値化して距離として表し、さらにユーザ要件定義と対応する開発項目を確定することで、プロトタイプから製品へ方向付けを行う方式の提案を目的とした。その結果、図5.7に示す様に、「プロトタイプから製品完成までの距離の算出方式」および「ユーザ要求仕様の抽出方式」を提案し、水耕栽培プロトタイプおよび、害鳥駆除UAV（ドローン）プロトタイプへ適用し、本手法が有効であるという成果を得た。

本章では、プロトタイプ of 第三者的検証を用いたプロトタイプから製品までの開発の方向付け方式を提案し、実装評価を行った。

成果 : ・プロトタイプをそのまま継続開発して製品化する場合の、開発の難易度を数値化した。
・プロトタイプの時点で限定ユーザによる第三者評価を行いユーザ要求の抽出と対応する開発項目の明確化した。

成果物: ・「プロトタイプから製品完成までの距離の算出方式」。
・「ユーザ要求仕様の抽出方式」

図 5.7 研究の成果

6. 結言

本研究では、IoT システムの応用分野が広がり一般消費者との係りが深くなることから、民生品 IoT システム製品の信頼性・強靱性・安全性向上のための第三者検証に着目し、システム製品とプロトタイプの第三者検証実施の課題の解決のため、以下の 3 項目を研究の目的とした。

- (1) 民生品 IoT システム製品の不具合推定を含んだ第三者検証方式
- (2) IoT プロトタイプの開発者による第三者的検証方式
- (3) IoT システムのプロトタイプの第三者的検証による開発の方向付け

なお本論文においては、「第三者検証」および「第三者的検証」をつぎのように定義して使用する。「第三者検証」は、開発者とは異なる第三者が開発者とは異なる評価軸で、システム製品に対して実施する検証を表す。「第三者的検証」は、開発者が第三者検証を模して、プロトタイプに対して客観的に実施する検証を表す。

(1) の研究では、民生品の IoT システム製品に対する、第三者検証実施のための方式を提案した。この方式では、開発者が他の組織である検証部門へ IoT システムの検証を依頼する際に、IoT システムの構成をサブシステム単位に分割して機能とインタフェーストレースを記載、さらに予めインタフェーストレースとログを採取できるテスト環境を記載した「検証要求仕様書」を作成し、これに基づいて検証部門では「検証設計書」を作成し開発部門の「承認」後に検証を開始する。このように、検証が必要な時期に第三者検証部門を編成可能とした。特に検証部門で検出した障害について、サブシステム単位に障害の発生箇所と原因の推定を可能とすることで開発部門での障害原因解析を容易にすることができる。以上の「検出障害の推定機能をもつ第三者検証方式」および「検出障害の発生箇所と原因の推定方式」を、家庭電化製品の遠隔操作システムに適用し、その有効性を確認した。

(2) の研究では、IoT システム開発におけるプロトタイプに対する、開発者による第三者的検証方式を提案した。この方式では、プロトタイプに対して開発者が製品同様の機能・性能・規格・責任範囲とテスト環境を明記した「プロトタイプ機能仕様書」を作成し、さらに検証部門の立場で「検証設計書」を作成して検証を実施する。これにより、開発者によるプロトタイプの客観的な第三者的検証を可能とし、特に製品化までに行うべき事柄を明らかにすることができる。以上の「開発不足項目の指摘機能をもつプロトタイプの第三者的検証方式」を、植物の育成データの採取を目的とした水耕栽培システムと、害鳥駆除を目的としたドローンシステムのプロトタイプに適用し、その有効性を確認した。

(3) の研究では、IoT システム開発におけるプロトタイプの第三者的検証結果に基づく

開発の方向付けの方式を提案した。この方式では、プロトタイプの第三者的検証の試験の観点を評価基準とし、対象 IoT システムの特性に合わせて試験結果の評価水準を設定し、プロトタイプの第三者的検証結果から AHP を用いて製品に対してプロトタイプの絶対評価を行う。これにより、プロトタイプを継続して拡張開発を行い製品化するまでの難易度を数値化して、プロトタイプから製品完成までの距離として表すことができる。またプロトタイプを人数・性別・年齢を限定したユーザ評価による第三者評価を行うことで、ユーザ要求と対応する実現方式を明確化することができる。開発完了までの距離の把握および、要件定義と実現方式の明確化を合わせて、プロトタイプから製品化への開発の方向付けを行うことができる。以上の「プロトタイプから製品完成までの距離の算出方式」および「プロトタイプの限定ユーザ評価方式」を、植物の育成データの採取を目的とした水耕栽培システムと、害鳥駆除を目的としたドローンシステムのプロトタイプに適用し、その有効性を確認した。

