2016年12月1日発行
第12巻 第3号(通巻50号)
ISSN 1349-8622

SEC journal 47

巻頭言

徳田 英幸

慶應義塾大学環境情報学部教授
一般社団法人 重要生活機器連携セキュリティ協議会 会長

所長対談

つながる世界の情報セキュリティと人材育成を考える

後藤 厚宏 情報セキュリティ大学院大学 情報セキュリティ研究科 研究科長・教授

論文

組込みシステムにおける検証アーキテクトと育成プログラム

西原 秀明 国立研究開発法人 産業技術総合研究所／大野 喜宏 株式会社インサイト
木村 浩司 AVC テクノロジー株式会社／瀬野 恭彦 組込みシステム産業振興機構

Goal Structuring Notationを用いた汎用的な安全要求の明確化と評価

柿本 和希・高井 利憲・飯田 元 奈良先端科学技術大学院大学 情報科学研究科
川口 真司・石濱 直樹・片平 真史 宇宙航空研究開発機構

特集

セキュリティ設計・高信頼化設計

つながる世界におけるセキュリティ

中尾 昌善 SECソフトウェアグループリーダー／宮原 真次 SEC研究員

セキュリティ・バイ・デザインとアシュアランスケース

金子 朋子 SEC研究員

組込みシステム セーフティ・セキュリティ検討WGの取り組み

石田 茂 SEC調査役

最先端ICT技術の実証プラットフォーム

水落 祐二 国立研究開発法人 情報通信研究機構 総合テストベッド研究開発推進センター テストベッド連携企画室長

つながる世界の検証

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

ICTシステムのサイレント故障・予兆監視

西川 昌利 NEC・クラウドプラットフォーム事業部・主任

報告 SEC BOOKS活用事例

技術力向上を目的に全社で活用／研修事業の教材として活用

Column

シェアリング・エコノミーの時代、「所有」から「利用」へ

IPA 独立行政法人 情報処理推進機構
Information-technology Promotion Agency, Japan

SEC journal No.47
Vol.12 No.3 Dec. 2016

巻頭言

IoT時代のセキュリティ課題

徳田 英幸
慶應義塾大学環境情報学部教授
一般社団法人 重要生活機器連携セキュリティ協議会 会長

IoTの進化

世界中でIoT（Internet of Things）化の流れが加速しています。国内においても昨年10月に官民連携でスタートしたIoT推進コンソーシアムは、当初約900社の企業がメンバーでしたが、現在、2,200社以上の企業が参加しており、まさに、“IoT x ビジネス”といった社会的創発現象が起きています。IoTは、単に、モノ（物）のインターネットと訳されることが多いですが、実際には、身のまわりの物や人工物だけではなく、人や生物、ビジネス上のデータやプロセス、空間上の位置情報などあらゆるモノ（Everything）がインターネットに接続され、情報を交換し、相互利活用される環境です。更に、これらのIoT環境の中で、つながれたモノたちがサイバー空間と物理（フィジカル）空間をまたいで制御されているシステムのことをサイバーフィジカルシステム（CPS）と呼んでいます。IoTがInternet of Connected Thingsであるのに対し、CPSは、Internet of Controlled Thingsを指しています。ITSに代表される交通系のシステムや気象情報と連動して水資源を管理しているようなシステムを“システムにつながれたモノたちを制御するシステム”という意味でCPSと呼んでいます。

IoT/CPS環境では、ビジネスドメインを飛び越えて、いろいろな機器、サービス、データ、プロセスなどがつながり、連携することで新しい価値や新しいコネクテッドサービスが創出されています。また、AI技術を使ってシステムから送られてくる様々なデータを解析し、システム内の機械やデバイスなどの故障予防や予知にも積極的に活用され、より信頼性の高いシステムなどの構築に役立ってきています。

IoTのセキュリティ

一方、IoT環境では、第４次産業革命のように期待され“メリット”が強調されている反面、従来のインターネット環境でのセキュリティ脅威とは異なる新たなIoT環境での脅威や“リスク”に対しても“先回りした対応”をしていかなければ、社会システムのあらゆるところで混乱が発生する懸念があります。IoT環境で対象となるシステムは、大きく２つのグループに大別されます。１つは、CPSに代表される電力管理システムやITSシステムなど、社会インフラとしてしっかり管理されている機器やサービス群から構成されている重要インフラ系システム。もう１つは、個人の家庭に設置され、一度設定された後は、あまり管理が行き届かないIoTゲートウェイ、スマートホーム、スマート家電、スマート健康機器、スマホ、パーソナルロボットなどの生活関連系システムです。重要インフラ系も、生活関連系システムもいずれも複数のシステムが連携しているので、機器に障害が発生した場合や攻撃を受けた場合には他のシステムや連携しているサービスに伝搬し、被害が大きく広がってしまうリスクがあります。また、コネクテッドカー・サービスなどのように、自動車と連携したサービスの場合、車載器の脆弱性を突いてモバイルネットワーク経由で侵入し、ファームウェアを書き換え、車のエアコン、ワイパー、ブレーキ、変速、ステアリング機能などを乗っ取られたクライスラー社のJeepのような事例も報告されています。また、IoT環境では、多くの個人に関連したデータがクラウド内やスマートデバイス内にストアされる場合、それらの情報が漏洩や改ざんされてしまうリスクに対しても対処する必要があります。このように、スマートデバイス層、ネットワーク層、クラウド層、コネクテッドサービス層の４つのレイヤにおける新しい脅威に対して、システムの企画、設計時からセキュリティを意識し、ライフサイクル全体を考慮した“Security by Design”やプライバシー保護を考慮した“Privacy by Design”の実践が必須になってきています。

CCDSの活動

このような状況の中、一般社団法人 重要生活機器連携セキュリティ協議会（Connected Consumer Device Security Council）は、2014年10月に設立され、我が国の重要生活機器全体のセーフティ＆セキュリティレベルの向上を目指し、重要生活機器のセキュリティ技術に関する調査研究、ガイドラインの策定や標準化の検討、及び普及啓発を行い、ものづくり産業の発展、新規事業創造、そして国民生活の向上を目指して活動しています。2016年６月には、安心・安全に利用できるIoTサービスの実現に不可欠な“Security by Design”の考え方を広くIoT製品開発ベンダーでの普及を目指し、車載・IoT ゲートウェイ・金融端末（ATM）・決済端末（POS）の４分野の製品分野別セキュリティガイドライン第１版を公開しました。また、本ガイドラインは、IPAが策定した「つながる世界の開発指針」とも連携しており、各製品分野の視点で脅威やリスクをより具体的に想定してあり、関連文書として位置付けして構成されています。

今後も、IPA技術本部ソフトウェア高信頼化センター（SEC）、内閣サイバーセキュリティセンター（NISC）、IoT推進コンソーシアムなどと連携し、我が国における安心・安全なIoT環境の実現に向けて尽力していきたいと考えております。

つながる世界の情報セキュリティと人材育成を考える

情報セキュリティ大学院大学
情報セキュリティ研究科　研究科長・教授
後藤　厚宏

×

SEC所長
松本　隆明

IoT時代を迎え、様々なモノがネットワークを介してつながり連携することで、私たちの社会生活はより高度にかつ便利になっていくと期待される。一方で、今まで想定されなかったモノがつながることにより、情報セキュリティのリスクの増大は避けられない。人材育成を中心に情報セキュリティに関して多方面で活躍されている情報セキュリティ大学院大学教授で情報セキュリティ研究科長を務められている後藤先生に、新たな時代を迎えた今後の情報セキュリティ対策のあり方や求められる人材像についてお話を伺った。

情報セキュリティ大学院大学の取り組み

松本　IPAでは、様々なモノがネットワークを介してつながるIoTの時代に、セーフティやセキュリティに関して開発・運用時に、どういうことに気を付けなければいけないかを17の指針としてまとめた「つながる世界の開発指針」を策定しました。後藤先生にはその検討の中心メンバーとしてご活躍いただきましたが、今後ますます重要になる情報セキュリティ対策のあり方や人材育成などについてお話を伺いたいと思っています。最初に、情報セキュリティ大学院大学の概要を教えていただけますか。

後藤　厚宏（ごとう　あつひろ）

1984年東京大学大学院 情報工学 博士課程修了（工博）。同年よりNTTにて情報技術に関する研究開発に従事。情報流通プラットフォーム研究所長、サイバースペース研究所長を歴任。2011年に情報セキュリティ大学院大学教授に転身し、2014年より研究科長。enPiTセキュリティ分野代表として、情報セキュリティ人材育成にも尽力。IEEE Computer Society理事。情報処理学会フェロー。現在、内閣府 SIP プログラムディレクター（重要インフラ等におけるサイバーセキュリティの確保）。

後藤　本学は2004年に設立されました。きっかけは2001年の米国における9.11事件です。9.11以降、広い意味でセキュリティに関する見直しが必要だということから、岩崎学園の岩崎理事長が、情報セキュリティについての高等教育を専門とする教育機関の設立構想を描きました。準備期間を経て2004年に開学、現在12年を経過したところです。

修士課程と博士課程からなる大学院のみというユニークな構成ですが、役割は日本における情報セキュリティのリーダーを育てることです。また、社会人に広く門戸を開放し、学びやすい環境を用意することに努めてきました。例えば平日の夕方や土曜日にも授業を設け、働きながら学べる機会を作っています。それが特徴の一つです。

もう一つの特徴は、セキュリティは決して技術だけの話でも、また、法律だけの話でもありません。組織のマネジメントも含め、すべてを融合したものであると考えています。そこで当初から、社会科学系の講義と技術系の講義の両方を用意しています。単一専攻なので、すべての学生が両方の授業を受けることになります。ただし、講義の選択に迷う学生が出てくるかもしれませんので、目指す方向に分けてモデルカリキュラムのセットを用意しました。最も技術系寄りなのが［数理科学コース］で、暗号のアルゴリズムとかビッグデータのような統計のアナリシスの考え方を中心に学びます。社会科学的なものでは［セキュリティ/リスクマネジメントコース］、それから技術と社会科学の両方を必要とする［サイバーセキュリティとガバナンスコース］と［システムデザインコース］があります。

［システムデザイン］と［サイバーセキュリティとガバナンス］は、それぞれ物を作る段階と運用する段階でのセキュリティを扱います。しばしば「システムデザインは技術だけの話ではないのか」と言われるのですが、それは違います。システムを設計する段階から、システムがどう使われ、どういう法律や制度に合わなければいけないのかということが分かっていなければ安全なシステムの設計はできません。一方、運用時にサイバーセキュリティ上の事故が起こったときには、対策を考え「犯人」を探すわけですが、そのときにも、どういう法律がかかわるのか、という知識が必要です。例えばマルウェアかどうかの分析をするときには、著作権法がかかわってきます。

松本　マルウェアでも著作権を気にするのですか？

後藤　そうなんです。マルウェアと確定するまでは、そのソフトウェアに著作権があるということを意識しないといけません。もちろんマルウェアの管理がまずいと自分がまき散らしてしまいますから、それも警戒しながら、法律を知った上で扱い、分析をしなければいけないわけです。

松本　情報セキュリティ大学院大学における教育の狙いはトップレベルの人材育成というところにあるのでしょうか。

後藤　修士課程を卒業したからといって直ちにトップレベルだ

とは言えないかもしれませんが、それを目指す人、あるいは次の世代を教育できる人を育てることを考えています。

松本　企業で言えばキャリアパスのようなことを意識しながらカリキュラムの選択ができるところは大きな魅力ですね。

後藤　例えば20代で来ている方は技術を磨くという志向が強く、30代以上でとくにIT企業から来ている方は、既に技術レベルは高く一定の自信もある、それを活用する組織構造や法律などを学びたい、という方が多いです。

松本　大学のカリキュラムというと、画一的にコースが決められていて、それに従ってやっていくのが一般的だと思います。キャリアパスを意識しながら、その人なりにキャリアが積めるという点はユニークですね。

後藤　設立当初から産業界の方に意見をいただく“応援団”のような組織を持っていたので、いろいろなリクワイヤメントを出していただきながら作ってきました。その成果だと思います。

セキュアシステム研究所では産学共同研究を進める

松本　後藤先生は情報セキュリティ研究科長を務められると同時に、大学内に設置されているセキュアシステム研究所の所長もされています。この研究所はどういう役割を持っているのですか？

後藤　社会や産業界と共同プロジェクトができる場所として位置付けています。客員研究員として日本や世界のトップレベルの人も招きながら、いろいろなプロジェクトを進めていて、例えばセキュア法制を考えたり、マルウェアの分析を進めたりしています。また、法律関係ではシンクタンク的な役割を持たせています。セキュリティに関する法制度をどう作っていけば良いのか、何をカバーしていかなくてはならないのかといったことを各方面の専門の方が真剣に議論する場所にして、年に１回レポートにまとめて提言しています。

松本　実際に社会に対して提言をしていく機関でもあるわけですね。

後藤　そうです。同時に技術面であればマルウェアの現物を持っている人に入ってもらって共同研究もしています。

松本　解析ツールの制作もするわけですか。

後藤　ツール制作というより、こういうアイデアで取り組んではどうか、という研究や提案ですね。

松本　セキュリティは産業界と連携して対策を進めないと現場で実際に起きていることへの対応が難しい。そういう意味で研究所の存在は大きいですね。

後藤　セキュリティの分野はまだ研究の仕方も対策も確立していない“なまもの”の状態ですから、現場の人と研究者が共同で作業していかなければなりません。

松本　確かに、ある物理法則に従って研究だけを進めていけば良いという世界ではないですからね。人の問題やガバナンスの問題など、幅広い分野から考えていかなければなりませんね。

国が進めるサイバーセキュリティへの取り組み

松本　先生は国の取り組みについても重責を果たされています。とくに文科省の「enPiT-Security」（エンピットセキュリティ）では代表として活動されていますが、どんなことをされているのですか？

後藤　エンピット自体は文科省の事業で、４分野にまたがるものです。一つがセキュリティ、それからクラウド、組込みシステム、ビジネスアプリケーションです。それらについて日本のそうそうたる教育機関が分担して活動しています。セキュリティ分野では本学の他、東北大学、慶應義塾大学、奈良先端科学技術大学院大学、北陸先端科学技術大学院大学の５大学が協力して取り組んでいます。特徴は５大学が共同でセキュリティに関する大学院のコースを作っているところですね。

　このコースは、今回のエンピットの狙いでもありますが「実践力を育成する」ということを考えており、座学を少なめにして多彩な演習――セキュリティの世界では「ハンズオン」という言い方をしますが――を実施し、セキュリティの実践的な様々な技術を、学生が体験を通して学ぶことに力を入れています。また、実践力を高める一環として、産業界の方から直接話を聞くことができる機会を数多く設けています。１年を通して所定の単位を取得すると修了の証であるサーティフィケイトがもらえます。ゆくゆくはこのサーティフィケイトを持っている学生が産業界で引っ張りだこになる、という状況を期待しています。

松本　大学にはそれぞれ校風のようなものがあると思いますが、５大学で共通のコースを作るということに難しさはありませんか？

後藤　確かに大変です。校風だけではなく、成績の点数の付け方も違いますし、細かいことを言えば、修了判定のためのデータの締め切り日も違います。とにかく最初は徹底した打ち合わせを重ねました。更に今は毎月５大学の先生が集まって運営について議論しています。

松本　なるほど、なかなか大変ですね。しかし、大学が連携して一つのコースを設けることには意味がありますね。

後藤　大学ごとに得意不得意があるんです。例えばある大学は暗号に強い、またある大学はネットワークに強いといった具合です。その強みを束ねてひとつのコースにしているところは学生にとって大きな意味があると思います。

松本　先生は内閣府の「SIP（重要インフラ等におけるサイバーセキュリティの確保）」という取り組みでプログラムディレクターをされていますが、こちらはどのような内容ですか？

後藤　SIPというのはCross-ministerial Strategic Innovation Promotion

松本　隆明（まつもと　たかあき）

1978年東京工業大学大学院修士課程修了。同年日本電信電話公社（現NTT）に入社、オペレーティング・システムの研究開発、大規模公共システムへの導入SE、キャリア共通調達仕様の開発・標準化、情報セキュリティ技術の研究開発に従事。2002年に株式会社NTTデータに移り、2003年より技術開発本部本部長。2007年NTTデータ先端技術株式会社常務取締役。2012年７月より独立行政法人情報処理推進機構（IPA）技術本部ソフトウェア・エンジニアリング・センター（SEC）所長。博士（工学）。

Program、つまり戦略的イノベーション創造プログラムと呼ばれるもので、社会イノベーションを起こすテクノロジーの開発を目指しています。２年半ほど前に10テーマでスタートしました。航空機材料やパワーエレクトロニクス、その他のデバイスなどの基礎研究を産学で立ち上げようというのが中心です。その後、2020年オリンピック・パラリンピック東京大会に向けて、セキュリティが大事だ、とくに重要インフラのセキュリティが弱いのではないかということから、昨年の夏にサイバーセキュリティの確保が11番目のテーマとして設定されました。プログラムディレクターが公募されましたので、本学は社会のためのサイバーセキュリティ技術の教育研究を担っていますから、取り組みは当然と考えて立候補し、選任されたという経緯です。

ビルトインとボルトオン ―セキュリティの２つの技術

松本　重要インフラのセキュリティ確保について、具体的にはどういう取り組みを進めるのですか。

後藤　日本政府では重要インフラとして13分野を定義していますが、その中からオリンピックに直結しそうなものにフォーカスし、その産業自体のセキュリティの体力を強める土台作りをしようと考えています。とくにエネルギー分野の電力、それから交通、通信が基本です。セキュリティの強化は２つの観点から考えています。１つ目はインフラシステムを作り上げるときに最初から仕込んでおかなければいけないものです。セキュリティ・バイ・デザインと呼ばれるものですね。いわばセキュリティ機能を“ビルトイン”しておくということです。インフラの場合、設備の更改は何十年単位で行われますから、大変時間がかかります。計画的にしっかり作り上げていくことが必要です。２つ目は、すぐ今日、明日のためにセキュリティを強化しなければならない設備への対策です。つまり、今ある設備を横から動作状況を監視したり、不具合を見つけ出す技術です。こちらは“ボルトオン”と呼んでいます。

松本　なるほど、最初の構築時に使う技術と運用時に使う技術ということですね。

後藤　その二つが基本です。更に言えば、これからのIoT時代に向けて、長く使える要素技術やセキュリティ技術を社会実装するための技術も必要になります。現場に入れ込むということです。単に技術として入れ込むだけではなくて、入れ込むことを助ける仕組み、例えば、適合性の評価や認証制度、人材育成、組織が情報共有する仕組みなども考えられます。コア技術を導入することを助ける仕組みについても研究目標に入れています。

まず現場で気づくことが重要

松本　情報共有の仕組みとして、IPAではサイバー情報共有イニシアティブ（J-CSIP〔ジェイシップ〕）の運用を始めていますが、そういう仕組みを更に活用していくということですか？

後藤　そうですね。情報共有については、例えば現場のオペレーション担当者が「サイバー攻撃かもしれない」と気づいたときに、どういうふうに組織の中に伝えれば良いか、現場の人にとって分かりやすいツールは何かを研究し、重要インフラの現場をJ-CSIPやJPCERT/CCなどとつなげられる仕組みを開発しています。いわば現場の人向けの情報共有システムですね。

松本　現場力を高めていくということですね。

後藤　重要インフラのセキュリティを考えたときに、どこが準備不足かというと、それは現場のオペレーションのところであり、そのための人材育成が必要だと思っています。大学院の学生だけではなくて、現場のエンジニアの人がセキュリティを学ぶための教材や演習コースが必要なんです。システムの不具合があったときに、これは故障ではなくてサイバーアタックかもしれないと判断できないとまずい。それができるようになるための教材が必要です。

松本　おっしゃる通りですね。セーフティという観点からですが、我々も重要インフラの障害の情報共有をきちんとやっていきましょうという取り組みをしています。そして、実際の障害をみんなで分析して何が課題であったかを議論して教訓の形にまとめています。その中にも、最初の気づきが周りの人にうまく伝わっていないために、障害が大きくなってしまったという教訓があります。現場の運用者は何かおかしいと感じていたのですが、そのままにしていた。マニュアルにも気づきを報告しろとは書いていない。その結果大障害につながってしまったという事例があります。現場をどう育てていくかというのはとても大きな課題ですね。

後藤　重要インフラの安定運用は現場の人の力で支えられています。これからはサイバー攻撃への対処も担っていただく必要があります。

松本　2020年オリンピック・パラリンピック東京大会は、セキュリティ上非常に大きなイベントになる可能性があり、とくに重要インフラは狙われるでしょうね。ロンドンオリンピックのときも相当な攻撃があったと聞いています。

後藤　あの頃から重要インフラが狙われ始めているんです。警戒を強めるべきだと思いますね。

専門化、分業化するサイバー攻撃

松本　情報セキュリティを取り巻く環境は大きく変化しているのではないかと思います。つながる世界の出現ということもありますが、それ以上に昨今は対策の範囲がどんどん広がっているような気がします。従来はITのWeb系の攻撃対策をどうするかという範囲だったと思いますが、最近は組込みの世界、航空機や自動車のセキュリティをどうしていくかということなど、範囲が広がる一方です。こうした最近の変化をどうご覧になっていますか？

後藤　おっしゃる通り、広がりは実感しますね。端的に言えば、大学に問い合わせをされる方の業界が変わりました。従来はセキュリティのベンダー側、つまりIT技術でいえば開発側の方が多かったのですが、今はユーザ側の方が増えています。とくに、自動車、家電、金融関係などですね。

松本　そもそも攻撃する側も、従来は有名になりたいといった愉快犯的な傾向が強かったけれども、最近は金銭目的とか、経済的に損失を与えるとか、あるいは国家に対する攻撃であるとか、質が変わってきていると思います。

後藤　言葉は悪いですが「裏社会」ができてしまっている感じがします。サイバー攻撃を使ってビジネスをしている人、あるいはそれを産業にしている国もある。市場が生まれ、サイバー攻撃をする人、それにお金を出す人、サイバー攻撃を成就することによって利益を得る人、この分業ができているんです。よ

く「何のために攻撃するんだろう？」という人がいますが、株価も原油価格も穀物相場も、サイバー攻撃によって大きく変動します。それによって儲ける人がいる。有名企業のWebサイトへの攻撃も、株の売買で利益を出すことにつながります。株価が下がればM&Aも有利に進められますね。大きなお金が裏の市場で動いてしまっているのです。そういう意味ではサイバーセキュリティは、技術だけではなくて国際的な取り組みが必要ということになります。

松本　攻撃も組織的なものになってきたということですね。

後藤　専門化、分業化しています。

松本　守る方も組織的に取り組まなければ、個別の対応では守れないですね。

後藤　おっしゃる通りだと思います。

松本　ところで、IoT時代になり膨大なモノがネットワークにつながってくると、データの拡散の問題も出てきますね。あらゆるところでデータが流通する。データを流通させ、共有することで社会をより便利にしていこうということだと思いますが、逆にリスクも高まるわけですね。データがいろいろなところにばらまかれ、それによって個人データの流出や企業情報の流出が起こる。

後藤　社会的にも産業上も、もはやデータの流通をストップすることはできません。データをうまく活用しないとビジネスが成り立たなくなっています。ITはまずオフィスを変えました。更に業務を変え、産業全体を変えた。データの活用で生産性は桁違いに上がっているわけです。もう後戻りはできません。そこは当然のことだと思って、情報流通の世界をいかに守っていくか、守るのは当然だという意識で動かなければいけない。今まで交通や物流の安全を守ってきたように、情報の流通を守るのは当然だという認識に立たなければいけないと思います。

松本　「データは社会的に重要な価値を持つものだ」と考えていかなければならないでしょうね。

求められる セキュリティ・バイ・デザインの取り組み

松本　セキュリティを取り巻く環境がどんどん変わってきたという意味では、これまでのセキュリティ対策は運用型であった。つまり攻撃されたときにどう守るかという対策が中心だったと思いますが、先ほどの後藤先生のお話にもあったように、これからはそもそも攻撃を受けにくくする、設計時にきちんとセキュリティを考慮してものを作っていくというセキュリティ・バイ・デザインの考え方が重要になりますね。

後藤　その通りだと思います。セキュリティを最初から仕込んでおくということですね。しかも、一つひとつのソフトウェア、ハードモジュールを作っている人のセキュリティ・バイ・デザインがあり、社会的なシステムとして、例えばオリンピックの会場を作るときにどうするかという段階のセキュリティ・バイ・デザインもある。何段階もあると思います。いろいろなレベルのセキュリティ・バイ・デザインが必要です。

松本　適切な例かどうか分かりませんが、例えば住宅や建築のバリアフリーを考えるときに、ここに階段があるからどうしようかという対策と併せて、そもそも最初に全体を設計するときに、バリアフリー化が可能となるように設計しておくということが必要ですよね。そういうふうにセキュリティの分野もなっていくべきなのでしょう。IPAでもセキュリティ・バイ・デザインに関する検討チームを立ち上げ、当面は組込み系を中心に議論を進めていこうと思っているのですが、実際のセキュリティ・バイ・デザインへの取り組みを、どうご覧になっていますか。

後藤　ひと言で言えば、まだまだだと思います。必要だという議論は出ていますが、実際の取り組みとしてはほとんど進んでいない。現場の開発部門の方も、言葉としては知っているけれども具体策としては持っていないという現状ではないでしょうか。先行例も乏しいので、早急に強化しなければいけないと思います。

松本　そうですね。実際に産業界の方とお話しすると「確かにセキュリティ・バイ・デザインの取り組みは重要だよね」という合意はできますが、「でもそれはセキュアコーディングのことじゃないの？」というふうに捉える人が多い。もっと全体的に見ていくことが必要ですよね。やはりまだ実例が見えないのだと思いますね。

後藤　実際にどうするかとなったときに、お手本がないんですね。そこは時間をかけても成功例を示していくしかないのかなと思っています。

松本　情報セキュリティ大学院大学では、そういう観点からの取り組みを進めていらっしゃるんですか？

後藤　まだ計画段階ですが、システムの設計時に脅威分析をどう組み込んだら良いかを学べる場を考えています。これまでは修士論文や博士論文でのチャレンジだったのですが、実際の講義に組み込み、更に演習に取り入れる予定です。今までもWebアプリケーションのシステムの脆弱性を見つけ出す演習を実施してきたのですが、では作るときに何をすべきだったのか、設計段階ではこうやるべきだった、動き出したらこういう観点からチェックすべきだと、両面から取り組むような講座を考えています。

松本　まず設計時にセキュリティ上で守らなければいけないのはこれだよね、というところからスタートしていくわけですね。

後藤　そうですね。このシステムはどう使われるのだろう、そのときに何を考えなければいけないんだろうというところからセキュリティ対策を考え、それが実際にできているのかをチェックするということですね。

松本　実際、何かシステムを想定して演習されるのですか？

後藤　脆弱性が多く残っているWebアプリケーションシステムを例として用意しているので、それを分析しながら、脆弱性が残らないようにするにはどうすれば良いかということを検討していく演習にするつもりです。しかしいずれにしても例題が少ないので増やしていかなければならないですね。組込みソフトがどうなっているか、重要インフラを支える大規模な制御システムがどうなっているか、それを順次検討の俎上に乗せていく必要があると思っています。

松本　脆弱性が多い設計と少ない設計が対比できると良いですね。どういうところに考え漏れがあったからこうなってしまったのかが分かるというように。

後藤　確かにそうですね。

プラスセキュリティという発想が必要

松本　セキュリティ・バイ・デザインの考え方を普及させていくためには、何がキーになるのでしょうか。設計の方法論のようなものを確立していくということでしょうか。

後藤　今物づくりをやっている会社は、会社ごとにルールや手

順があり、技術者はそれを身に付けて設計していると思います。その現場に馴染むものであることが必要です。すぐにはできないので少しずつ進めるしかないと思います。

松本　開発環境のようなものを整備していくことも必要なのでしょうか。セキュア開発環境みたいなものがあれば良いのかもしれません。

後藤　そうですね。それぞれの開発現場で新しい開発環境を実際に使えるようにすることが大事なのだと思います。とにかく実際にやってみて、こうやったらうまくいった、じゃあ真似してみようというような。狭い意味でのツール、手順書に始まって、人材の手当て、その育成まで含めてということになるでしょう。当然、設計のコストや期間という問題が出てきます。その点では産業に対するインセンティブも必要ですね。セキュリティ・バイ・デザインをしている製品は価値が高く、価格面でも高く売れるという市場があるべきだと思います。セミナー会場で言葉として「質の悪いものを出してしまって後々対策で苦労するより、最初からきちんとセキュリティをデザインしておけば、コスト面でもメリットがある」とは言えるのです。しかし、実際の現場の人の共感を得るのはなかなか難しい面があります。

松本　いったんセキュリティが破られたときの影響の大きさを自覚する必要があるかもしれませんね。しかも、バグがあるといったレベルの品質の話とセキュリティ上の品質管理の話では、大きな違いがあります。セキュリティでは外からのアタックに対してどうするかということが出てくる。普通の品質管理ならバグをいかに減らしていくか、ということになり、比較的閉じた世界の話で済みます。しかし、セキュリティの場合は外からの攻撃パターンとか、そういうことまで考慮していかなければいけない。

後藤　そうですね。想定と違うものが来たら、今までバグではなかったものがバグになるわけです。実は今、サイバーアタックの規模が非常に大きくなり複雑になっていて、運用段階では扱いきれません。解決できなくなっています。その複雑さを避ける意味でも、設計段階で頑張っていただくことが重要で、仮に運用段階で問題があったとしても、設計がしっかりしていれば対象がぐんと絞れるので、大きな意味があります。セキュリティアナリストが対応できる範囲に絞っていただければ対応が可能になるんです。

松本　セキュリティのスキルのある設計者を育てていくことが重要になるということですが、そのときに、セキュリティが分かっている人が開発に入っていくのが良いのか、開発がある程度できる設計者がセキュリティの勉強を積んだ方が良いのか、どちらが良いとお考えですか？

後藤　私は後者だと思っています。「プラスセキュリティ」と言っているのですが、それぞれの専門の方がセキュリティも習得する、これしかないのではないかと。物づくりの世界は非常に幅が広いわけです。物づくり以外でも社会システム的なものまで含めれば、例えば金融システムも含まれますから非常に広い。そこにセキュリティだけに特化した専門家を送り込むのは無理がある。であれば、自動車のエンジンを作っている人に少し時間をもらってセキュリティを勉強してもらう。金融のATMを作っている人に、セキュリティを勉強をしてもらう。そういうふうにしていかないとカバーしきれないのではないかと思います。セキュリティの専門家が、物づくりができるかと言えばそうは言えないですからね。それぞれの分野ドメインの人に「プラスセキュリティ」という取り組みをしてもらうのが現実的だと思います。実際、本学に入ってくる方の所属も、以前はICTベンダーの方が中心だったけれども、現在は金融であり鉄道であり、警察であったりしています。「プラスセキュリティ」の中で、今まで機械は作ってきたけれどもセキュリティは知らなかったという人に学んでいただいている。これは大事だと思います。そういう人のために、今後は大学院としての取り組みだけではなく、週単位の短期講座のような機会も作っていきたいと思っています。

松本　確かに「プラスセキュリティ」の考え方が基本なのかもしれないですね。実際の開発現場では、開発プロセスは既に厳密に決められているわけです。その中にどうやってセキュリティの考え方を入れていくべきか、と考えるべきですね。プラスするという発想ですよね。

後藤　開発現場に外部からこうしなさいと言っても絶対に変わってもらえないですね。今のプロセスを理解し分かっている人に、「あ、セキュリティが大事なんだな、じゃあどうすれば良いかな」と自ら気づいていただくことが一番重要なことだし、価値があることだと思います。

セキュリティ・ガバナンスの重要性

松本　もう一つ重要なテーマとして、セキュリティ・ガバナンスということがあると思います。そもそもセキュリティは技術だけでは守れない。ソーシャル・エンジニアリングなど人の要素も含めて考えていかなければいけない。法律面のこともあると思います。そのあたりはどうお考えですか？

後藤　ソーシャル・エンジニアリングについて言えば、ようやくいろいろな人が気づき出したなと感じます。本学でも社会科学と技術の両方が学べるようにしているわけですが、ただ単に学べるだけではなくて、お互いにぶつけ合ってみることに意味があると思います。私自身は技術系の人間ですが、実は法律の先生と一緒に一つの授業を担当しています。私が講義するときは、その先生が学生と一緒に聴いています。そして法律の観点から質問などもされる。逆に法律の先生が講義をするときは私が聴いています。そして学生と一緒に議論を深めていきます。両面あるんだ、ということがその場で理解できるので、非常におもしろいと思います。学生にも好評です。

松本　それはおもしろそうですね。

後藤　例えば認証とかITマネジメントの話を私がするのですが、それを個人のプライバシーとか法制度の観点からどうか。その場合、日本と欧米では違うとか、だから同じ技術ではだめだということが講義の場で分かってくる。技術の使い方の違いや普及度の違い、といったことについてディスカッションしながら考えていくといったことを進めています。技術だけでなくマネジメントやガバナンスが大事だよねということを講義の中で実際に理解するということに意味があると思います。

松本　セキュリティにとっては、法律と並んで、人間的な要素というか、心理学や社会行動学的な要素も重要ですね。

後藤　その通りです。２つあると思いますが、一つは大きな事件は内部犯行だったりしますね。するとそこには働いている人の満足度ということがかかわってきます。もう一つは「割れ窓理論」などと呼ばれていますが、ニューヨークのスラム街や地下鉄で、割れた窓や落書きなどを放置すると、ますます治安が悪化していくけれども、それを直すようにしたら犯罪が減った

といったことが言われています。同じように、ちょっとした社内ルール違反、ポリシー違反を認めていると、最終的に大きな情報漏洩に結び付くといった研究もしています。また、サイバーセキュリティではCSIRT（シーサート：Computer Security Incident Response Team）がオペレーション上非常に大事だと言われていますが、あれは組織経営学としてどう取り組むべきなのか、トリアージみたいな、何を緊急に選ぶべきかという判断であったり、どういう対策をとっていくか。最後は広報活動まで関係してきます。そういうことまで含めて、経営学や社会行動学の先生に来ていただいて講義を受けています。

松本　セキュリティをそこまで総合的に扱っている大学はほとんどないでしょうね。

後藤　そういう意味では貴重な学びの場所になっていると思っていますし、ニーズも高まっていると思います。

松本　セキュリティは経営上の課題として考えるべきだと思います。ただ、そういう意味ではセキュリティは負の側面が強くて、経営者はどこまで費用をかけるかという“コスト”の意識に陥りがちです。しかし、セキュリティを扱うことはプラスの要因なんだ、“投資”なんだという意識に変わっていかなくてはいけないと思いますね。

後藤　それは大事な点ですね。例えば有価証券報告書のようなところでもセキュリティ対策をどうしているかを情報開示して、きちんとしている会社は評価する、投資しやすくするという仕組みも必要なのではないかと思います。保険的な考え方というんでしょうか、自動車では安全運転を続けていれば保険料が安くなるということがありますね。企業活動においても、セキュリティ対策をきちんとしている企業は評価され、投資家が積極的に投資するという文化があるべきだと思います。

松本　運転免許証でも無事故無違反であればゴールド免許になり保険料が安くなるように、企業も長期間セキュリティ事故を起こしていなければ税金が安くなるとか、そういう施策があっても良いですね。セキュリティのプラス面が語られるような取り組みも必要だという気がします。

求められるのはクリエイティビティ

松本　最後に現在はセキュリティ人材が13万人も不足しており、2020年には19万人にも達すると言われています。早急な人材育成が必要だと思いますが、どういうスキルが求められるとお考えですか？

後藤　エンジニアに絞った言い方になりますが、必要なのは、創造性、クリエイティビティだと思います。物を作るときはすべてそうだと思いますが、バラバラにあるものを形のある物にしていく力です。問題解決のスタイルもそうですね。ある事象が起きているときに何が起こっているのかについて仮説を立てる。仮説をクリエイトして、検証して、間違っていたら修正し、情報を収集しながら目的の物を作り上げていく。自分のプラン、戦略に基づいて、それを解決していく。そのようなクリエイティビティが必要なのです。営業の人がマーケットを分析して物を売るときも同じです。ですからセキュリティの教育も総合教育なのだと思います。教科書にはない世界なんです。「OSの勉強をしました、ランゲージの勉強をしました、データベースの勉強をしました」でも、セキュリティの問題は全体に絡んできてしまう。物を作るときも、インシデントに対応するときも、何がいけないんだ、何が起きているんだ、ということを様々な知識を組み合わせ、様々な仮説をクリエイトしていく、その意味でのクリエイティビティが非常に重要なのだと思います。

松本　ある意味ではシステム思考ですね。個別の要素で考えるのではなくてシステム全体で俯瞰してみて開発設計していかなければいけない。IPAでもシステムズエンジニアリングをスタートさせてその重要性を考えてきていますが、まさにセキュリティの世界がそうなのですね。

後藤　大事なのは、上から俯瞰する全体的な視点と個別のしっかりした知識です。そこを目指さなければいけません。

松本　高度ですね。確かにソフトウェアだけ分かっていてもだめで、ハードウェアやネットワークなどの知識がないと全体のセキュリティ設計はできない。そう考えると人材育成はますます難しいという印象ですが、産学が連携した育成の取り組みを更に強化していく必要があるのでしょうね。

　グローバルに活躍できる人材の育成という点ではいかがですか？　企業もグローバル展開していますし、海外も含めてセキュリティが分かる人材というのはどうやって育成していったら良いのでしょうか。

後藤　大事なポイントですが、なおいっそう難しいテーマですね。簡単にはいかない。米国、英国、イスラエルなどはサイバーセキュリティへの取り組みが盛んです。交流しましょうということで海外の大学と国際交流協定（MOU）を結んで情報交換をしています。具体的には、例えばイギリスのロイヤルハロウェイに若手教員が１年間勉強に行ったりしています。イスラエルのベングリオン大学とも情報交換をして、カリキュラムや教え方などについて、お互いに見えるようにしようと話しています。また今後はASEANに交流を広げていくつもりです。留学生の派遣の後押しのような形でASEANにおけるセキュリティ人材の育成に貢献しよう、と考えているところです。

松本　セキュリティは一国だけの問題にはとどまらない。世界的にも連携して取り組んでいかなければいけない課題ですから、そういう意味でも日本はもっと世界に貢献したいですね。大変よく分かりました。本日は貴重なお話をありがとうございました。

組込みシステムにおける検証アーキテクトと育成プログラム


西原　秀明※1


大野　喜宏※2


木村　浩司※3


瀬野　恭彦※4

組込みシステムの検証における高度人材育成についての活動を紹介する．システム開発全体の中で検証を位置付け，幅広い視野とアイデアを基に検証をデザインし実施する検証アーキテクトの育成を目的に，その役割を定義し，必要なスキルを習得するためのカリキュラムを策定した．「組込み適塾」にてカリキュラムを試行し，有効性を評価した．

Fostering Verification Architects for Embedded Systems

Hideaki Nishihara※1, Yoshihiro Oono※2, Koji Kimura※3, and Yasuhiko Seno※4

This article introduces activity for fostering verification architects in embedded systems development. Verification architects specify verification activities in overall system development, conduct verifications with great insight, and contribute to the quality of products. A curriculum for verification architects is considered based on their expected roles in each phase of the system development process. The result was applied in Kumikomi-Tekijuku, an education program for embedded system architects.