図 6.1 に、研究の成果を示す。

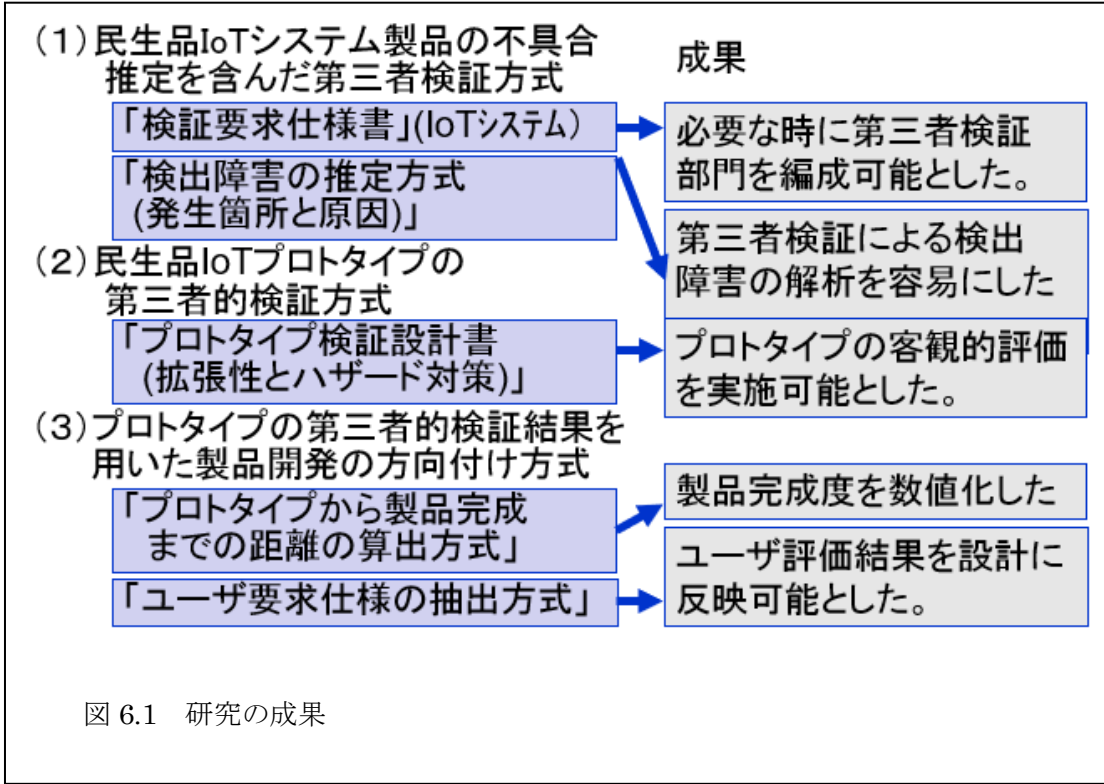


図 6.1 研究の成果

検証は、開発者とは異なる検証の観点を探し続ける必要があり、新たな技術を用いたシステム製品に対応した検証の実践の積み重ねが重要である。このため、本方式についても、家庭電化製品の遠隔操作・水耕栽培・害鳥駆除のドローンシステム以外の適用事例を増やし、評価項目・テスト手法・不具合原因タイプ・ユーザ操作記録などの実績を蓄積することが課題となる。

また本研究では、民生品 IoT システム製品に対する、開発者主導の第三者検証方式を提案したが、今後この方式を普及させ、第三者機関による認証制度として確立させるよう努力したい。

今後上記の課題解決へ向けた研究と組織的取り組みを行い、IoT システム製品の品質・生産性向上に貢献する所存である。

図 6.2 に今後の研究課題を示す。

新技術へ対応した検証手法の獲得と開発者と異なる評価軸による検証視点を探し続けるため、**実践経験**の積み重ねと**組織的な取り組み**を行う。

- 民生品IoTシステム製品に対する第三者検証の実施を可能とした。さらにプロトタイプ of 第三者的検証を実施可能として、開発の方向付けも可能とした。
今後、これらの第三者検証方式の普及に努め、IoTシステム製品の開發生産性の向上に貢献したい。
- 民生品IoTシステム製品に対する、開発者主導の第三者検証方式を提案したが、今後この方式を普及させ、第三者機関による認証制度として確立させるよう努力したい。

図 6.2 今後の研究課題

謝辞

本研究を学位論文として提出するにあたりまして、お忙しい中多くの時間を割いてご指導いただきました、東京電機大学大学院 先端科学技術研究科 神戸英利教授に、厚く御礼申し上げます。

東京電機大学 小泉寿男名誉教授には、学会発表、論文投稿、査読回答など、研究活動の様々な場面におきまして、ご指導および激励をいただきました。東海大学 情報通信学部 組込みソフトウェア工学科 大江信宏教授には、M2M・IoT研究会の活動におきまして様々な助言をいただきました。株式会社 VSN の中島幸一様には、ドローン検証に関して協力をいただきました。福井工業大学 経営情報学科 北上眞二教授には、開発者の立場から第三者検証のメリットについてアドバイスをいただきました。芝浦工業大学 システム理工学部 電子情報システム学科 井上雅裕教授には、検証結果の客観的評価手法について技術的なアドバイスをいただきました。サレジオ工業高等専門学校 機械電子工学科 米盛弘信准教授には、IoT プロトタイプ法の実践に関して協力をいただきました。皆さまには心から感謝を申し上げます。

所属元の株式会社メルコテクノ横浜 居駒哲夫社長には、学位取得に関してご理解をいただきました。深く感謝致します。

末筆ながら、論文作成をいつも暖かく励ましてくれた私の家族 朋子、由香、茉莉に、心から感謝します。

参考文献

(筆者関連発表論文)

【学術論文リスト】

- (1) 堂坂 辰・大江 信宏・北上 眞二・神戸 英利・小泉 寿男：「不具合の推定を含んだシステム製品における第三者検証とその実装評価」，電気学会論文誌 C（電子・情報システム部門誌），Vol.136，No.6 pp.868-880 (2016)
- (2) 堂坂 辰・大江 信弘・中島 幸一・米盛 弘信・井上 雅裕・神戸 英利・小泉 寿男：「IoT プロトタイプシステムの第三者視点からの検証方式とその実装評価」，電気学会論文誌 C（電子・情報システム部門誌），Vol.138，No.6 pp.743-754 (2018)
- (3) Shin Dosaka: "Prototype for product validation process of IoT system", International Journal of Internet of Things 2018, 7(2):23-29, Scientific and Academic Publishing
<http://article.sapub.org/10.5923.j.ijit.20180702.01.html> (reference 2018/8)

【国際学会発表論文】

- (1) Shin Dosaka, Nobuhiro Ohe, Masahiro Inoue, Hidetoshi Kambe: "Distance Measurement Between a Prototype and Products in an Internet Things System Using Analytic Hierarchy Process", The Sixth IEEE Global Conference on Consumer Electronics (GCCE 2017), pp. 124-128 (2017)

1 章分

- (1) 「IV&V ガイドブック」, JAXA 出版 ISSN1349-113X JAXA-SP-12-016 (2013.3)
- (2) 「M2M/IoT システム入門」，電気学会 第2次 M2M 技術調査専門委員会，2016 年