1　導入

1.1　組込みシステムと検証

生活を支える機器やシステムにおいて，ソフトウェアが制御や機能の中心的役割を担うようになり，高機能化，複雑化が進んでいる．それに併せて機器やシステムの品質・信頼性の確保に対する考え方も変化し，様々な取り組みがなされている．

組込みシステムの開発では多くの場合，テスト工程に大きな比重をおいてシステムの検証を行っている．しかし例えば際どい操作に対するテストや妥当性確認では，観点の適切な選択や柔軟な発想といった属人的な作業が結果に影響することもあり，テスト技術の水準を維持していくのは容易ではない．またテスト工程は開発プロセスの後半に位置し，実施に制約がかかりがちで開発全体へのフィードバックも十分に行えない．検証工程を改善する余地はまだ大きく残っており，とくに工学的な観点をとることで開発全体を改善し信頼性の確保に貢献すると考えられる．

1.2　組込み適塾とキャリアガイド

組込みシステム産業振興機構（以下振興機構と呼ぶ）[1]では，組込みソフトウェアに重点をおいた高度人材育成プログラム「組込み適塾」を2008年より実施している．システムアーキテクト，すなわち開発対象である組込みシステムのアーキテクチャを把握し技術視点から開発を統

※1　国立研究開発法人 産業技術総合研究所
※2　株式会社インサイト
※3　AVC テクノロジー株式会社
※4　組込みシステム産業振興機構

括する技術者の育成を軸とし，開発の段階に合わせた「実装エンジニアリングコース」「アーキテクチャ設計コース」「アドバンストコース」の３つのコースを設定している．組込みシステム開発企業がカリキュラム検討に多く参画し，実践的な知識を体系的に身に付けることができる特徴的なプログラムである．

振興機構では，組込み技術者のキャリアガイド並びにキャリアマップ[2]を作成し，組込み適塾カリキュラムとの対応付けを行った．図１に2014年版のキャリアマップを示す．「実装エンジニアリングコース」がソフトウェアエンジニア，ハードウェアエンジニアのキャリアを広くカバーしており（図中（C）で示されている部分），同様に「アーキテクチャ設計コース」がミドルレベルのシステムアーキテクトを中心にプロジェクトマネージャとソフトウェアエンジニア，ハードウェアエンジニアを（図中（B）），「アドバンストコース」がハイレベルからミドルレベルのシステムアーキテクトをカバーしている（図中（A））．

キャリアガイドにはソフトウェアエンジニアやシステムアーキテクト，プロジェクトマネージャを含む「開発系キャリア」，開発プロセス改善スペシャリストや開発環境エンジニアを含む「支援系キャリア」に加え，検証・テスト担当者のキャリアとして「検証系キャリア」が設定されており（図１の点線枠），組込みシステム開発におけるテストの重要性を示している．一方で従来の組込み適塾のカリキュラムでは設計視点に重点がおかれていたため，キャリアマップを十分に網羅していない状態にあった．

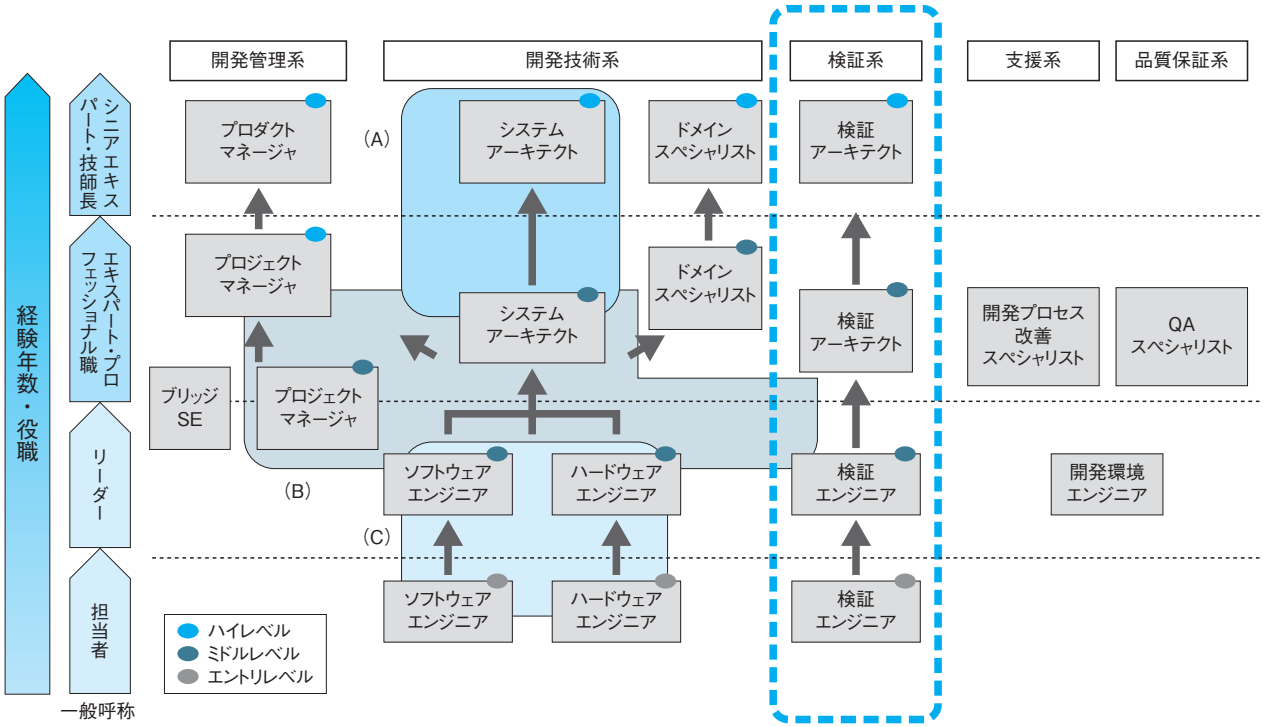


図１　キャリアマップとカリキュラムの網羅性

1.3　検証系人材の育成に向けて

このような状況のもと振興機構関連の企業から検証に関する教育への大きな要望があり，組込み適塾にて検証系の内容を拡充することとなった．大野喜宏主査の下，検証系人材育成サブワーキング（以下サブワーキングと呼ぶ）が組織され，本論文の著者を含め10名で活動を開始した．サブワーキングのメンバーは組込みシステム開発企業で開発経験を積んでおり，現場における検証の実態や課題を熟知している．またメンバーの多くが管理者的な立場にあり，開発全体の視点で必要とされる知識や人材について議論を行った．

サブワーキングは2014年度に６回開催され，現状と課題の共有を念頭においた討論，育成すべき人材像の明確化に続いてカリキュラム策定へと議論を進めた．サブワーキングでは，将来必要とされる検証技術者は開発全体における検証工程を位置付け，検証の各作業を設計するアーキテクトであるとの認識に至り，テスト観点の抽出，テスト結果の適切な分析とフィードバック，再発防止のための活動・提案をテーマとしてカリキュラムを検討した．

2015年には組込み適塾「検証アーキテクティング科目」として策定したカリキュラムが試行された．サブワーキングでは2015年度に６回の会合を行ってカリキュラムを評価し，改善点を整理した．その結果は2016年に実施されるカリキュラムに反映される．

1.4　本稿の内容

以上のように，著者らはサブワーキング活動の中で，組込みシステム開発における検証技術の向上を目指してカリ

キュラムを策定し実施した．検証視点のアーキテクトという概念は当時一般的ではなく，その役割やスキルについて一から検討して整理し，適切なカリキュラムを策定する必要があった．本稿ではその詳細と結果について述べる．

まずサブワーキングでは組込みシステム開発における検証アーキテクトの人材像について議論し，軸となるスキルと開発プロセスの中での位置付けを整理した．その結果を第二節で説明する．

育成したい人材像に基づいて，テーマ設定，順序，分量などを考慮しカリキュラムが作られる．サブワーキングの議論では，テスト観点の抽出，テスト結果の適切な分析とフィードバック，再発防止のための活動・提案が検証アーキテクト育成のテーマとして設定された．そこで，これらの３つのテーマに対応する３つの講義と，検証アーキテクトの活動全体を俯瞰する講義，合わせて４つの講義からなるカリキュラムが定められた．検証アーキテクトを対象とするため，個別の検証技術の知識よりは開発プロセス全体の視点から検証の効果を高める知識に重点がおかれていることが特徴である．カリキュラム策定の方向性や各講義の内容については第三節で説明する．

カリキュラムの試行と評価の結果については第四節で説明する．評価はサブワーキングによる講義内容のレビューと受講者の事後アンケート，講師のコメントを基にして行った．改善すべき点が幾つか指摘されたものの，検証アーキテクトに有益な知識や技能を提供するプログラムであることが確認された．今後，内容の拡充や見直しを進めつつ，継続してカリキュラムを実施し，検証アーキテクトの育成を続けていく計画である．

2　検証アーキテクトの人物像

2.1　検証アーキテクトについての現況

検証アーキテクト或いはテストアーキテクトはまだ一般的に知られた職種ではないが，テストエンジニア※1の一段上の視点から検証工程を捉え推進する技術者やスキルが必要とされていることは以下のように幾つかの文献で指摘されている．

ソフトウェアテストに関する国際規格IEC/ISO/IEEE29119[3]では，組織のテスト方針や開発計画を基に個々のプロジェクトにおけるテストの戦略や明示的な計画を定めるテスト管理プロセスが記述されている．

Microsoftではテストアーキテクトをおき，開発のエンジニアリングプロセスの改善に責任を持つポジションとしている[4]．更に同社のテスト体制について書かれた書籍[5]の第二章では，上級のテストエンジニアがテストアーキテクトの役割を果たしているケースがあることを指摘している．これらのエンジニアはテストのためのインフラ構築や複雑なテストを創造する際の事項の評価などを担当し，技術的観点から検証工程の向上に貢献する．

JSTQB[7]ではソフトウェアテスト技術者のスキルレベルをFoundation/Advanced/Expertの三段階に分けている．Advanced levelでは個々のテストプロジェクトにおいてパフォーマンスを向上させるテストマネージャ，テクニカルテストアナリスト，テストアナリストの３つの資格が設定され，更にExpert levelではテストプロセスの改善，管理，テストの自動化，セキュリティテストといった項目の資格が設定されている．

そして組込み技術者向けキャリアガイド※2[2]では，検証アーキテクトを「システムアーキテクトと共にアーキテクチャ観点で品質を確保し保証する技術者」と定義している．検証アーキテクトはシステムアーキテクトの成果のレビューなど上流の工程にも参画することが想定されており，その結果，テスト工程（とくにシステムテスト工程）にかかわるスキルに加えてシステム設計などシステムアーキテクトと同範囲のスキルが必要とされている．

2.2　期待される役割

これらの状況を踏まえ，サブワーキングにて組込みシステムの領域における検証アーキテクトの人物像を整理した．図１に示したキャリアマップでは，検証エンジニアの直接のキャリアアップ先としてミドルレベルの検証アーキテクトが示されている．つまり検証系技術者が中長期的なキャリアを考えるとき，最初の目標がミドルレベルの検証アーキテクトであると考えられる．よってサブワーキングでは検討のベースをミドルレベルの検証アーキテクト，すなわち検証の各工程を自立的に遂行できる技術者においた．

人物像の検討においては，技術的側面，経験的側面，ビジネス的側面のそれぞれから検証アーキテクトが持つべき知識や想定される活動を考察し，求められるスキルとして整理した．検討の過程では実際の組込みシステム開発における検証の課題と期待についても意見が出され，議論に反映された．例えば第三者的視点をとるべきか，開発者視点をとるべきかについて，若干の議論が起きたが，双方に利点と欠点があり一方に偏ることはしないとの結論になった．ただし，開発の現場では意識が開発者側に寄り気味なので，第三者的視点を少し強調して扱うこととした．開発企業での検証系技術者キャリアパスや製品開発における関係者との間の役割なども実状を基に検討されている．

結果として主に以下の三点を軸として組込みシステム開発に貢献できる人材を育成するとした．

A）　ユーザの「利用シナリオ」を想像力を働かせてイメージし，テストの戦略や評価に活かせる．実際の開発案件では設計開発側の担当者が検証を行うことも多い．開発情報から検証すべき個所を的確

脚注

※1　「テスト設計，テスト実行等のテスト作業の実施を担当する技術者」（ETSS[6]より）

※2　2014年にキャリアガイドの見直しが行われ，テストアーキテクト，テストエンジニア，といった呼称は検証アーキテクト，検証エンジニアと改められた．

に指定できる利点がある一方，例外事象や想定外の操作に気づきにくいという側面がある．ユーザ視点や第三者的な視点から利用シナリオを想像することで，テスト分析の範囲を広げ，検証を設計できる能力が必要となる．

B）　適切な手法を用いてテスト結果を分析し，製品品質に対する課題分析及び対策提案ができる．複雑化，大規模化する組込みソフトウェアの検証においても，工学的アプローチの重要性が高まっている．統計的手法やリスクベースの考え方を用いて，効果的にテスト結果を分析し不具合原因をつきとめ対策を立てる能力が求められる．また検証結果が設計開発側やマネジメント側にフィードバックされる際，根拠や対策の有効性を工学的な視点から説明できる能力が求められる．

C）　再発防止の観点で改善及び提案ができる．検証は個別製品の品質を確認して終わりではなく，そこで得られた知見を後続のプロジェクトに展開し組織的な品質向上に貢献することが望ましい．見落としやすい観点や技法・ツールの特徴を検証技術者間で共有したり，設計開発側にプロセス視点で改善点を提案し，同じような不具合を繰り返さない能力が求められる．

2.3　開発における位置付けと役割

検証アーキテクトは開発設計側や経営層へ検証視点で提案したり，各種の調整をしながら検証を進めていくことが求められる．よって開発体制においては図２のようにシステムアーキテクトと並立し，検証に関する部分を統括することになる．また，開発プロセスにおいては図３に示すように実装と単体テストを除いて開発全般に関与することになる．これらの点で，主に検証工程だけに携わる検証エンジニア（図３の点線部分）と検証アーキテクトは明確に区別される．

開発において検証アーキテクトが果たすべき役割を工程ごとに整理した（図４～図８）．図中にて検証工程に直接かかわる技術者を，実施を主に担当する検証エンジニア，プロジェクトの円滑な遂行を行う検証マネージャや検証リーダー，そして検証アーキテクトと分けた．更に設計開発側の技術者について記載し，開発全体における

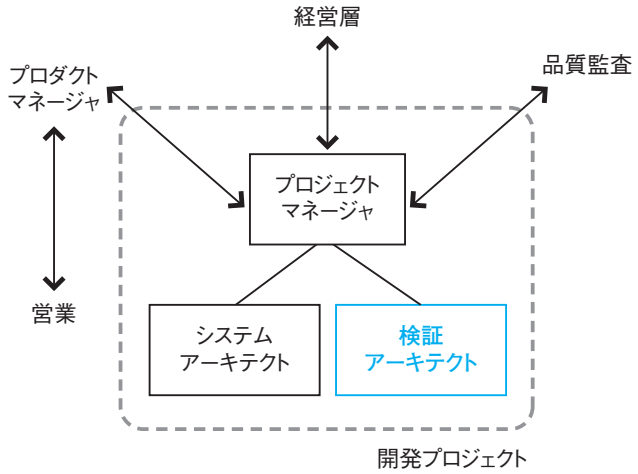


図２　検証アーキテクトの位置付け

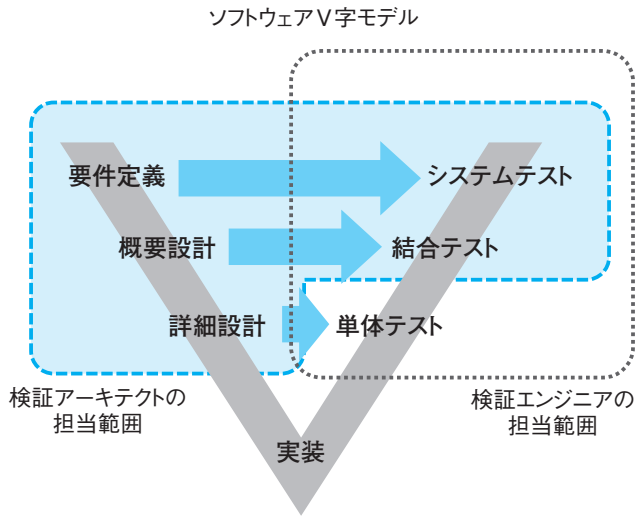


図３　検証アーキテクトの担当範囲

検証工程の各作業を位置付けた．実開発では一人の人物が検証マネージャと検証アーキテクト，またはソフトウェアエンジニアと検証エンジニア等複数の役割を果たしていることも多い．これらの区分は概念的なものであることに注意する．図中，検証アーキテクト等それぞれの技術者が行う作業を箇条書きで記載し，そのために必要とされる知識を枠を付けて表している（例：図４中の「アーキテクチャ設計技法」）．

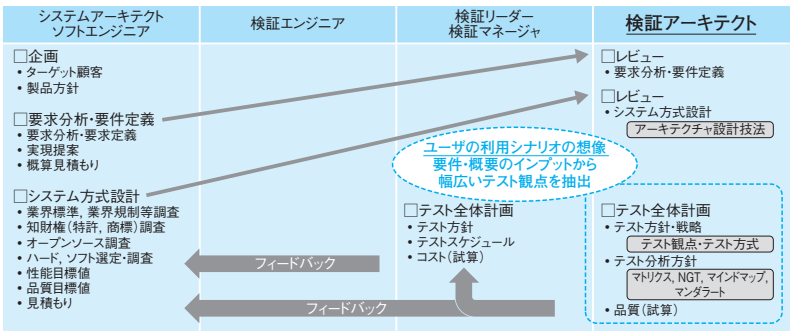


図４　検証アーキテクトの役割（要件定義）

要件定義：この工程では，検証アーキテクトはシステムアーキテクトの作成した要求分析・要件定義，システム方式設定を検証観点からレビューする．また検証マネージャと連携してテスト全体計画を策定する．検証アーキテクトにはテスト観点やテスト方式を検討するスキルに加えてアーキテクチャ設計や分析に関するスキルが求められる．検証アーキテクトに期待される役割のA）「ユーザの利用シナリオをイメージする」はこの工程で果たされる．

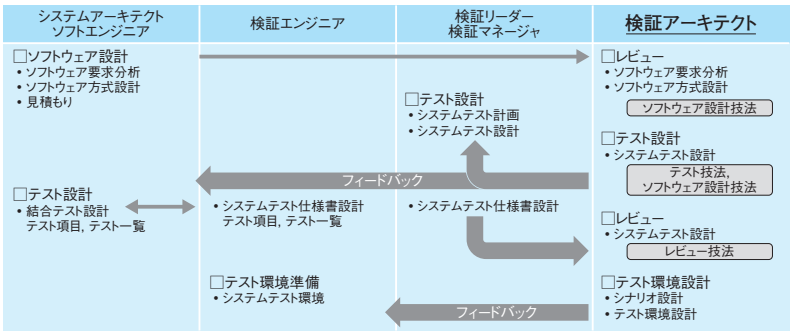


図5　検証アーキテクトの役割（概要設計）

概要設計：検証アーキテクトはソフトウェア設計ドキュメントを検証観点からレビューし，システムテストを設計する．またシステムテストの環境を設計し，テスト環境や使用ツールを設計する．検証アーキテクトにはテスト技法のほかソフトウェア設計やレビューのスキルが求められる．

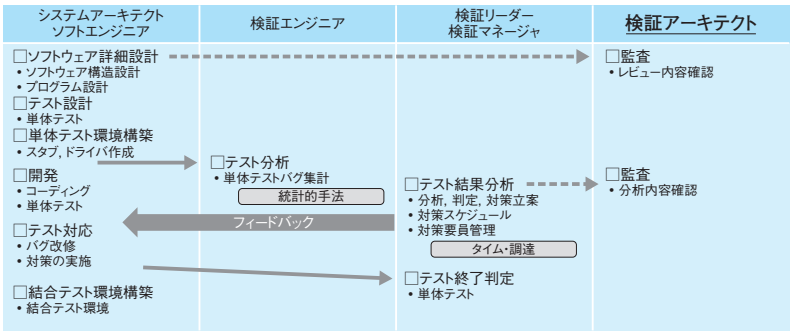


図6　検証アーキテクトの役割（詳細設計～単体テスト）

詳細設計～単体テスト：この工程における作業は単体テストまで含めて多くの場合設計開発側で実施されるので，検証アーキテクトはその結果，つまり詳細設計ドキュメントや単体テストの分析結果を確認することになる．