2 章分

- (1) ISO24765 : Systems and software engineering
<https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:24765:ed-1:v1:en> (参照 2015.10)
- (2) USB Official site : <https://www.usb.org/usb-press-room> (参照 2018.9)
- (3) Bluetooth Official site : <https://www.bluetooth.com/ja-jp> (参照 2018.11)
- (4) ISO15408 : 国内外の政府調圧のためのセキュリティ要件の認証制度,
<https://www.ipa.go.jp/security/jisec/index.html> (参照 2018.9)
- (5) 「ECHONET Lite 規格書」, エコーネットコンソーシアム公式サイト :
<http://www.echonet.gr.jp/spec/index.htm> (参照 2015.10)
- (6) 「組込みシステムの第三者検証に関する調査」, 調査報告書 IPA, 独立行政法人情報処理推進機構(2012), <https://www.ipa.go.jp/sec/softwareengineering/reports/20120315.html>
(参照 2018.8)
- (7) S. Matsumoto, H. Uwano, A. Monden, K. Matsunoto, M. Katahira, N. Kohtake, Y. Miyamoto, S. Ujihara, S. Yoshikawa : “Quantitatively Characterizing Software IV&V Activities Based on Defect History”, D, Vol.J92-D, No.12, pp.2159-2206(2009) (in Japanese)
 松本真佑 : 「不具合履歴に基づくソフトウェア IV&V 活動の定量的見える化手法」,
 電子情報通信学会論文誌, D, Vol.J92-D, No.12, Pp.2159-2206, 電子情報通信学会(2009)
- (8) Kataihira Masafumi , Okuda Kazumi: “Strategy to enhance IV&V in JAXA”,
 International Workshop on Strategic and Economic Method for Assessment of
 IV&V Activities, pp.19-21(2005)
 片平真史, 奥田一美 : 「JAXA における IV&V 活動の強化戦略」, 第 38 回 HISS に関
 連した IV&V ワークショップ, pp.19-21(2005)
- (9) 「IV&V ガイドブック」, JAXA 出版 ISSN1349-113X JAXA-SP-12-016 (2013.3),
<https://repository.exst.jaxa.jp/dspace/handle/a-is/19895> (参照 2018.8)
- (10) S. Miyahara, T. Ibusuki, H. Tanaka : “TMP for Assuring Quality of Hardware Productgs ”,
 FUJITSU.58, 4, pp.365-369(2007)(in Japanese)
 宮原真次・指宿剛・田中旬 : 「ハードウェア製品の品質を支える試験・保守プログ
 ラム : TMP」, FUJITSU.58,4,pp.365-369(2007)
- (11) A. Suzuki, T. Suzuki, T. Morinaga : “Approach to Process and Product Quality Improvement
 by Mean of Verification and Testing Conducted by Third Parties”, FUJITSU.62, 2,
 pp.248-253(2011) (in Japanese)
 鈴木在久・鈴木哲二郎・森永実 : 「第三者検証とテスト支援によるプロセス・製品
 品質向上の取組み」, FUJITSU.62,2,pp.248-253(2011)
- (12) 「つながる世界の品質確保に向けた手引き」、IPA、独立行政法人情報処理推進機構
 (2018)、<https://www.ipa.go.jp/files/000064877.pdf> (参照 2018.8.)
- (13) 電気学会 第 2 次 M2M 技術調査専門委員会編 : 「M2M/IoT システム入門」, 森北出
 版株式会社 (2016)

- (14) N. Ohe, S. Kitagami, H Yonemori, M Inoue, T Shiotsuki, H. Koizumi : "A Proposal of Education System for Manufacturing Technology using M2M Prototyping and its Practice", IEEJ Trans. EIS, Vol.135, No.11, pp. 655-665(2015)(in Japanese)
大江 信宏・北上 眞二・米盛 弘信・井上 雅裕・汐月 哲夫・小泉 寿男:「M2Mのプロトタイプ構築によるものづくり教育システムの提案と実践」, 電学論 C, Vol.135, No.11 pp.655-665 (2015)

3 章分

- (1) ISO24765 : Systems and software engineering
<https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:24765:ed-1:v1:en> (参照 2015.10.12)
- (2) 「組込みシステムの第三者検証に関する調査」,調査報告書 IPA, 独立行政法人情報処理推進機構(2012)
- (3) S. Miyahara, T. Ibusuki, H. Tanaka : “TMP for Assuring Quality of Hardware Products ”, FUJITSU.58, 4, pp.365-369(2007)(in Japanese)
宮原真次・指宿剛・田中旬 : 「ハードウェア製品の品質を支える試験・保守プログラム : TMP」, FUJITSU.58,4,pp.365-369(2007)
- (4) A. Suzuki, T. Suzuki, T. Morinaga : “Approach to Process and Product Quality Improvement by Mean of Verification and Testing Conducted by Third Parties”, FUJITSU.62, 2, pp.248-253(2011) (in Japanese)
鈴木在久・鈴木哲二郎・森永実 : 「第三者検証とテスト支援によるプロセス・製品品質向上の取組み」, FUJITSU.62,2,pp.248-253(2011)
- (5) S. Matsumoto, H. Uwano, A. Monden, K. Matsunoto, M. Katahira, N. Kohtake, Y. Miyamoto, S. Ujihara, S. Yoshikawa : “Quantitatively Characterizing Software IV&V Activities Based on Defect History”, D, Vol.J92-D, No.12, pp.2159-2206(2009) (in Japanese)
根本真佑 : 「不具合履歴に基づくソフトウェア IV&V 活動の定量的見える化手法」, 電子情報通信学会論文誌, D, Vol.J92-D, No.12, Pp.2159-2206, 電子情報通信学会(2009)
- (6) 「STAMP 手法に関する調査報告書」,調査報告書 IPA, 独立行政法人情報処理推進機構(2015), <https://www.ipa.go.jp/files/000047911.pdf> (参照 2016.2.7)
- (7) 「IV&V ガイドブック」,JAXA 出版 ISSN1349-113X JAXA-SP-12-016 (2013.3)
- (8) 「システム記述言語 (AADL) による複合システム設計」,先進的な設計・検証技術の適用事例報告書, IPA, 独立行政法人情報処理推進機構(2015),
<https://www.ipa.go.jp/files/000049374.pdf> (参照 2016.2.7)
- (9) 稲田修一 : 「M2M/IoT 教科書」, インプレス社(2015)
- (10) H. Tsuji, J. Sawamoto, K. Seo, K. Kitagami : “M2M (Machine-to-Machine) Technology Trend”, IEEJ Transaction on Electronics, Information and Systems, Vol.133(3), pp.520-531(2013)(in Japanese)
辻秀一・澤本潤・瀬尾克彦・北上真二 : 「M2M(Machine-to-Machine)技術の動向」, 電気学会論文誌 C (電子・情報・システム部門誌) , 133(3), pp520-531(2013)
- (11) N. Ohe, S. Kitagami, H. Yonemori, M. Inoue, T. Shiotsuki, H. Koizumi : “A proposal of an Education System for Manufacturing Technology using M2M Prototyping and its Practice” , IEEJ Transaction on Fundamentals and Materials Vol.135 No.11 pp.655-665 (in Japanese)
大江信宏・北上真二・米森宏信・井上雅裕・汐月哲夫・小泉寿男 : 「M2M プロトタイプ構築によるものづくり教育の提案と実践」, 電気学会論文誌 A (基礎・材料・共通部門誌) , 2015 年 11 月号 Vol.135, No.11 pp.655-665 (2015)

- (12) ECHONET Lite 規格書, エコーネットコンソーシアム公式サイト :
<http://www.echonet.gr.jp/spec/index.htm> (参照 2015.10.12)
- (13) 「えねっとくん」 : <http://www.xxcal.co.jp/testing/echonet-lite/about.html> (参照 2015.10.12)
- (14) Wireshark, Wireshark, Official site : <https://www.wireshark.org/> (参照 2015.10.12)
- (15) JMeter, Apache JMeter Official site : <http://jmeter.apache.org/> (参照 2015.10.12)
- (16) JBento, Sourceforge Official site : <http://jbento.sourceforge.net/ja/> (参照 2015.10.12)
- (17) 「システム及びソフトウェア品質の見える化,確保及び向上のためのガイド」,ソフトウェアメトリックス高度化プロジェクト プロダクト品質メトリックス WG,経済産業省(2010)
http://www.meti.go.jp/policy/it_policy/softseibi/metrics/product_metrics.pdf (参照 2016.2.7)
- (18) 「情報処理システム高信頼化教訓ハンドブック(IT サービス編)」,IPA, 独立行政法人情報処理推進機構(2014)