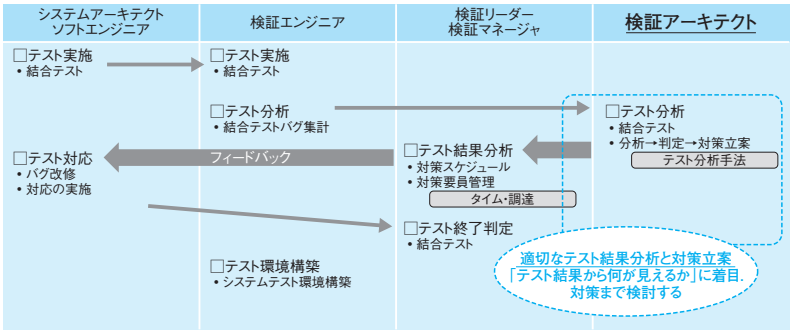


図7　検証アーキテクトの役割（結合テスト）

結合テスト：検証アーキテクトはテスト結果を分析し，不具合の原因解析や対処方針の策定を行う．実施に際して各種テスト分析手法を身に付けていることが求められる．検証アーキテクトに期待される役割のB)「適切なテスト結果分析と対策提案」はこの工程と次のシステムテスト工程で果たされる．

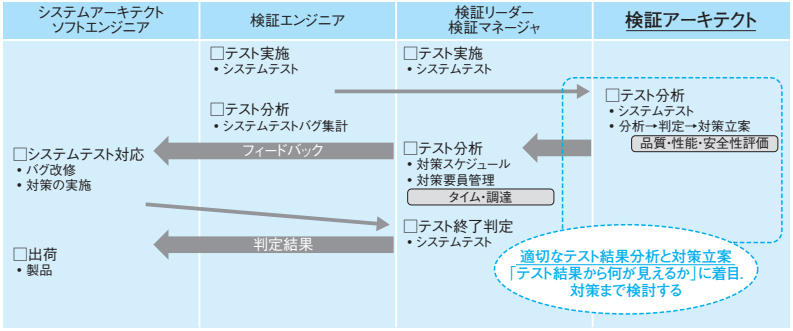


図8　検証アーキテクトの役割（システムテスト）

システムテスト：検証アーキテクトは第三者的視点からシステムテストを分析する．更に不具合の原因解析や対処方針の策定を行う．実施に際して各種テスト分析手法を身に付けていることが求められるほか，品質やシステムの性能，安全性に関して判断できるスキルが求められる．

検証アーキテクトに期待される役割のC)「再発防止の観点で改善及び提案ができる」は特定の工程と関連付けることができないことに注意する．他の開発案件や開発設計者に得られた知見を展開する活動は，個々の開発案件を越えている活動であり，展開すべき注意点や改善点は開発プロセスのいずれの工程からでも得られうる．

3　育成カリキュラム

3.1　カリキュラム策定の指針

前節で整理した人材像に照らすと，検証アーキテクトは単に知識が多ければなれるというものではなく，個々のプロジェクトに対して考察を加え，調整し，新たな発

想や視点を加えて検証を実施する能力が求められる．そこでカリキュラム策定の指針として以下の三点を設けた．

- グループ演習やワークショップを積極的に取り入れ，受講者自身が課題解決に向けて考えることを重視する．
- 事例を多く扱い，検証における課題が実開発においてどのように対処されるかを示す．
- ポイントを絞り，重要なスキルが確実に身に付くカリキュラムとする．

3.2 カリキュラムの講義体系

検証アーキテクトに期待される役割とスキルを核として講義で扱うべき内容を検討し，以下の４つの講義からなるカリキュラムとして整理した．組込み適塾の一部として実施することを想定し，一講義あたり0.5日或いは１日で実施する内容としている．

- [D04-01]組込み開発現場から見た検証アーキテクト（0.5日）
  カリキュラム全体の概観として，検証アーキテクトの位置付け，開発プロセスにおける役割，スキルを解説し，動機付けを行う．
- [D04-02]検証アーキテクトとしてのシステム分析・テスト設計演習（１日）
  図４に示したテスト全体計画と分析に関する講義．期待される役割A)で述べた，ユーザの利用シナリオを想像し，テストの戦略や評価に活かすためのスキルを養う．マインドマップのような発想の幅を広げる手法により，多様な視点から網羅的にテスト観点を抽出する演習を行う．また，その抽出結果を分析し質の良いテスト設計を行う演習を行う．
- [D04-03]テスト結果分析とフィードバック演習（１日）
  図７と図８に示したテストの結果分析・対策に関する講義．期待される役割B)で述べた，テスト結果を適切に分析し，対策を提案するためのスキルを養う．テスト結果のデータを分析し，事前のリスク評価を基に効果的に対処する手法を扱う．またプロジェクト中に起きたトラブルの対策をたて，関係者との調整などプロジェクトへフィードバックする演習を行う．
- [D04-04]事例から学ぶ検証アーキテクティング（0.5日）
  検証或いは開発の全体的な視点から，再発防止の観点で改善及び提案する際の知見について学ぶ（期待される役割C)．組込みシステムの開発事例において検証に関する活動から得られた知見と，それが以後のプロジェクトにどのように反映され不具合発生の予防につながったかを紹介する．

本カリキュラムの受講者は講義[D04-01]で検証アーキテクトの全体的なイメージを持ち，検証のプロセスに沿って講義[D04-02]と[D04-03]で具体的なスキルを身に付け，講義[D04-04]にて実開発での適用の際のヒントを得ることになる．

講義や演習で扱う題材，紹介する事例はハードウェア制御や組込みシステムから選び，課題や説明のポイントが組込みシステムに特徴的なものになるよう留意した．ただし講義内容の詳細については講師の知識と経験による所も大きく，講義の趣旨やカリキュラムを講師に伝え十分に調整を行うこととした．

組込み適塾では既に講義「テスト技法」が開講されている．こちらは組込みソフト開発者が一般的に持つ知識の習得に重点をおき，本カリキュラムの前提となる講義として位置付けられる．

4 育成プログラムの実施と評価

4.1 2015年の試行カリキュラム

サブワーキングの検討を基に，2015年度の組込み適塾[8]にてアーキテクチャ設計コース検証アーキテクティング科目として前節のカリキュラムを実施した．講義ごとのシラバスを表としてあげる．

講義名	シラバス
[D04-01]組込み開発現場から見た検証アーキテクト	1. 検証アーキテクトとは 2. 検証アーキテクトの役割 3. 開発現場の事例 4. 高信頼システム開発のための検証 5. まとめ
[D04-02]検証アーキテクトとしてのシステム分析・テスト設計演習	1. テスト計画とテストプロセス（座学） 2. テストの観点とテスト分析（座学） 3. テスト分析演習 4. テスト設計演習
[D04-03]テスト結果分析とフィードバック演習	1. 講義：テストプロジェクトの計画の立て方と管理手法 2. ワークショップ：テスト分析と対策トレーニング 3. ロールプレイ：トラブル対策会議
[D04-04]事例から学ぶ検証アーキテクティング	事例１（FPGA開発事例） 事例２（車載システム事例）

それぞれの講義につき，講義内容をよく知る有識者に講師を依頼し，扱う題材など詳細は講師とサブワーキングとの調整により確定した．例えば，[D04-02]は以下の流れで進めることとした．

- テスト設計の概要を座学で説明する．テストの戦略，設計の際の考え方（シラバス第１項「テスト計画とテストプロセス」）や，テスト設計における観点の重要性，分析手法（第２項「テストの観点とテスト分析」）を扱う．
- 分析手法としてマインドマップに注目し，発想を広げテストの観点を抽出する演習を行う．マインドマップの書き方を簡単に解説し，練習として自己紹介をテーマにマインドマップを書く（シラバス第３項「テスト分析演習」前半）．
- 現実的な題材に対してテスト分析演習を行う．受講者を数人ずつのグループに分け，自動販売機の機能仕様書を対象にマインドマップを作成する（シラバス第3項後半）．
- テストの戦略を定め，テストを設計するグループ演習を行う．前項の分析結果を題材とし，テストで確認したい品質や観点の優先度を決定する．更に，用いるテスト技法や範囲を検討しテストの計画を立てる（シラバス第４項「テスト設計演習」）．

講義[D04-02]と[D04-03]はそれぞれ一日の講義を想定していたが，講義内容の定着と演習内容の事前把握を狙い二日にまたがって開催した．つまり講義は午後に開始し，演習課題や演習で使う技法の紹介を一日目終了までに済ませておく．実際の作業は二日目に行うので，受講者は必要ならば一日目の講義終了後に復習や準備の時間をとることができる．

4.2 実施の概要

検証アーキテクティング科目は，アーキテクチャ設計コースの選択科目として実施された．組込み適塾では，「コース」「科目」「講義」の三層構造でカリキュラムを整理し，受講の単位としている（下表）．検証アーキテクティング科目はアーキテクチャ設計コースの共通科目であるベース科目の履修後に，方向性を絞った内容の知識習得を目指す課程と位置付けられた．

コース	科目	講義
アドバンストコース	システムズエンジニアリング科目	（個別講義）
	アドバンストシステムデザイン科目	（個別講義）
アーキテクチャ設計コース	ベース科目	（個別講義）
	コア技術科目	（個別講義）
	マネジメント科目	（個別講義）
	検証アーキテクティング科目	組込み開発現場から見た検証アーキテクト
		検証アーキテクトとしてのシステム
		分析・テスト設計演習
		テスト結果分析とフィードバック演習
		事例から学ぶ検証アーキテクティング
	システムデザイン科目	（個別講義）
実装エンジニアリングコース	基礎科目	（個別講義）
	実装演習（初級）	（個別講義）
	実装演習（実践）	（個別講義）

受講者募集は組込み適塾全体の募集に合わせた．受講要件は講義ごとに定めたが，全体としてはテスト専業の技術者には限定しなかった．一定のソフトウェア開発経験を持ち，主にシステムアーキテクトを目指す組込みシステム分野の技術者でとくに検証に興味を持つ者一般とした．これは，設計開発と検証の両方を経験しながらキャリアを積むケースが多いことがサブワーキングの議論で指摘されたことによる．

4.3 実施結果と評価

受講者募集に対して６名の応募があり（内１名は特定の講義のみ受講），カリキュラムの４つの講義すべてを2015年８月に実施した．受講者全員が組込みシステム開発企業に所属しており，主に開発を担当する技術者であった．

成績は講義ごとに評価した．課題レポートにて講義内容の理解度を評価し，また演習への参加の様子から自律的に作業や考察を進めているか，課題の達成度を評価し，総合して100点満点で評点をつけた．結果は最低80点，最高100点，平均86.9点であり，カリキュラムの内容を伝えられたと考えられる．

カリキュラム実施結果の評価は受講者への事後アンケートとサブワーキングによるレビューにて行った．受講者アンケートについては有益性の評価，また改善のポイントについての意見集約を主眼に質問を設定した．またサブワーキングメンバーで分担して各講義を見学し，評価コメントを整理した．更に講義の担当講師を交えて改善に向けて検討した．

受講者アンケートは９つの質問を設定し，原則として各設問に対して選択式の回答と付加コメントでの回答を求めた．質問の内訳は，受講目的（１問），講義内容の評価（５問），その他要望・現場の課題など（３問）である．講義内容の評価では，期待への満足度，受講効果，業務への有益性と，期待通りの部分，期待外れの部分の５問を質問した．図９～図11に示すように，科目全体としては受講者全員から肯定的な評価が得られた．更に「期待通りの部分」については，「テスト計画の概念をきちんと知ることができた」「すぐに活用できる情報が入手できた」として[D04-02]，[D04-03]の講義内容が挙がった．一方「期待外れの部分」については「業務に反映させるところが見えない」「内容が多岐にわたり消化しきれない」との意見が出たものの，特定の講義や内容に回答が集中することはなかった．現場の課題へのコメントで「テストの方針決めで困っている」「計画が不十分」といったものが得られている．総じて検証アーキテクトのスキルが開発現場の要望から外れておらず，カリキュラムの方針は適切であったと考えられる．

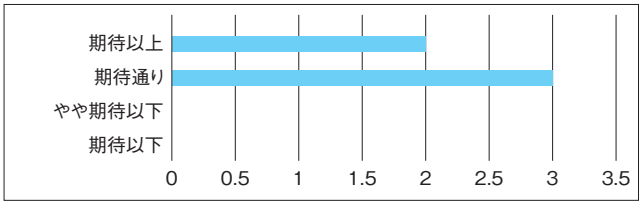


図９　検証アーキテクティング科目は全体として「期待通り」であったか

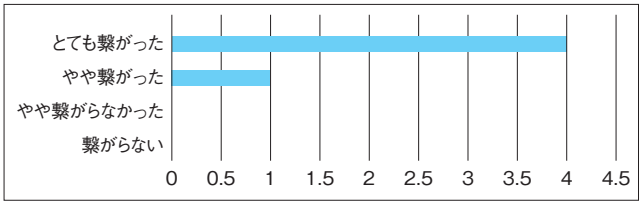


図10　検証アーキテクティング科目はスキルアップにつながったか

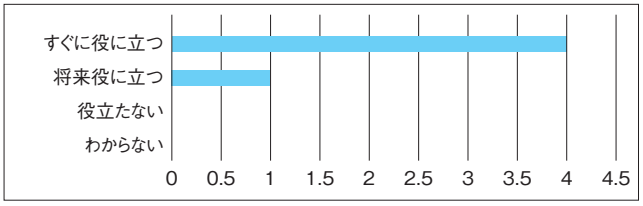


図11　検証アーキテクティング科目は自身の業務に役立つか

サブワーキングによるレビューでは，一つの講義あたり２～３人のメンバーが講義を見学してレポートを作成した．観点を以下のように設定し，各講義の章ごとに評価コメントと10段階の評点をつけて評価を行った．

- 講義の題材・講義内容が人材像に合致しているか
- 講義・実習方法が効果的か
- 受講生の興味・活性度
- その他見学者からのコメント

すべての講義が終了した後，レポートを基にサブワーキングで総合的に評価を行った．例えば講義［D04-02］に対する議論では以下の点が指摘された．

- 内容はサブワーキングの議論に沿っており問題はない．
- 当初，受講者の反応が堅く議論への積極性が足りない様子．座学が続き受講者間のコミュニケーションが不十分なことが原因と思われる．
- マインドマップの作成で受講者が迷っていた．技術の習得には試行錯誤が重要であるが，適当なタイミングで助言してはどうか．
- 演習題材（仕様書）の分量が多く，その理解に時間がとられているように見えた．
- 演習を進めるグループが一つだけだったので，アプローチの広がりや比較が不十分であった．

サブワーキングでの評価においては，資料の多さや講義間で説明の繰り返しが起きたことにより，内容と時間のバランスが適切でない講義が幾つかあったことが指摘された．講義で扱われる技法や事例は3.2節で述べたカリキュラムに沿ったものであるが，それらの意義や検証アーキテクトの視点，考え方にまで考察を深める余裕を受講者が持てなかったことが懸念される．講師担当者と十分に調整し，カリキュラムの特徴を伝えられる題材の選択，議論や総括のための時間確保といった対策により，育成プログラムの効果が高まると考えられる．

講義実施に対し，ほかにも以下の表に挙げるような改善事項が指摘された．それらへの対策も併せて検討されており，関連する資料の作成や講師へのフィードバックとして対応がとられている．

改善すべき点	対策
講義間の内容に重複がある，用語が統一されていない．	事前に講義内容を確認し調整する．
講義間の関連や講義内容の位置付けが受講者に伝わっていない．	目指す人材像や役割に関する資料を作成し，関係者（講師，受講者など）に早い段階で説明する．
グループワーク，議論の活発性が不十分．	講義冒頭でアイスブレイクを行う．多くの受講者を確保する．

5　総括

組込みシステム開発の検証系人材育成カリキュラムを開発し，組込み適塾にて実施した．カリキュラム策定に当たっては開発企業の要望・意見を取り入れ，開発全体の視点で検証を統括する検証アーキテクトの育成を検討のベースにおいた．実践的なスキル習得を目指して演習や事例を重視し，４つの講義からなるカリキュラムを定めた．

検証アーキテクトとして実開発に貢献していくためには，多くの事例や経験を基にして自身の検証に関する知識を見直し，実開発に適用できるものとして定着させる必要がある．本論文で紹介したカリキュラムは検証アーキテクトに求められる技術や視点を扱っており，検証技術の向上に有益だと考えられる．しかし知識の見直しと定着はカバーできていない．講義期間終了後に質疑・相談の機会を設ける，事例を紹介するなどのフォローアップにより育成プログラムとしての質の向上が期待できる．

試行にて一定の有効性が確かめられたため，次回の組込み適塾から本格的にカリキュラムを実施する予定である．カリキュラムの内容，実施体制などについて挙がった改善点については既に検討が進んでおり，次回は更に効果の高い講義を実施する．本カリキュラムにより高度な検証技術者が育成され，組込みシステムの品質や信頼性が向上することが期待される．

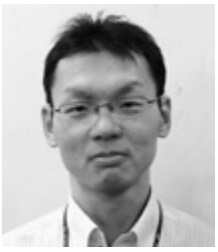
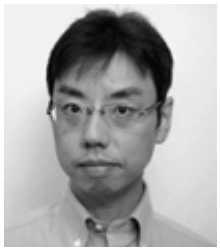
6　謝辞

本育成プログラムを組込み適塾の中で企画実施する際にご協力いただいた，検証系人材育成サブワーキングの皆様，教育事業部会長前川隆昭様並びに部会の皆様，組込み適塾塾長井上克郎先生に感謝いたします．またカリキュラム内容の検討に当たり宮崎大学工学教育研究部片山徹郎先生に多くの助言をいただきました．ここに記して感謝いたします．最後に本原稿に関し有益で的確なコメントをいただきました査読者の方々に感謝いたします．

参照文献

[1] 東田光裕, 岩井匡代, 八木浩, 奈良木英人, “組込みシステム産業振興機構の紹介,” SEC journal, Vol.9, No.3, pp.150-151, 2013.
[2] 井上克郎, “組込み技術者向けキャリアガイドの開発,” SEC Journal, Vol.8, No.2, pp.85-88, 2012.
[3] ISO/IEC/IEEE, 29119-2 “Software and systems engineering - Software Testing Standard -Part2:Test processes”, 2013.
[4] B.Rollison, “How We Test At Microsoft マイクロソフトでどのようにテストをしているのか？,” JaSST '12 Tokyo, 2012.
[5] A. Page, K. Johnston, B. Rollison, “How we test software at Microsoft(R)”, Microsoft Press, 2008.
[6] 組込みソフトウェア管理者技術者育成研究会，情報処理推進機構ソフトウェアエンジニアリングセンター, “ETSS標準ガイドブック”, 日経BP, 2006.
[7] JSTQB, http://jstqb.jp/index.html.
[8] 第八回組込み適塾講座一覧, http://www.kansai-kumikomi.net/ptraining/8th/index.html.

Goal Structuring Notationを用いた汎用的な安全要求の明確化と評価


柿本 和希[※1,a)] 川口 真司[※2,b)] 高井 利憲[※1] 石濱 直樹[※2,c)] 飯田 元[※1,d)] 片平 真史[※2,e)]

一般安全要求や安全に関する標準規格など，特定の分野のシステムに汎用的に適用される安全要求はあいまいな記述を含んでいる．あいまいな記述に対する解釈の誤りは要求の意図から外れた過不足のある設計につながるため，必要のないコストの増大や事故の原因となり得る．本研究では，汎用的な安全要求が暗黙的に仮定する知識などの暗黙知に着目し，それらを明確にすることにより，関係者の相互理解の促進やシステムの安全性の向上を目指した．具体的には，宇宙分野において用いられる，コンピュータによるハザード制御を行うシステムに対する安全要求（Computer Based Control System Safety Requirements，CBCS安全要求）を対象として明確化を行った．CBCS安全要求の明確化にはゴール構造化記法（Goal Structuring Notation，GSN）を用いた．更にその有効性を評価するため，宇宙航空研究開発機構の技術職員を対象とした比較評価実験を行った．実験の結果，一般安全要求の意図から外れた誤りの発見と訂正においてGSNによって正当の平均点が26%向上することを確認した．更に，GSN化によって思い込みによる危険性の見過ごしを防止する効果があることが確認できた．

Explication and evaluation of general safety requirements using GSN

Kazuki Kakimoto[※1], Shinji Kawaguchi[※2], Toshinori Takai[※1], Naoki Ishihama[※2], Hajimu Iida[※1], Masafumi Katahira[※2]

Safety requirements for a specific system domain, like general safety requirements and safety standards, tend to have obscure and ambiguous descriptions. Misinterpretations of them can cause excessively redundant or simply deficient safety design, which can be a trigger for cost escalation or an accident in the worst case scenario. In this research, we focus on implicit assumptions as a root of the ambiguousness mentioned above and propose a method to explicate an article in general safety requirements aiming for mutual understanding among stakeholders and improving system safety. The target of our research is Computer-Based Control System (CBCS) safety requirements, which are a safety standard for spacecraft systems, and the explication in our method is carried out by means of Goal Structuring Notation (GSN). In order to evaluate our proposal in a quantitative way, we performed a comparative experiment with the help of the Japan Aerospace eXploration Agency's engineers. The results of the experiment show that the average score for GSN-based CBCS safety requirements are 23% more effective in detecting and correcting errors in a given document on system safety than the average score for usual safety requirements written using natural language. Moreover, we conclude that GSN-based CBCS safety requirements can reduce misunderstandings among developers of a system and certifiers for safety.

※1　奈良先端科学技術大学院大学 情報科学研究科　Nara Institute of Science and Technology
※2　宇宙航空研究開発機構　Japan Aerospace eXploration Agency

1　はじめに

安全性にかかわるシステムの開発において，開発者はシステム固有の安全要求のほかに，特定分野のシステムに対して汎用的に適用されるような一般安全要求と呼ばれるものや，国際標準規格などの安全規格を考慮して開発を行い，定められた安全審査を受ける必要がある．一般安全要求や安全規格に準拠することにより保証を行うことや，審査を受けることはシステムが最低限の安全性を満たす客観的な証拠となるという点で重要である．

しかし，一般安全要求や安全規格の意図を理解した上で適切に安全性を審査することは困難である．その理由は大きく二つ存在する．一つめの理由は，審査活動にかかわる開発者及び審査者両方が持つ背景知識や経験に応じて解釈の深さが異なることである．例えば安全規格などでしばしば表れる「安全な状態」という表現をとっても，それを見るものが自動車組込みエンジニアか，航空機エンジニアかによって解釈が異なる．また経験の浅いエンジニアであればそもそも安全な状態が何かが分からない可能性もある．二つめの理由は，一般安全要求や安全規格は過去の設計や事故事例などの想定を踏まえて策定されていることである．評価対象システムを評価するには，これらの想定を理解した上で，評価対象システムでも同様の想定が成り立っているか，更には成り立っているとすればどの部分かを把握する必要がある．

一方，ゴール構造化記法（Goal Structuring Notation, GSN）[GSN2011]と呼ばれる記法を用いた安全性の保証が近年注目されている．従来の安全保証において，安全性に関する証拠が具体的にどのような安全性を保証しているのかはあいまいであった．しかしGSNを用いた安全保証においては，安全性にかかわる抽象度の高い主張が複数の具体的な主張に分解され，またそれらが客観的な証拠によって保証されることで，証拠が何を保証しているのかを明確にしている．

そこで本論文では，コンピュータによるハザード制御を行うシステムに対する安全要求（Computer Based Control System safety requirements，CBCS 安全要求）[NASA 1995]を対象にGSNを用いて明確化を行った．本研究における明確化とは，安全規格などにおける抽象度の高い条文が具体的にはどのような要求によって保証されるのかを明確にすることである．更にその明確化の試みが安全性の向上に有効かどうかを評価した．

以降，本論文は２節で背景を，３節でGSNを用いたCBCS安全要求の明確化の指針と実例について説明する．４節では評価実験の概要と結果を述べ，５節でその結果と限界について考察する．６節ではまとめとして，本論文の結論と今後の課題を述べる．

2　背景

2.1　Computer Based Control System安全要求

CBCS安全要求とは国際宇宙ステーション（International Space Station，ISS）の建造にあたってアメリカ航空宇宙局（National Aeronautics and Space Administration，NASA）が定めた一般安全要求の一つであり，ISSに関連するコンピュータを用いた制御機能を含むシステムに対して適用が義務付けられている．またCBCS安全要求の適用対象となるシステムの開発において，開発者はシステムがCBCS安全要求を満たすことをNASAに対して保証しなければならない．実際に日本が設計及び開発を行ったきぼう（The Japanese Experiment Module, JEM）と，こうのとり（The H-2 Transfer Vehicle, HTV）の開発においても安全審査が行われ，NASAに対して安全性の保証を行った．

2.2　CBCS安全要求適用時の課題

CBCS安全要求はISSに関連するシステムに広く適用されることを想定し，自然言語によって抽象的に記述されている．そのためCBCS安全要求を適用するには，抽象的に記述された条文において実際にはどのようなことが求められているのかを解釈しなければならない．しかし開発者や審査者の背景知識や経験によって，条文の解釈には幅が生じてしまう．また，過去の開発における解釈自体がCBCS安全要求を理解する上での暗黙知として存在している．背景知識や経験の違いによる解釈のずれは，条文の意図から外れた過不足ある設計につながるため，後の手戻りや事故の要因となり得る．

本研究では，CBCS安全要求におけるあいまいな記述がもたらす課題を以下の二つと考える．

課題１　条文を満たすために考慮する必要がある情報が，条文において明示されていない記述（読み取ることが難しい情報を暗黙的に仮定している記述）

課題２　条文の解釈がシステムに依存する記述

課題１及び２について実際の条文を用いて説明する．

CBCSは既知の安全な状態で起動する（箇条 3.1.1.1）

この条文は「システム起動時の初期化が完了した時点で既知の安全な状態であれば良い」と解釈することもできるという点であいまいである．しかし，本来の意図は「システムに電源を投入してから初期化を行っている間においても安全な状態を保つべき」という所にあり，安全審査においても起動中に安全性が求められる．ところが条文の本文中では起動中の状態については直接触れられておらず，背景となる知識がなければ起動中の安全化が求められていることは読み取れない可能性が高い．

また条文中の「安全な状態」という記述は，解釈がシステムに依存すると考えられる．なぜなら安全な状態はシステムごとに異なり，また同一のシステムであってもシステムの状態や周囲の状況によって異なり得るからである．このように条文の意図を理解して開発を行うには，明示されていない情報やシステムに依存する情報を知識や経験をもとに正確に解釈することが求められる．

2.3 Goal Structuring Notation (GSN)

GSNとはKellyら [Kelly1997]によって提唱された，安全性に関する議論を構造化するための記法であり，現在はGSN Community Standardによって記法が定義されている [GSN2011]．GSNでは，システムが満たすべき抽象的な要求をトップゴール(主張)とし，それらがストラテジ(観点)によってより具体的なサブゴールに分割される．ゴールの分割を繰り返すことで依存関係が可視化され，末端のサブゴールがそれぞれソリューション(証拠)によって保証されることでトップゴールが満たされることを客観的に保証することができる．ゴール, ストラテジ, ソリューションはそれぞれ長方形，平行四辺形，円形のノードで示される．記述例を図 1 に示す．

図 1 ではG1のトップゴールがS1のストラテジによってハザードごとの議論に分解されている．C1はコンテキストと呼ばれるノードであり，議論の背景や前提条件を示す役割を持っている．ここではハザードごとの議論を記述する際の前提となる，ハザード分析結果へのリンクを示している．またG3の下についているひし形のノードはundevelopedと呼ばれ，現段階では未定義であり，開発の今後において達成すべき主張や観点であることを示す．

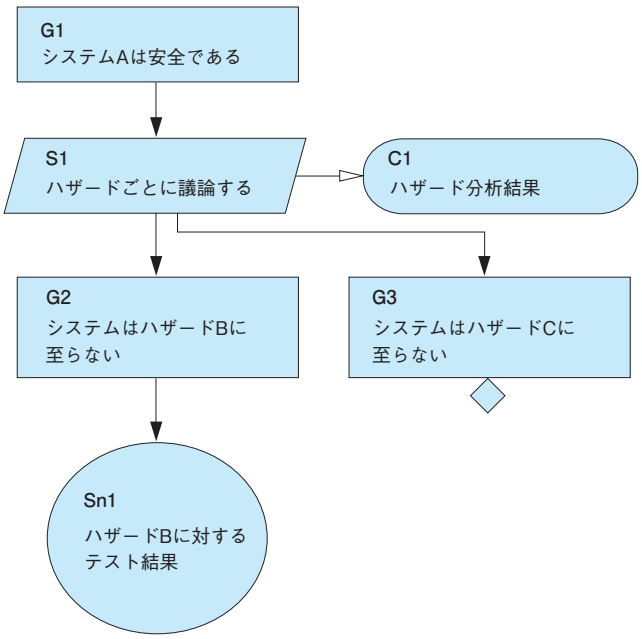


図1　GSNによる記述例

GSNを用いてシステムの安全性に関する主張を記述したものはアシュアランスケース [OMG2013] やD-Case [DEOS2015]などの形で標準化が進んでおり, 自動車 [Palin 2011]や医療 [Ray2013]，鉄道 [Souma2014]，航空宇宙 [Denny2012]，軍事 [Tanaka2012] などの分野で安全性の保証を目的に利用が検討されている．過去の研究においても無人軍事航空システムへの適用 [Denny2012]や超小型衛星への適用 [Tanaka2012]など高い安全性が求められるシステムへの適用において，更に，安全性だけでなくセキュリティの分野 [Kawakami2015]においてもその有効性が確認されている．

2.4 関連研究

一般安全要求や安全規格を用いた規範的な安全保証とGSNを用いた安全保証の比較を行った研究として，Hawkinsらは国際安全標準とGSNによる保証を実際に行った上で定性的な比較を行った [Hawkins2013]．Hawkinsらは，規範的な安全保証では規格に従うことで安全性にかかわる証拠をそろえやすいメリットがある一方で，それらの証拠がどのように安全性に寄与するのかは暗黙的であると主張した．またGSNを用いた安全保証においても，どのように安全性が保証されているのかが明確になる一方で，経験の浅い開発者はどのようにゴールを設定し記述すれば良いか分かりづらいとしている．結論として，Hawkinsらは安全要求，安全規格によって求められる保証を，GSNを用いて記述することで，二つの安全保証を相補的に用いるべきであると主張している．

GSNを用いて一般安全要求・安全規格の明確化を行った研究として, Hollowayら [Holloway2015]はDO178C [RTCA 2011] という航空機搭載システム・機器を対象とした安全規格の明確化を行っている．Hollowayらは既存の安全規格では求められるエビデンスがどのように安全性に寄与するかは暗黙的であると主張した上で，それらのエビデンスがどのような理由で求められているのかをGSNを用いて明確化した．しかしこの研究では設計に対する要求の明確化は行われておらず，また評価実験が行われていないためその有効性については明らかになっていない．

3 GSNを用いたCBCS安全要求解釈の明確化

本節ではCBCS安全要求の明確化を行った際の指針と実例について述べる．

3.1 CBCS安全要求をGSN化する上での指針

過去の開発で得られた，読み取ることが難しい情報を明確化する

課題 1 を解決するため，暗黙的に仮定された読み取る

ことが難しい情報を明示することで開発者と審査者との相互理解を支援できるように記述した．読み取ることが難しい情報の明確化にはJEMやHTVの開発における解釈から暗黙知を引き出し，それらをGSNに記述した．過去の開発において行われた過去の解釈をもとにした設計は安全審査を通過しているため，条文本来の意図に近い解釈が行われていると考えられるからである．

解釈がシステムに依存する記述に対してはステークホルダ間での合意を要件とした

解釈が対象となる個別のシステムに依存する記述は一般的な解釈を持たないため，過去の解釈を記述するだけでは条文の明確化を行うことができない．解釈がシステムに依存する記述に対しては，開発対象ごとに解釈を定め，それらについてステークホルダ間で合意を得る必要がある．

そこで課題２を解決するため，条文中のシステムに依存する部分を明確化し，それらへの解釈についてステークホルダ間で合意を得ることを目的とするゴールを導入することにより，GSN版CBCS安全要求における要求とした．

議論構造を設計と検証に分けて記述する

CBCS安全要求は設計に対する要求と検証に対する要求を明確に区別して構成されている．そのためGSNで記述する場合においても，要求が設計と検証のどちらに対するものなのかを明確にするため，GSNにおける議論構造上で分けて記述した．

明確化された条文に対して複数回のレビューを行う

CBCS安全要求の明確化において誤った解釈や不足した記述を防ぐため，一つの条文に対して安全審査の参加経験がある開発者二人以上により複数回のレビューを行った．

3.2 GSNによって明確化したCBCS安全要求（GSN版CBCS安全要求）の条文例

3.1節で述べた指針に従って明確化を行った．2.2節で例示したCBCS安全要求の箇条3.1.1.1の条文を明確化したものが図２である．一番上のゴールは実際の条文である．

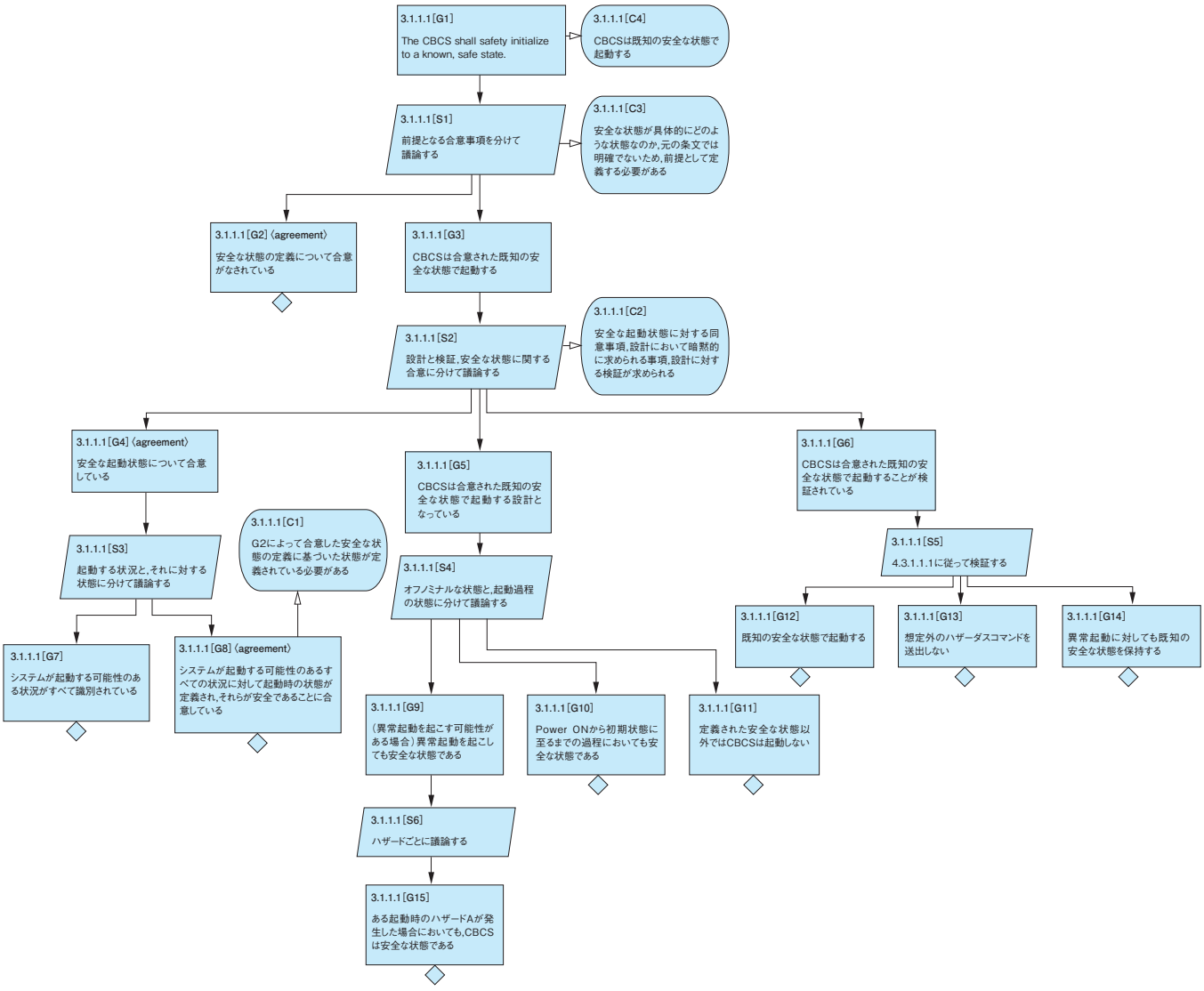


図２　GSNによって明確化されたCBCS安全要求の条文

課題１への対応として，読み取ることが難しい情報を暗黙的に仮定した記述に対して明確化を行った部分に対応するのがG5「CBCSは合意された既知の安全な状態で起動する設計となっている」以下からなる議論構造である．ここでは2.2節で示した例のほかに，異常起動を起こす場合でも安全化がなされていれば良いといったような例外事項などを記述している．

課題２への対応として，システムに依存する記述に対する明確化を行っているのがG2「安全な状態の定義について合意がなされている」はじめG4，G7，G8，S3からなる議論構造である．ここではシステム起動時の安全な状態を，システムが起動する可能性のある場合すべてに対して定義し，それぞれが安全であることについて合意することを求めている．

その他，CBCS安全要求をGSN化した結果は［Kakimoto 2016］を参照のこと．

4　評価

4.1　評価実験

GSN版CBCS安全要求に対する評価実験として，安全審査に対する有効性の調査を目的とした実験を行った．本稿における評価実験に関する諸元をまとめたものが表１である．評価実験においては，以下の点に着目して評価を行った．

１．安全性に対する効果
２．コストに対する効果

本評価実験ではJAXA技術職員を対象として安全審査を想定した評価を行った．被験者数は20名で，GSN版安全要求を用いたグループ（以下，GSNグループ）と従来のCBCS安全要求のみを用いたグループ（以下，従来グループ）の各10名ずつに分けた．グループ分けはGSNや対象システムに対する知識や開発経験の有無を事前に調査し，大きな偏りがないように努めた．

表１　実験諸元

評価対象	被験者数	設問設定	作業内容	実験時間
JAXA 技術職員	全20名 GSNグループ：10名 従来グループ：10名	HTVを想定した，誤りを含む 架空のハザードレポートに対する 安全審査	誤りの発見 対策の修正 自己評価	事前説明：10分 安全審査：～40分 自己評価アンケート：10分

各グループは公開されたHTVの情報［Torano2009, Shirasaka2011］をもとに作成した架空の「ハザードレポート」に対して安全審査を行った．ハザードレポートには想定される「システムの逸脱（想定外のシステムのふるまい）」と「逸脱の原因」が既知の情報として記載されている．また「逸脱に対応するCBCS安全要求の条文」及び「逸脱に対する対策」が記載されている（これらには誤りが含まれる）．被験者は対応する条文と対策がCBCS安全要求に沿った内容であるかを審査し，「誤りの発見」及び「対策の修正」を行った．「誤りの発見」では審査時の誤りの検出への支援効果を，「対策の修正」では開発時への支援効果の測定をそれぞれ想定している．評価実験で用いた問題文は［Kakimoto2016］を参照．