4 章分

- (1) 「組込みシステムの第三者検証に関する調査」,調査報告書 IPA, 独立行政法人情報処理推進機構(2012)
<http://www.ipa.go.jp/files/000004607.pdf> (参照 2017.03.12)
- (2) 「IV&V ガイドブック」,JAXA 出版 ISSN1349-113X JAXA-SP-12-016 (2013.3)
- (3) S. Dosaka, N. Ohe, S. Kitagami, H. Kambe, H. Koizumi : "A Method of Third Party Verification for System Products Includeing the Presumpiton of Fault Location and Cause, and its Evaluation on the Implementation", IEEJ Trans. EIS, Vol.136, No.6, pp.868-880(2016)(in Japanese)
堂坂 辰・大江 信宏・北上 眞二・神戸 英利・小泉 寿男:「不具合の推定を含んだシステム製品におえる第三者検証とその実装評価」, 電学論 C, Vol.136, No.6 pp.868-880 (2016)
- (4) 電気学会 第2次 M2M 技術調査専門委員会編:「M2M/IoT システム入門」, 森北出版株式会社 (2016)
- (5) N. Ohe, S. Kitagami, H Yonemori, M Inoue, T Shiotsuki, H. Koizumi : "A Proposal of Education System for Manufacturing Technology using M2M Prototyping and its Practice", IEEJ Trans. EIS, Vol.135, No.11, pp. 655-665(2015)(in Japanese)
大江 信宏・北上 眞二・米盛 弘信・井上 雅裕・汐月 哲夫・小泉 寿男:「M2M のプロトタイプ構築によるものづくり教育システムの提案と実践」, 電学論 C, Vol.135, No.11 pp.655-665 (2015)
- (6) ISO24765 : Systems and software engineering
<https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:24765:ed-1:v1:en> (参照 2017.5.6)
- (7) 「IoT 開発におけるセキュリティ設計手引き」, 独立行政法人情報処理推進機構 (2016) : <https://www.ipa.go.jp/files/000052459.pdf> (参照 2016.8)
- (8) Nancy G. Level on: "Engineering a Safer World", MIT Press,(2011)
<http://sunnyday.mit.edu/safer-world.pdf> (参照 2017.2)
- (9) 「STAMP 手法に関する調査報告書」, 独立行政法人情報処理推進機構 (2016) : <http://www.ipa.go.jp/files/000047911.pdf> (参照 2016.8)
- (10) 「はじめての STAMP/STPA」, 独立行政法人情報処理推進機構 (2016) : <http://www.ipa.go.jp/files/000051829.pdf> (参照 2016.8)
- (11) 「大規模・複雑化した組込みシステムのための障害診断手法」, 独立行政法人情報処理推進機構 (2016) : <http://www.ipa.go.jp/files/000051768.pdf> (参照 2016.8)
- (12) T. Iijima, A. Nagao, H. Yonemori:"A Case on the Hydroponic Culture by Different Field Technology", Japanese Society for Engineering Education, NII-Electronic Library Service. PP. 364-365(2014)(in Japanese)
飯島卓弥, 長尾明美, 米盛弘信:「異分野技術の融合による水耕栽培の実現」, 日本工学教育協会平成 26 年度工学教育研究講演会講演論文集, P364-365(2014)
- (13) AWS Official site : <https://aws.amazon.com/jp/> (参照 2017.5)
- (14) Rduino, Rduino Official site : <https://www.arduino.cc> (参照 2017.2)

- (15) Zigbee, Zigbee Alliance : <http://www.zigbee.org/> (参照 2017.2)
- (16) Processing, Processing Official site : <http://www.zigbee.org/> (参照 2017.2)
- (17) K. Nakajima, K Seo, S. Kitagami, T. Shiotsuki, H. Koizumi: “Development of Unmanned Aerial Vehicle Systems for Monitoring and Prevention of Bird and Animal Damage and its evaluation”, IPSJ SIG Technical Report (2015)(in Japanese)
中島幸一・清尾克彦・北上眞二・汐月 哲夫・小泉 寿男：「鳥獣被害対策用監視・防除 UAV システムの開発と評価」, 情報処理学会 電子図書館, IPSJ-CDS15014010.pdf (2015)
- (18) Nmap : <https://nmap.org/man/jp/> (参照 2017.12)