本実験では，下記３つの観点についてGSN版CBCS安全要求を用いた場合とそうでない場合でデータの収集を行った．

観点１　課題に対する正答数（全４問）
観点２　平均回答時間（最大40分）
観点３　達成度に対する自己評価（５段階）

観点１及び観点３では，それぞれ安全性に対する定量的，定性的な観点からの評価を行った．また観点２では安全審査にかかった時間から，安全審査におけるコストに対する効果をそれぞれ評価している．

観点１では被験者が誤りを発見し，対策を正しく修正できていたものを正解として採点した．観点２については，両方のグループに対して事前に10分の説明を行った上で，上限40分で課題の回答時間を計測した．また観点３について課題終了後にアンケート形式で達成度に関する以下の質問を行った．

質問a　誤り，不足を見つけることができたか
質問b　CBCS安全要求を満たすためにどのような対策が必要なのかが分かったか

各質問に対して被験者は「５：よくできた」から「１：できなかった」の５段階で回答した．質問aと質問bではそれぞれ，誤りの発見に対する達成度と対策の修正の達成度に対する自己評価の収集を目的とした．

4.2　実験結果

観点１に対する結果を図３に示す．図３では1Q－（IQR×1.5）～ 3Q＋（IQR×1.5）の範囲から外れたものを外れ値として扱っている．（1Qは第１四分位数，3Qは第３四分位，IQRは四分位範囲．）観点１において，GSNグループの正答数は従来グループに対して高くなっていることから分かる．また二つのグループの平均点の差が偶然誤差の範囲内で

あるという帰無仮説を立て，有意水準５％でWelchのt検定を行った．その結果，危険率2.77%で帰無仮説が棄却されたため，二つのグループには有意な差があると言える．

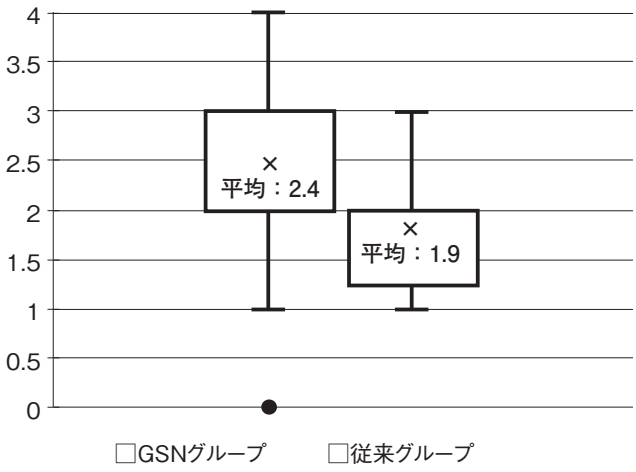


図３　正答数の分布

観点２に対する結果を表２に示す．観点２についてはGSNグループの回答時間が38.4分と従来グループ（33.7分）と比較して14%長くなった．これはGSNによってCBCS安全要求の解釈を記述することで情報量が増加し，GSNを読む作業とハザードレポートの情報をGSN版CBCS安全要求と比較する作業が加わったことが原因であると考えられる．

観点３に対する結果を図４に示すこの結果から誤りの発見と対策の記述の両方において，従来グループの方が高い自己評価を行う傾向にあることが分かる．

表２　平均回答時間

	平均回答時間
GSNグループ	38.4分
従来グループ	33.7分

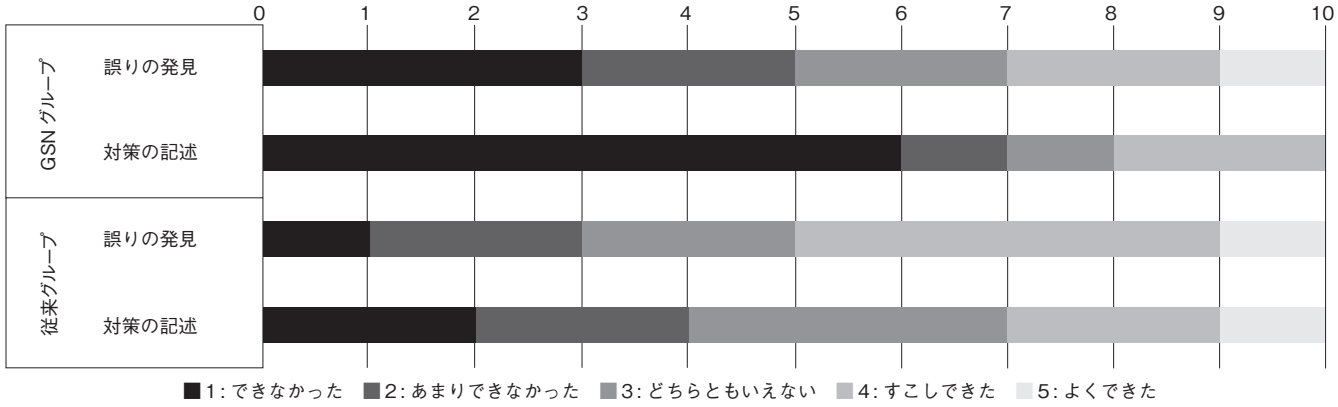


図４　達成度に対する自己評価結果

5　考察

5.1　実験結果に対する分析と考察

観点１の結果から，GSN版CBCS安全要求を用いることで安全審査の精度が上がることが分かる．しかし観点３の課題の達成度に関する質問への回答結果では，従来のCBCS安全要求を用いた場合のほうが課題を達成できたと答えた割合が大きく，実際の正答数と被験者の実感に差が生じていることが分かる．しかし課題の実際の達成度と自己評価には正の相関があるべきである．なぜなら，審査対象システムが安全審査を通過するのは，審査者が十分に安全であると判断した，すなわちその実感が得られたときだからである．このことから我々は以下の仮説を立て，追加の分析を行った．

仮説１　GSN版CBCS安全要求を安全審査に用いた場合，審査結果と審査者の実感に正の相関がない

仮説２　従来のCBCS安全要求を安全審査に用いた場合，審査結果と審査者の実感に正の相関がない

そこでこれら仮説について分析するため，課題の達成度に関する質問に対して１または２と回答したグループをネガティブ群，４または５をポジティブ群に分けて表３の比較を行った．また統計的な分析として，正答数と課題の達成度に関する自己評価への回答結果を用いてピアソンの積率相関係数（n=10）を求めた．

表３の結果からGSNグループでは正解数と質問への回答に正の相関関係が見られた．よって仮説１は成り立たず，GSNグループでは実際の正当数と被験者の実感に差が生じづらいと考えられる．一方，従来グループでは相関関係が確認できなかった．よって仮説２は成り立ち，従来グループでは実際の正答数と自己評価が必ずしもが関係しないと言える．

この結果からGSNによる暗黙知の明確化は経験の少な

い開発者だけではなく経験を積んだ，とくに自分の経験による設計で十分に安全要求が満たされると考える開発者に対しても有効であると考えられる．なぜなら従来の安全要求を用いた場合，経験を積んだ開発者が十分であると判断した場合においても不十分な設計がなされていた可能性があるからである．

また観点２ではGSN版CBCS安全要求を用いた場合の方が審査に時間がかかったことから，安全審査において従来よりも工数が掛かってしまうと予想される．しかし審査時間の増加は一割程度に留まっており，危険性の見過ごしによる手戻りや宇宙機システム自体の喪失，過剰な安全設計によるコストの増加に比べれば小さいと考えられる．

表３　正答数と自己評価の関係に対する分析結果

		全体の平均正答数	ポジティブ群平均	ネガティブ群平均	積率相関係数 (n=10)
GSNグループ	誤りの発見	2.78問	3.33問	2.25問	0.549..
	対策の修正		3.50問	2.5問	0.658..
従来グループ	誤りの発見	1.90問	1.80問	1.67問	−0.251..
	対策の修正		1.67問	2.25問	0.146..

5.2　限界と妥当性への脅威

本研究における一般安全要求の明確化の限界として，GSN版CBCS安全要求はあくまで条文に対する理解の助けや安全審査における議論の土台として利用することを想定していることが挙げられる．なぜならGSN版CBCS安全要求はCBCS安全要求を完全に満たすことを保証しておらず，従来のCBCS安全要求で明示的でない記述を過去の開発における解釈やエンジニアの知見の範囲で明確にしたものだからである．従って規格の認証プロセスなどの網羅性の求められる暗黙知の明確化においては，本稿において述べた手法に加えて網羅性の保証される情報の記述を行う必要があるといえる．

また明確化には情報の引き出し方に留意する必要がある．CBCS安全要求の明確化においては安全審査の結果や有識者間での議論の結果が記録されていた．よって条文の問題点や解釈が有識者間では共有されており，明確化を行う情報源となる文書の選定や有識者によるレビューを通して情報を引き出すことに対して特別な工夫を必要としなかった．しかしそれらの情報が有識者間においても共有されていない場合には，明確化の対象となる事項を明確にし，暗黙知の前提条件や特殊な状況下での制約といった引き出す際に漏れやすい知識に注意して情報を引き出すなどの工夫が必要であると考えられる．

内的妥当性への脅威

本研究の評価実験ではHTVを想定した架空のハザードレポートを用いたが，理想的な評価実験環境では実際のハザードレポートを用いるべきである．しかし現在運用されているシステムの，公表されていない安全性にかかわる問題を用いた結果を使って公に議論することは困難である．また架空のハザードレポートは公表されているHTVの情報に則って作成された上でJAXA技術職員のレビューをうけており，一定の妥当性は担保されると考えられる．

次に，４節の評価実験での採点の妥当性について述べる．被験者回答において，正しく対策の修正が行われているかどうかは我々の主観によって判断されている．しかし，採点には一定の基準を設けており，複数の評価者によって一貫して基準を適用していることをレビューしているため，主観による影響は最小限に抑えている．

外的妥当性への脅威

最初に，HTV以外のシステムに対するGSN版CBCS安全要求の適用性について述べる．CBCS安全要求は複数のシステムに対して適用される安全要求である．本研究ではHTVに対する安全審査を想定して評価実験を行ったが，本来は他のISSに関連するシステムに対しても評価実験を行うことが望ましい．しかし本研究における評価実験ではCBCS安全要求適用の一般的な手続きを採用しているため，他のシステムの保証においても同様に適用することができると考える．

次に，CBCS安全要求以外の，他の一般安全要求や安全規格に対するGSNを用いた明確化の有効性について述べる．Hawkinsら [Hawkins2013] は従来の安全規格は暗黙的であり，開発者が趣旨を理解するのは困難であると主張しており，CBCS安全要求と同様の問題がほかの安全要求においても存在することが分かる．GSNは議論構造を可視化するための汎用的な手法であるため，CBCS安全要求以外の一般安全要求や安全規格に対しても適用可能であると考える．しかし安全要求や規格ごとに保証プロセスの違いが存在するため，それらを考慮して明確化を行う必要がある．

6 おわりに

6.1 まとめ

本論文では宇宙分野で用いられる一般安全要求が実際にはどのような要件によって満たされるかを，GSNを用いて明確化した．一般安全要求の明確化は，開発者及び審査者の相互理解や安全審査の支援を目的として行った．また明確化したCBCS安全要求の有効性の評価を目的として，JAXA技術職員を対象とした評価を行った．評価の結果，GSNによって明確化されたCBCS安全要求を用いた場合，以下のメリット・デメリットがあることを確認した．

メリット1　安全審査による誤りの発見とその修正数が26%向上した．

メリット2　従来の一般安全要求では審査者の思い込みによる危険性の見過ごしが発生する懸念があるが，GSNを用いた明確化ではそれが見られない．

デメリット1　従来の手法と比較してGSN化した安全要求による審査の所要時間は14%大きい．

6.2 今後の課題

本研究で提案した，GSNによる暗黙知の明確化を一般化することが今後の課題として挙げられる．他分野への応用として，ソースコードレビューにおける暗黙知の明確化が考えられる．ソースコードレビューにおいては，チェックリストを用いたレビューが代表的な支援手法として使用されている［Bando2011］．過去の研究においてDengerらはレビュー者の経験に応じてレビューの観点を設定するべきである［Denger2007］と主張しており，開発者の経験はコードレビューの品質に影響を与えるとされている．しかし背景知識や経験に応じてチェックリストの観点を設定した場合においても，チェックリストの各項目は過去の設計や事故事例などの想定をもとに策定されると考えられるため，一定の暗黙知を含むと予想される．暗黙知を含むチェックリストによるレビューにおける判断基準はレビュー者によって解釈が異なると考えられるため，従来の安全要求と同様にレビュー者の思い込みによる危険性の見過ごしが懸念される．そこで，チェックリストに含まれる暗黙知を，GSNを用いて明確化することによってソースコードレビューの品質の向上が期待できる．

謝辞　本稿4節の評価実験にご協力いただいた，宇宙航空研究開発機構 第三研究ユニットの方々に深く感謝いたします．

参考文献

［NASA1995］SSP 50038B Computer-Based Control System Safety Requirements –International Space Station, 1995.
［Kelly1997］Tim Kelly and John A McDermid, Safety Case Construction and Reuse using Patterns, 16th SAFECOMP, pp.55-69, 1997.
［GSN2011］The GSN Working group, GSN COMMUNITY STANDARD VERSION 1, 2011.
［OMG2013］Object Management Group, Structured Assurance Case Metamodel (SACM), 2013.
［DEOS2015］DEOS協会 D-CASE部会, D-CASE構文定義書, 2015.
［Palin2011］Rob Palin, David Ward, Ibrahim Habli and Roger Rivett, ISO 26262 safety cases: Compliance and assurance, 6th IET International Conference on System Safety, pp.1-6, 2011.
［Ray2013］Arnab Ray and Rance Cleaveland, Constructing Safety Assurance Cases for Medical Devices, 1st International Workshop on Assurance Cases for Software-Intensive Systems, pp.40-45, 2013.
［Souma2014］相馬 大輔, 田口 研治, 西原 秀明, 大岩 寛, 矢田部 俊介, 森 崇, RAMS の認証とセーフティケース, クリティカルソフトウェアワークショップ (WOCS2: Workshop of Critical Software System), 2014.
［Denny2012］Ewen Denny, Ganesh Pai and Ibrahim Habli, Perspectives on Software Safety Case Development for Unmanned Aircraft, Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp.1-8, 2012.
［Tanaka2012］田中康平, 松野裕, 中坊嘉宏, 白坂成功, 中須賀真一, アシュアランスケースを用いた小型人工衛星の品質保証 – REAJ 第20回春季信頼性シンポジウム, pp.63-66, 2012.
［Kawakami2015］Henrique Kawakami, Roberto Gallo, Ricardo Dahab and Erick Nascimento, Hardware Security Evaluation Using Assurance Case Models, 10th International Conference on Availability, Reliability and Security (ARES), pp.193-198, 2015.
［Hawkins2013］Richard Hawkins, Ibrahim Habli, Tim Kelly and John McDermid, Assurance cases and prescriptive software safety certification: A comparative study, Safety Science 59, pp.55-71, 2013.
［Holloway2015］C. Michael Holloway, Explicate '78: Uncovering the Implicit Assurance Case in DO-178C, 23rd Safety-critical Systems Symposium, pp.2-5, Bristol, UK, 2015.
［RTCA2011］RTCA, Software Considerations in Airborne Systems and Equipment Certification DO–178C, 2011.
［Kakimoto2016］柿本 和希, ゴール構造化記法を用いた汎用的な安全要求の明確化, 奈良先端科学技術大学院大学　情報科学研究科　修士論文, 2016.
［Torano2009］虎野 吉彦, 小鑓 幸雄, 佐々木 宏, 鈴木 裕介, 植松 洋彦, 深津 敦, 山中 浩二, 麻生 大, 宇宙ステーション補給機 (HTV) 技術実証機の飛行結果, 平成21年度宇宙環境利用の展望, 第7章, pp.1-28, 2009.
［Shirasaka2011］白坂成功, 堀田成紀, 蒲原信治, 階層化 FDIR による高安全性航法誘導制御系の提案と宇宙ステーション補給機「こうのとり」での実現, 計測自動制御学会産業論文集 Vol.10, No.11, pp.91-99, 2011.
［Bando2011］坂東 祐司, ソフトウェアレビューにおける構造化チェックリストの表記方法の実験的評価, 奈良先端科学技術大学院大学　情報科学研究科　修士論文, 2011.
［Denger2007］C. Denger and F. Shull, A practical Approach for Quality-Driven Inspections, IEEE Software, Vol.24, Issue 2, pp.79-86, 2007.

つながる世界におけるセキュリティ

SECソフトウェアグループリーダー　中尾 昌善　　SEC研究員　宮原 真次

1 はじめに

IoT（Internet of Things）時代には、自動車、家電、ウェアラブル機器、ロボットなど様々な「モノ（things）」がネットワークに接続され、新しいサービスの創出や制御の高度化が期待される。このようなIoT時代のことを、我々は別名で「つながる世界」と呼んでいる。つながる世界では、ますます利便性が高まる一方で、これまで閉じた範囲でしか通信を行っていなかった製品やシステムに、外部からもアクセス可能な通信の入り口が付加されるため、そこからセキュリティ上の攻撃が発生したり、悪影響が他のモノに波及するなどのリスクが懸念される。

そこでIPA/SECは、つながる世界のリスクに対応するために、以下のような分野横断的に活用できる「つながる世界の開発指針」を取りまとめ、2016年３月に公開した。

安全・安心なIoTを実現するために、IoT製品やシステムの開発者が開発時に考慮すべきリスクや対策を17の指針として明確化

http://www.ipa.go.jp/sec/reports/20160324.html

前述の開発指針は、セーフティやリライアビリティについても考慮したものとなっているが、本稿ではとくにセキュリティに着目して、その考え方を解説する。

2 つながる世界とは

つながる世界、あるいはIoTの捉え方はまちまちである。IPA/SECで捉えているつながる世界のイメージを、図１に示す。図の横軸は、異なる分野の製品やシステムがつながって新しいサービスを創出することを表している。例えば、屋外からスマホで屋内の家電を制御するというサービスが考えられる。一方縦軸は、実製品や実システムから得られたデータに対して、ビッグデータ解析したり、AI（人工知能）制御を用いることにより、実製品や実システムの動きにフィードバックすることを表している。CPS（Cyber Physical System）と呼ばれることもある。例えば、電車の車両にセンサを装着し、電車の運行中に線路の保全データを収集し、劣化の予兆解析を行うことで、線路の保全をタイムリーに行うことが考えられる。

このような図１の横軸と縦軸の組み合わせによって形成されるのが、つながる世界であると捉えている。

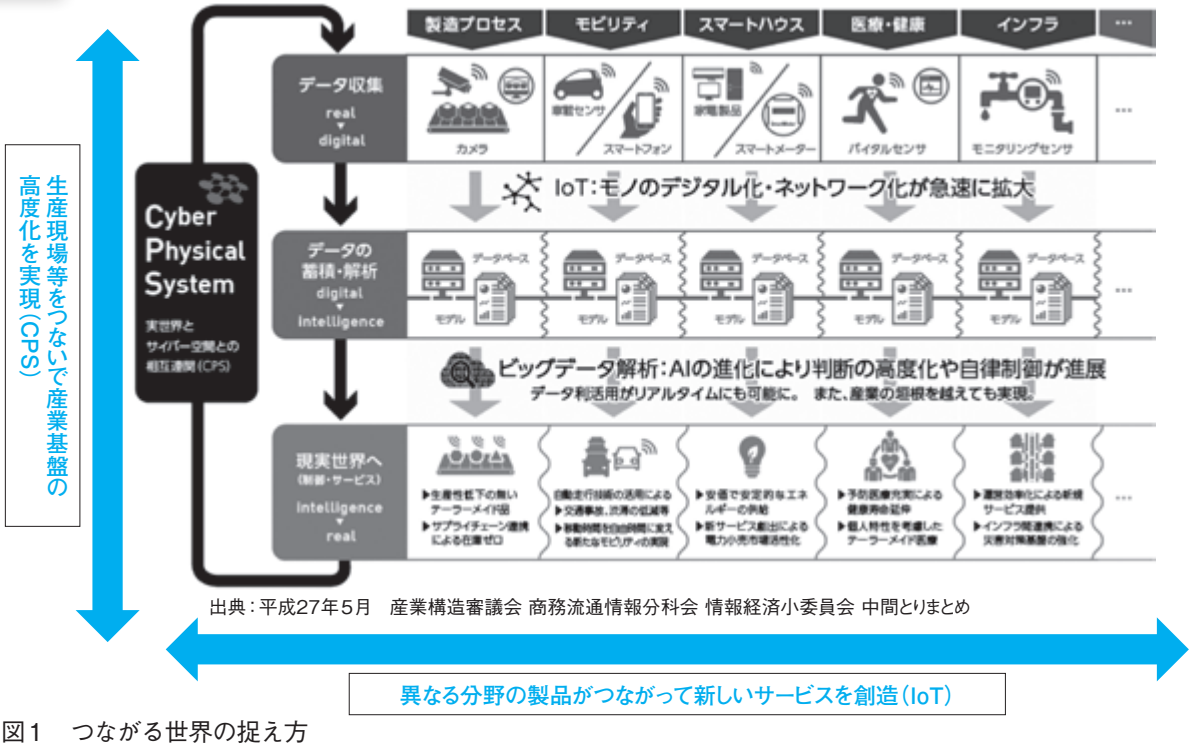


図１　つながる世界の捉え方

3 つながる世界のセキュリティリスク

3.1 つながる世界の特性

つながる世界のセキュリティリスクを考える前に、つながる世界の特性を考えてみたい。経済産業省と総務省が主導して設立した民間団体であるIoT推進コンソーシアムが、「IoTセキュリティガイドライン」※1を2016年７月に発行している。そこでは、IoTの特性は次のように整理されている。

① 脅威の影響範囲・影響度合いが大きいこと
② IoT機器のライフサイクルが長いこと
③ IoT機器に対する監視が行き届きにくいこと
④ IoT機器側とネットワーク側の環境や特性の相互理解が不十分であること
⑤ IoT機器の機能・性能が限られていること
⑥ 開発者が想定していなかった接続が行われる可能性があること

3.2 つながる世界のセキュリティリスクの特徴

つながる世界では、IoT機器やシステム同士がつながることにより、接続点から第三者に侵入され攻撃されるリスクが想定される。その対策を考える際には、次のようなつながる世界特有の課題、あるいはそこで顕著になると想定される課題を認識しておく必要がある。

(1) 想定しないつながりが発生する

IoT機器やシステムの開発者は、それがどういう場面でどのような使われ方をするかを想定し、設計条件を決める。しかし、IoT機器やシステムは、想定しない使われ方や接続が行われる可能性がある。例えば、工場、医療機関、家庭内のようなクローズドな環境でしか利用されないと想定していたIoT機器が、オープンな環境で利用されてしまうケースも出てくるであろう。更に、ユーザが興味本位でつなげてしまうケースがあるかもしれない。その結果、メーカが想定しないつながりによるリスクが高まる危険性がある。

(2) 管理されていないモノもつながる

IoT機器は、絶えずメーカや利用者の監視下にあるとは言えない。例えば、落とし物のスマホ、駐車場に放置された自動車、廃棄されたIoT機器などである。これらのIoT機器を悪意を持った第三者が手にすると、不正なソフトウェアを埋め込んだり、データを盗み出したりすることも可能である（図２）。

図２　物理的に管理されないIoT機器のイメージ

(3) 身体や財産への危害が波及する

自動車、家電、ヘルスケア製品、金融端末のようなIoT機器は、物理的な動作を伴うため、身体や財産への危害を与える危険性がある。被害が発生したとき、単体であれば範囲も限定的であるが、つながる世界では被害が波及し、より深刻な問題を招く可能性があると言っても過言ではない。

(4) 問題が発生してもユーザにはわかりにくい

故障や破損など物理的な異常は分かりやすいが、ウイルス感染や無線経由での不正アクセスなど、つながりに起因するセキュリティ上の問題は目に見えないため、利用者が気づかない可能性が高い。これは、もちろん既存のコンピュータシステムでも起こる事象であるが、IoT機器はアドレス管理などが弱いため原因を見つけにくく、より潜伏しやすいリスクとして捉える必要がある。

4 「つながる世界の開発指針」作成の背景

4.1 当初の課題認識

例えば、スマホで車を自動駐車することを考える。

図３に示すように、車は人の命を預かるレベルの厳しい設計条件で開発されており、一方スマホは通信やエンターテイメントで利用されることを想定した設計条件になっている。セキュリティの設計条件もおのずとこれに準じたものになっており、両者には違いがあるものと想定される。両者をつないだ製品やサービスを開発する際には、それぞれがつなぐ相手の設計条件を考慮した上で、対策を考える必要がある。このような課題認識のもとに、設計時に考慮すべき事項を指針としてまとめたのが「つながる世界の開発指針」である。

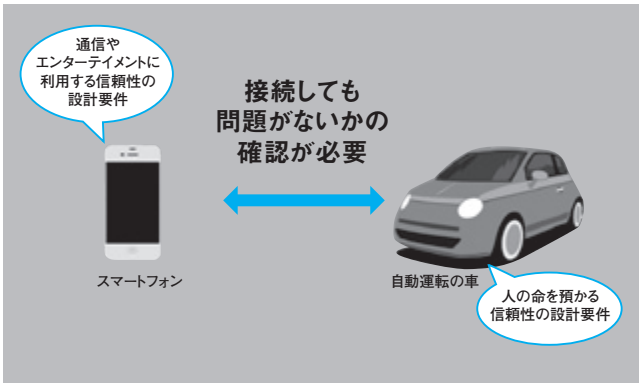


図３　異なる製品接続時の課題認識

4.2 課題認識に関する調査結果

つながる世界における課題認識を、セミナーなどの参加者にアンケート調査した結果を図４に示す。接続相手の信頼性が不明であることが、最も懸念される事項として指摘されており、上述の課題認識と一致している。

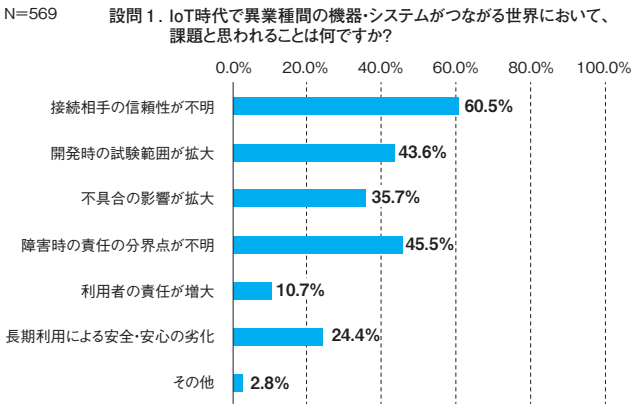


図４　IoT時代に向けての課題認識アンケート結果

5　つながる世界の開発指針

5.1　開発指針の導出方針

開発指針の策定においては、広く知見を集めるために学術研究者及び自動車、家電、住宅、ATM、産業機械など多様な産業の識者からなる「つながる世界の開発指針検討WG」を立ち上げ、WGメンバーのコンセンサスをとりながら検討を進めた。また、過去に発行した「つながる世界のソフトウェア品質ガイド」、「つながる世界のセーフティ &セキュリティ設計入門」など※2の作成において得られたセキュリティとセーフティの関係の整理などの知見も活用した。

(1)「IoTコンポーネント」に着目

IoTは、あらゆるモノがネットワークにつながり、新しい価値を生むが、新たなリスクの発生が懸念される。また、IoT同士がつながって拡大していく性質を有するため、IoTのシステム構成が刻々、変化し、リスク分析が難しいという課題がある。そこで、本開発指針では、IoTを「System of Systems（SoS）」と捉え、IoTを構成する機器やシステムのうち単独で目的や機能を果たす「IoTコンポーネント」に着目した。この「IoTコンポーネント」のリスクを想定し、対策を検討することで、IoTの安全・安心を実現することが可能と考えた。

(2) IoTコンポーネントのリスク分析

IoTコンポーネントのリスクを分析するために、IoTコンポーネントをモノ本来の機能や情報にIoT機能（通信機能など）を付加したものと仮定してモデル化し、守るべきものとリスク箇所を整理した。図５に示す。

また、IoTコンポーネントのつながりに着目し、誰がどのようにIoTコンポーネントをつなぐのかを洗い出し、つながり方のパターンを整理した。図６に示す。

IoTにおけるリスクの分析手法は、まだ、確立した手法がないため、我々は、上記の「守るべきものとリスク箇所」及び「つながり方のパターン」を横軸、IoTに関するリスク事例を縦軸としてリスク分析を行った。（詳細は本開発指針の付録Aを参照いただきたい）。

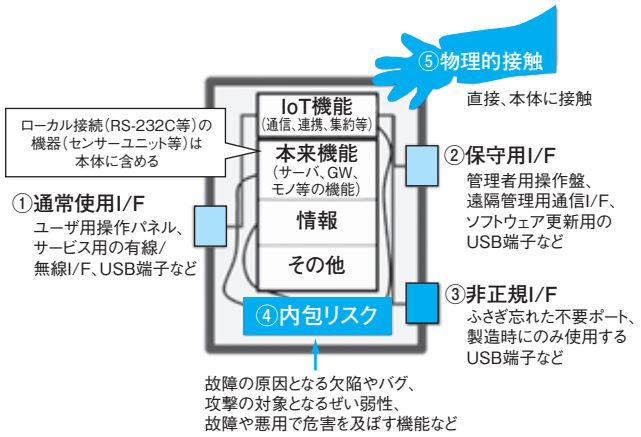


図５　IoTコンポーネントの守るべきものとリスク箇所

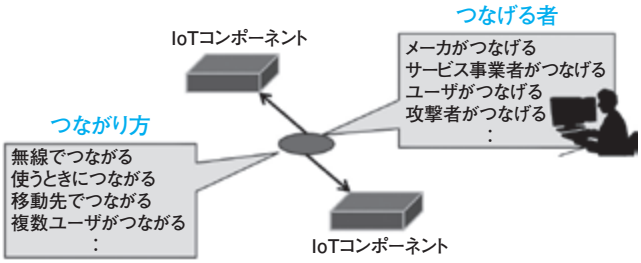


図６　IoTコンポーネントのつながりの捉え方（イメージ）

(3) 開発指針の導出

今回のリスク分析では、横軸の事項を６W1Hの視点で、誰がどのようにつないで、どこの箇所でどのような問題が発生したか。また、どの段階で発生したか、結果的にどのような被害になり、原因は何だったか、について分析した。この分析をもとに、発生要因を分類し、そこから本質的な課題を抽出し、IoT機器やシステムのライフサイクルの観点で整理することで、開発時に考慮すべき指針を導出した。

5.2　開発指針の概要

開発指針をまとめるにあたり、以下を考慮した。

- 分野横断的に活用できる抽象度の高い表現を採用
- IoTは利用期間が長いのでライフサイクルを意識
- 指針を具体的に検討するためのポイントを明示

開発指針は、表１の通り、ライフサイクルに合わせて、「方針」「分析」「設計」「保守」及び「運用」の５つのフェーズに分類し、17の指針としてまとめた。各指針は、指針／ポイント／解説／対策例により構成されている。実際の開発においては、ハードウェアの性能、開発コストなどの制約により各指針で例示した対策を実装できないケースも想定されるが、少なくとも各指針のポイントは、必ず検討していただきたいと考えている。

本開発指針は開発者を主たる対象としているが、「方針」に含まれる３つの指針はメーカ等の経営者にIoTのリスクに気づいていただくために有用である。また、「保守」「運用」に含まれる５つの指針は、開発者と保守者が連携してIoTコンポーネントの安全・安心を実現するために活用していただきたいと考えている。

表1　検討して欲しい開発指針一覧

大項目		指針
方針	つながる世界の安全安心に企業として取り組む	指針１　安全安心の基本方針を策定する
		指針２　安全安心のための体制・人材を見直す
		指針３　内部不正やミスに備える
分析	つながる世界のリスクを認識する	指針４　守るべきものを特定する
		指針５　つながることによるリスクを想定する
		指針６　つながりで波及するリスクを想定する
		指針７　物理的なリスクを認識する
設計	守るべきものを守る設計を考える	指針８　個々でも全体でも守れる設計をする
		指針９　つながる相手に迷惑をかけない設計をする
		指針10　安全安心を実現する設計の整合性をとる
		指針11　不特定の相手とつなげられても安全安心を確保できる設計をする
		指針12　安全安心を実現する設計の検証・評価を行う
保守	市場に出た後も守る設計を考える	指針13　自身がどのような状態かを把握し、記録する機能を設ける
		指針14　時間が経っても安全安心を維持する機能を設ける
運用	関係者と一緒に守る	指針15　出荷後もIoTリスクを把握し、情報発信する
		指針16　出荷後の関係事業者に守ってもらいたいことを伝える
		指針17　つながることによるリスクを一般利用者に知ってもらう

5.3　指針の例の解説

以下では特徴的な２つの指針について、その意図を説明する。

［指針５］つながることによるリスクを想定する

今までネットワークにつながっていなかった家電や生活機器などがつながるIoTの世界では、つながることによるリスクを想定することが大変重要になる。また、クローズドなネットワークでつながっていた機器やシステムが広域ネットワークにつながる場合も、想定していないことが起こる可能性があり、リスク対策が必要である。

具体例としては、2013年にプリンター複合機でパスワード未設定のまま、直接インターネットにつないだことによるデータ漏えいの危険性が指摘された。また、2016年にインターネットで閲覧可能となっている監視カメラが国内だけでも6000台あることがニュースとなった。

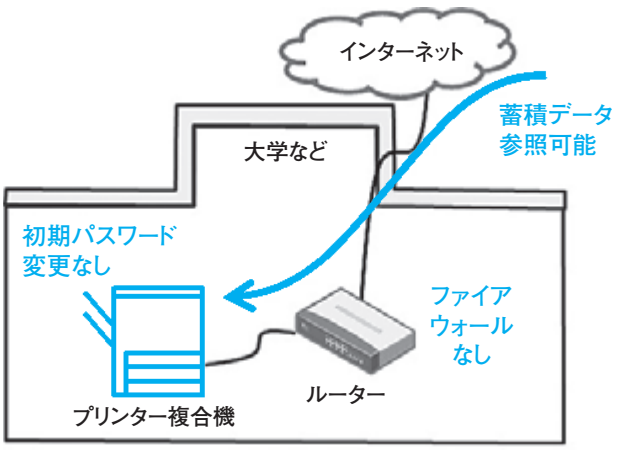


図7　インターネット接続によるリスクのイメージ

このような問題を防ぐには、利用される状況や環境などあらゆる場面におけるインターネット接続によるリスクを想定して、IoT機器・システムを開発すると共に、その危険性や対策機能について、利用者や設置事業者などに伝えることが重要となる。

［指針14］時間が経っても安全安心を維持する機能を設ける

IoTの特徴の一つに、ライフサイクルが長いことが挙げられる。10年以上にわたり利用されることを想定し、出荷後も安全・安心を維持する仕組みが求められる。とくに、製品サービスの重大な欠陥や脆弱性問題が起きたときは、速やかにかつ、長期にわたり改修するための機能が必要となる。また、屋外や人手が届かない場所に設置されるIoT機器などは、遠隔で改修できるリモートアップデートなどの仕組みが必須となる。

図8　経年で増大するリスク

一方で、このリモートアップデートの機能を具備することによるリスクの想定も必要である。アップデート実行中にIoTコンポーネントの性能が劣化することや多数のIoTコンポーネントの同時アップデートによるネットワーク帯域不足の影響など、運用を考慮する必要がある。

6　おわりに

本開発指針の内容は、IoT推進コンソーシアムが策定している「IoTセキュリティガイドライン」に採用された。現在、IoTにかかわる企業や業界団体などに対して、開発指針の普及活動を行っている。具体的には、開発現場で活用可能なチェックリストの整備や特定分野でのセキュリティガイドライン作成などの支援活動を行っている。また、今後は、開発指針の拡充や国際標準化に向けて、海外の関連団体との協調も進めていく。

脚注

※1　http://www.meti.go.jp/press/2016/07/20160705002/20160705002.html
※2　http://www.ipa.go.jp/sec/our_activities/iot.html

セキュリティ・バイ・デザインとアシュアランスケース

SEC研究員　金子 朋子

モノのインターネット（IoT: Internet of Things）と言われるIoTシステムは今後急激な普及・拡大が見込まれる。しかし、つながる世界は様々なリスクも抱えており、開発プロセスの早い段階から将来のハザードや脅威に備えていくことが必要とされている。またIoTシステムではつながる対象の広がりに応じて、要件はより複雑化するため、その可視化はとくに重要な課題である。そこで本稿では安全なIoT システムの枠組み作りのために開発プロセスの早い段階からセキュリティに対処する「セキュリティ・バイ・デザイン」の考え方を説明した上で、セーフティとセキュリティの要件すり合わせの意義とその手段の１つとしての「アシュアランスケース」について解説する。

1　セキュリティ・バイ・デザインとは？

1.1　現在のIoT開発の課題

現代のシステムはネットワークを介して様々な機器やクラウドと連携しながら動作している。このように異なる分野の製品や産業機械などがつながって新しいサービスを創造するモノのインターネットは新産業革命とまで言われ、大きな期待を集めている。IoTは家電、自動車、各種インフラ業者など新規プレーヤーの登場を産み、その取り込みは加速化している。しかし相互につながる際に最も懸念されるのは、IoTシステムへのセキュリティ上の脅威である。IoTシステムにおいても攻撃者はシステムの脆弱性を突いて攻撃を仕掛けてくるためである。その課題解決策として「セキュリティ・バイ・デザイン」という考え方が近年提唱されている。

1.2　セキュリティ・バイ・デザインの定義

内閣府サイバーセキュリティセンター（NISC）によるとセキュリティ・バイ・デザインの定義は、「情報セキュリティを企画・設計段階から確保するための方策」である[1]。（なお、本定義では「情報セキュリティ」としているが、「セキュリティ」とすればIoT製品やサービスにも適用できると考えられる。）

平成28年８月26日に発出された「安全なIoT システムのためのセキュリティに関する一般的枠組」においては、「将来、個々のシステムが相互に接続されることを見据え、システム相互間の接続が新たな脆弱性となる懸念があることを踏まえ、セキュリティ・バイ・デザイン（Security by Design）の思想で設計、構築、運用されることが不可欠」であるとセキュリティ・バイ・デザインの重要性が強調されている[2]。さらに「IoT システムの設計・構築・運用に際しては、セキュリティを事前に考慮するセキュリティ・バイ・デザインを基本原則とし、これが確保されていることが当該システムの稼働前に確認・検証できる仕組みが求められる。」と記述されており、安全なIoT システムのために、セキュリティ・バイ・デザインは基本原則として掲げられている。

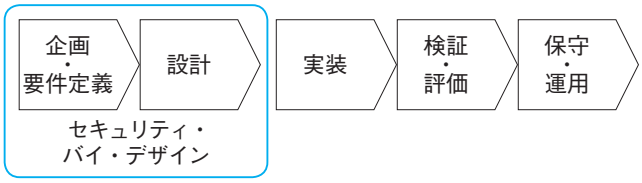


図１　セキュリティ・バイ・デザインの定義

1.3　セキュリティ・バイ・デザインのメリット

企画・設計段階という開発の早い段階からセキュリティを考慮することのメリットには、「手戻りがないため、納期を守れることや、コストも少なくできること」が考えられる。市場で運用されている段階で脆弱性が発見された場合には機器の交換やシステムの改修などが必要となるため、設計時のセキュリティ対策コストの100倍との試算もある（図２）[3]。また、他の機能ができあがってから後付けでセキュリティ対応するより、事前に対処したほうが「保守性の良いソフトウェアができること」もメリットとして挙げられる。