5 章分

- (1) 「システム記述言語 (AADL) による複合システム設計」, 先進的な設計・検証技術の適用事例報告書, IPA, 独立行政法人情報処理推進機構(2015),
<https://www.ipa.go.jp/files/000049374.pdf> (reference 2016.2.7) (in Japanese)
- (2) 「STAMP 手法に関する調査報告書」, 独立行政法人情報処理推進機構 (2016) :
<http://www.ipa.go.jp/files/000047911.pdf> (参照 2016.8)
- (3) Nancy G. Leveson, “Engineering a Safer World System Thinking Applied to Safety,” The MIT Press, 2011
https://static1.squarespace.com/static/53b78765e4b0949940758017/t/57d87eb6d2b8571af3501b26/1473898764674/Engineering_a_Safer_World+Nancy+Leveson.pdf (reference 2018.8)
- (4) 「IV&V ガイドブック」, JAXA 出版 ISSN1349-113X JAXA-SP-12-016 (2013.3) (in Japanese)]
- (5) Hiroko Umeda, Tsutomu Matsumoto, “Introduction of JAXA’s IV&V manual,” 2011 Annual Workshop on Validation and Verification, 2011
https://www.nasa.gov/sites/default/files/585640main_JAXAIVVmanual.pdf (reference 2018.8)
- (6) S. Dosaka, N. Ohe, S. Kitagami, H. Kambe, H. Koizumi : ”A Method of Third Party Verification for System Products Including the Presumption of Fault Location and Cause, and its Evaluation on the Implementation”, IEEJ Trans. EIS, Vol.136, No.6, pp.868-880(2016)(in Japanese)
堂坂 辰・大江 信宏・北上 眞二・神戸 英利・小泉 寿男: 「不具合の推定を含んだシステム製品におえる第三者検証とその実装評価」, 電学論 C, Vol.136, No.6 pp.868-880 (2016)
- (7) Shin Dosaka, Nobuhiro Ohe, Masahiro Inoue, Hidetoshi Kambe, “Distance Measurement Between a Prototype and Products in an Internet Things System Using Analytic Hierarchy Process,” The Sixth IEEE Global Conference on Consumer Electronics (GCCE 2017), pp. 124–128, 2017.
- (8) Shin Dosaka, Nobuhiro Ohe, Shinji Kitagami, Koichi Nakajima, Hidetoshi Yonemori, Masahiro Inoue, Hidetoshi Kambe, Hisao Koizumi, “A Method of Verification from the Third Party view for IoT Prototype System and its Implementational Evaluation,” IEEJ Trans. EIS, 138(6), 743-754, 2018 (in Japanese)
堂坂 辰・大江 信弘・中島 幸一・米盛 弘信・井上 雅裕・神戸 英利・小泉 寿男: 「IoT プロトタイプシステムの第三者視点からの検証方式とその実装評価」, 電学論 C, Vol.138, No.6 pp.743-754 (2018)
- (9) 木下栄蔵: 「入門 AHP」, 日科技連 (2000)
- (10) H. Ogasawara: “Analytic Hierarchy Process as a Research Method for Capability Review and Perspective”, Student on humanities and social sciences of Chiba University, PP.134-157(2006)(in Japanese)

小松原春菜：「Analytic Hierarchy Process とは何か」，千葉大学大学院人文社会科学研究科，人文科学研究 第 19 号 PP.134-157(2006)
<http://mitizane.ll.chiba-u.jp/metadb/up/irwg10/jinshaken-19-10.pdf>（参照 2016.8）

(11) James McCaffrey："Test Run The Analytic Hierarchy Process", MSDN Magazine 2005, June (2015)（参照 2017.2）

(12) 樽本 徹也：「ユーザビリティ エンジニアリング」第 2 版、pp.31-42、旺文社、(2014)

(13) Karen Holtzblatt, "Contextual Design", Morgan Kaufmann Publishers, 1999

(14) T. Iijima, A. Nagao, H. Yonemori:"A Case on the Hydroponic Culture by Different Field Technology", Japanese Society for Engineering Education, NII-Electronic Library Service. PP. 364-365(2014) (in Japanese)

飯島卓弥，長尾明美，米盛弘信：「異分野技術の融合による水耕栽培の実現」，日本工学教育協会平成 26 年度工学教育研究講演会講演論文集，P364-365(2014)

(15) K. Nakajima, K Seo, S. Kitagami, T. Shiotsuki, H. Koizumi: "Development of Unmanned Aerial Vehicle Systems for Monitoring and Prevention of Bird and Animal Damage and its evaluation", IPSJ SIG Technical Report (2015) (in Japanese)

中島幸一・清尾克彦・北上眞二・汐月 哲夫・小泉 寿男：「鳥獣被害対策用監視・防除 UAV システムの開発と評価」，情報処理学会 電子図書館，IPSJ-CDS15014010.pdf (2015)