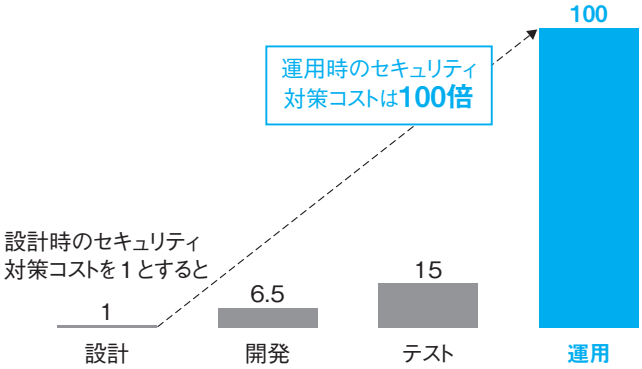


図２　開発工程別のセキュリティ対策コスト

1.4 セキュリティ・バイ・デザインの普及していない理由

それでは、なぜ、セキュリティ・バイ・デザインが普及していないのであろうか？セキュリティ・バイ・デザインが難しい理由としては以下の点が考えられる。

① セーフティ設計（設計段階で安全を作り込むこと）に比べ、セキュリティ設計（設計の段階で脆弱性の低減や脅威への対策を考慮にいれること）の歴史が浅く、上流工程の開発プロセスが定まっていない。

② 非機能要件なので、コンセプトを決める企画段階で考慮がされづらい。一般的な機能に対する要求はステークホルダ（利害関係者）の意図をシステムにより実現するのが目的であるが、セキュリティはステークホルダが実現を目的としてはいないが、当然、対応されていると考える非機能要件である。

実際、セキュリティ設計の基本方針を明文化している組織は多くはないのが現状である。「セーフティ設計・セキュリティ設計に関する実態調査結果」では、半数以上が明文化されたルールはないとしている[4]。同様の調査でセーフティに関しては自動車分野などでは明文化が進んでいることが明らかになっている。

1.5 セキュリティ開発プロセス

では上流工程の開発プロセスではセキュリティ設計として、具体的には何をすべきなのであろうか？一般的な脅威分析のアプローチは想定される脅威及び脆弱性を洗い出し、攻撃される可能性、攻撃された場合の想定被害からリスクを評価し、リスクの高い箇所にこれを抑止するための対策を検討する。具体的には①分析範囲の決定、②関係者の決定、③保護すべき資産の抽出、④前提条件の検討、⑤脅威の洗い出し、⑥対策方針の検討といった手順で検討される。

脅威分析の特徴は、セキュリティ特有の課題として、悪意の存在である攻撃者を仮定し、常に攻撃者がシステム関係者の意図しない動作をさせることを前提にリスク分析を行うことである。脅威分析は要求に基づく機能要件の分析に加えて攻撃者の存在を考慮した非機能要件の分析を必要とする。

攻撃とは、脅威を意図的に実現する手段であり、攻撃に対処できることの説明責任を果たすには、脅威分析が必要である。この脅威分析は従来の対応ではあまり実施されてこなかった施策であるが、今後のIoT時代の開発プロセスとして期待されている。

2 セーフティをまもれるセキュリティ

2.1 セーフティとセキュリティの違いと類似点

一般にセーフティとは偶発的なミス、故障などの悪意のない危険に対する安全を示すのに対し、セキュリティとは、悪意をもって行われる脅威に対しての安全を示し、セーフティとセキュリティは表1に示すように多くの違いをもっている。

表1　セーフティとセキュリティの相違点

相違点	セーフティ	セキュリティ
保護対象の違い	人命、財産（家屋等）など	情報の機密性、完全性、可用性など
原因の違い	合理的に予見可能な誤使用、機器の機能不全	意図した攻撃
被害検知の違い	事故として表れるため、検知しやすい	盗聴や侵入など、検知しにくい被害も多い
発生頻度	発生確率として扱うことができる	人の意図した攻撃のため確率的には扱えない
対策タイミング	設計時のリスク分析・対策で対応	時間経過により新たな攻撃手法が開発されるので、継続的な分析・対策が必要

両者は設計時に要件が相反することもあるので注意が必要となる。とくに生命、健康にかかわるセーフティの要件は重要である。

しかし、セーフティとセキュリティのリスク対応プロセスは類似している。セーフティではリスクの原因としてハザードを特定し、セキュリティでは脅威を特定するが、表現は異なるものの、リスクの特定、リスク分析、リスク評価、リスク対応というプロセスを繰り返すという基本的な流れは同様である。それ故、セキュリティ設計の脅威分析時に、セーフティ設計のハザードの特定と分析を行うことが可能であると考えられる。

2.2 セーフティ設計とセキュリティ設計すり合わせのメリット

IoT対応時にセーフティ設計とすり合わせをしながらセキュリティ設計を実施する手順について、考えてみよう。

インターネット冷蔵庫を新たに作成する例で考えてみると、まず冷蔵庫は既存の機能にインターネット機能を追加設計する必要が生じる。次にセキュリティを早期段階で考慮しない従来の対応ではセーフティ設計のみを実施する。そしてセーフティ設計を実施後のソフトウェア開発時にセキュアプログラミングをし、検証・評価時に脆弱性検査などを実施することになる（図3）。これに対して、セキュリティ・バイ・デザインの対応をする場合、セーフティ設計時にセキュリティ設計をすり合わせる。すり合わせをするとセーフティとセキュリティの部門間での作業の手戻りがないため、メリットが大きい。更に、IoT化することで変化する情報や通信方法の見直しだけでなく、セキュリティ上の脅威への対策を考えたインターネット冷蔵庫の企画、要件定義を実施する。事前にセキュリティ設計手順を踏むことで、セーフティとセキュリティ両方の観点からの安全性、コストなどのバランスのとれた設計を実施することが可能となる。

つまりセキュリティ・バイ・デザインはセキュリティだけのものではなく、事前にセキュリティを考慮することで、セキュリティ上の脅威にさらされるIoT機器などのセーフティも守ることができるのである。

セキュリティ・バイ・デザインはセキュリティのためだけの考え方に聞こえるが、実は「セキュリティで脅かされるセーフティも守ることができる」と言える。

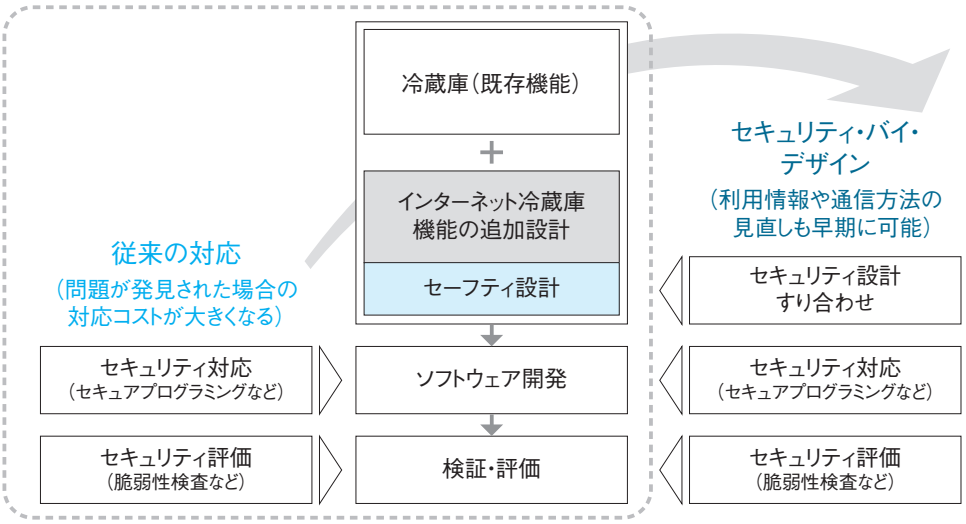


図３　IoT機器などのセキュリティ・バイ・デザインのイメージ（インターネット冷蔵庫の例）

3　アシュアランスケースとは？

3.1　ロジカルな設計品質の説明

セーフティとセキュリティを考慮した設計をしたとしても、「IoTでつながった製品を安全なものとして使って大丈夫か？」という利用者の不安に対して設計者は説明が求められる。

設計品質のロジカルな説明とは、「その設計によって目標が達成されることが、事実に基づき、論理的に説明されていること」である。

設計品質のロジカルな説明をするためには、アシュアランスケースの理論的背景となっているトゥールミン・ロジックが参考になる[5]。トゥールミン・ロジックは法律分野でイギリスの分析哲学者スティーブン・トゥールミン（Stephen Edelston Toulmin）が実社会の議論形態を分析して提唱した論証モデルである。

- 主張とは、論理として構築されるひとつの主張
- 基礎とは、論理の根拠となる、状態、事実など最初に呈示される説明情報
- 根拠とは、クレームの根拠としてデータが利用可能であることを正当化する情報

設計品質のロジカルな説明には、第三者でも分かりやすく、事実（証拠）に基づいて論理的に設計品質を説明できる「見える化」されたドキュメントが有用である。

3.2　アシュアランスケースの定義

アシュアランスケース（assurance case）とは、テスト結果や検証結果をエビデンスとしてそれらを根拠にシステムの安全性、信頼性を議論し、システム認証者や利用者などに保証する、あるいは確信させるためのドキュメントである[6]。アシュアランスは保証、ケースは論拠を意味している。

アシュアランスケースは欧米で普及しているセーフティケース[7]から始まっており、近年、安全性だけでなく、ディペンダビリティやセキュリティにも使われ始めている。アシュアランスケースはISO/IEC15026やOMGのARM[8]などで標準化が進められている。

アシュアランスケースの構造と内容に対する最低限の要求は、システムや製品の性質に対する主張（claim）、主張に対する系統的な議論（argumentation）、この議論を裏付ける証跡（evidence）、明示的な前提（explicit assumption）が含まれること、議論の途中で補助的な主張を用いることにより、最上位の主張に対して、証跡や前提を階層的に結び付けることができることである。

アシュアランスケースは対象となる機器やシステムについて、なぜその設計で目標が達成されるかを事実に基づき、論理的かつ第三者でも容易に理解できる表記で説明する手法である。

IoTの対象となる航空、鉄道、軍事、自動車、医療機器の分野の複数の安全性規格やガイドラインで要求され、欧州を中心に広く利用されている。

3.3　アシュアランスケースの表記法

「見える化」の手段としてGSN、CAE、D-Caseなどのアシュアランスケースの表記法がある（表２）。

表２　アシュアランスケースの表記法一覧

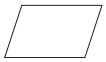
	CAE	GSN	D-Case
正式名称	Claim、Argument、Evidence	Goal Structuiring Notation	Dependability Case
登場時期	1998年	2011年	2012年
構成要素	３種類	６種類	GSNを拡張
開発組織	英Adelard社、ロンドン大学	英ヨーク大学	日本DEOSプロジェクト

代表的な表記方法は、欧州で約10年前から使用されているGSN[9]であり、要求を抽出した後の確認に用い、システムの安全性や正当性を確認することができる。日本国内ではGSNを拡張したD-CaseがJST CREST DEOSプロジェクトで開発されている[10]。要求、議論、証跡のみのシンプルなアシュアランスケースであるCAE[11]もある。

アシュアランスケースの代表的な表記法であるGSNの構成要素を表３に示す。

GSNでは前提とサブゴールに分かれる戦略の明示により論理関係を明確にした上で、各サブゴールが成り立つことで、最上位のゴールが成り立つことが保証される。

表３　GSNの構成要素

名称	図式要素	内容
主張（ゴール）		保証したいこと、命題（例：システムは安全である）ゴールは更に詳細なゴール（サブゴール）に分解される
説明（ストラテジ）		ゴールをサブゴールに分けるときの考え方（例：個別の障害ごとに議論する）
証拠（エビデンス）		ゴールが成り立つことを最終的に保証するもの（例：テスト結果、運用事例など）
前提（コンテクスト）		システムの状態、環境などゴールを議論するときの前提など（例：リスク分析の結果得られたハザードのリスト）
未定義要素		ゴールを保証するための十分な議論又はエビデンスがない（これはゴールやストラテジーにつけることができる）

3.4　アシュアランスケースによる脅威分析検討事例

本節ではIoTの具体的事例をもとにアシュアランスケースによるセキュリティ要件の可視化方法を示す。図４はスマートハウスの脅威と対策の検討例を図示したものである[12] HEMS コントローラを中心に接続されたHEMS対応機器やそれ以外のネットワーク対応機器がホームルータを介してインターネットに接続されており、外出先からスマートフォンを用いてクラウドサービス経由で家庭内の機器にアクセスすることによって、家庭内の機器の様子を監視したり、遠隔操作したりすることが可能となる。このシステムでは、スマートハウス内に設置された機器の一部に保存されたデータの漏えい、通信路上のデータの盗聴・改ざん、クラウドサービスやインターネット上に接続された中継機器への不正アクセス、（不正ログイン、その後の不正コマンド発行による許可なき遠隔操作）、クラウドサービスやインターネット上に接続された中継機器への DoS 攻撃、クラウドサービス上に保存されたデータの漏えいなどの脅威が想定される。

図４の事例をアシュアランスケースで記述したものが、図５である[13]。図５は「G_1 スマートハウスのセキュリティ設計は妥当である」というゴールを満たすために、「S_1 脅威分析の洗い出しと対策を示す」戦略を「G_2 スマートハウスの脅威の洗い出しは妥当である」と「G_3 スマートハウスの脅威に対する対策立案と選択は妥当である」の２つのゴールに分けて説明している。図５に示すG_2以下はスマートハウスにつながっている機器ごとと機器間の通信ごとに脅威を洗い出すことを求めている。各機器と機器間の通信の双方の脅威の出所をおさえれば、網羅的な脅威の洗い出しが可能になるからである。これらはG_4からG_11のゴールとして設定され、各ゴールで洗い出した脅威に対する対策をE_1からE_8の証跡として提示する。図４の事例に示された脅威の詳細が各証跡となる。

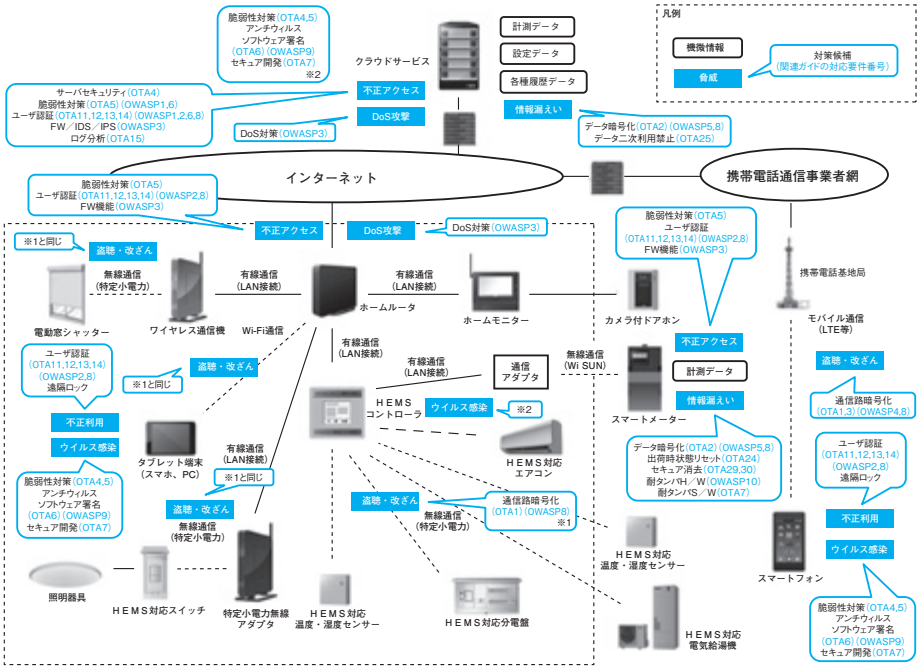


図４スマートハウスの脅威と対策の検討例

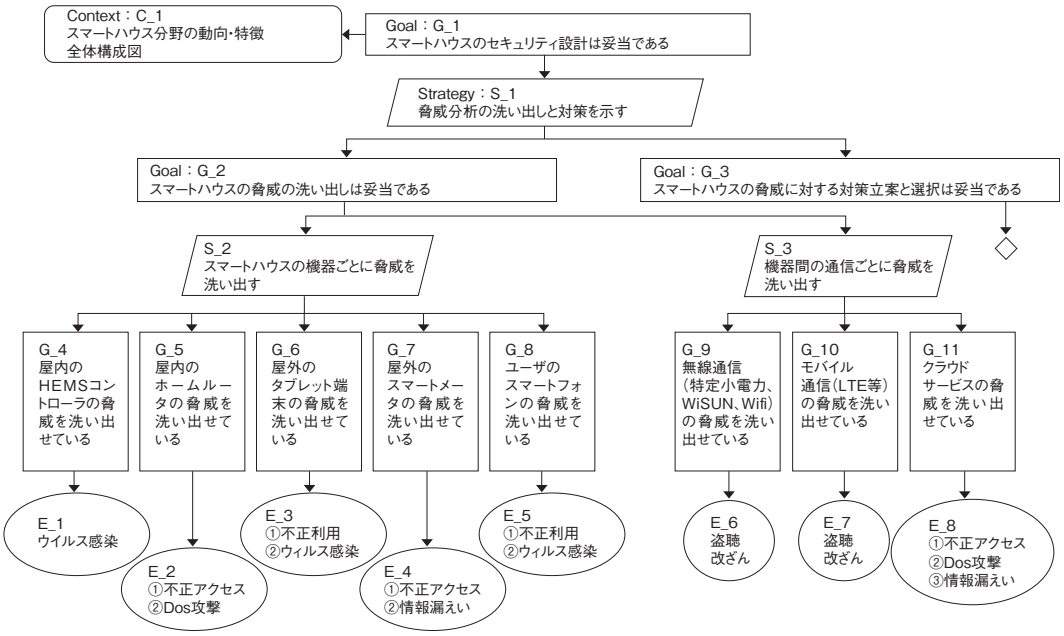


図５　スマートハウス事例へのアシュアランスケースの適用例（脅威の洗い出し部分）

図６に示すG_3以下は、洗い出した「S_4 対策ごとに分けて論証する」、「S_5 対策選択に合意する」、「S_6 残存リスクを影響分析する」という３つの戦略をプロセス化している。E_1からE_8で挙がった対策の中には、発生箇所が異なっていても対策として同じものが含まれるため、対策ごとに実施方法を証跡として示す。これらは脅威の洗い出しに対する重複の排除となる。また、実施する対策は経営層・顧客などのステークホルダとの合意が必要である。更にコストなどを考慮した実施可能な対策でなければ実施できない。そこで実施の合意を得られた対策は合意を証跡として残し、コストなどの事情で実施に至らなかった対策は影響分析をして残存リスクを証跡として示すことが必要である。これらは選択する対策と残存リスクに対処するプロセスとなる。なお、G_12からG_19のゴールが妥当である根拠としてE_9からE_16の証跡を示しているが、これらが実際に「妥当である」というためには、別の考察や判断基準が必要であろう。本提案の意義はハイレベルな脅威分析の妥当性提示である。本手法を実際に用いるためにはそのケースに応じた段階的詳細化が必要となる。本手法では脅威分析を実施したいケースをインプットとし脅威の洗い出しの結果、立てた対策がアウトプットになる、このアウトプットは運用による対処と設計による対処に分かれて実施される（図７）。設計者はアシュアランスケースを利用することで脅威対策の全体像を把握し設計できる。またトレーサビリティを保ちながら、修正、再利用をすることができる。

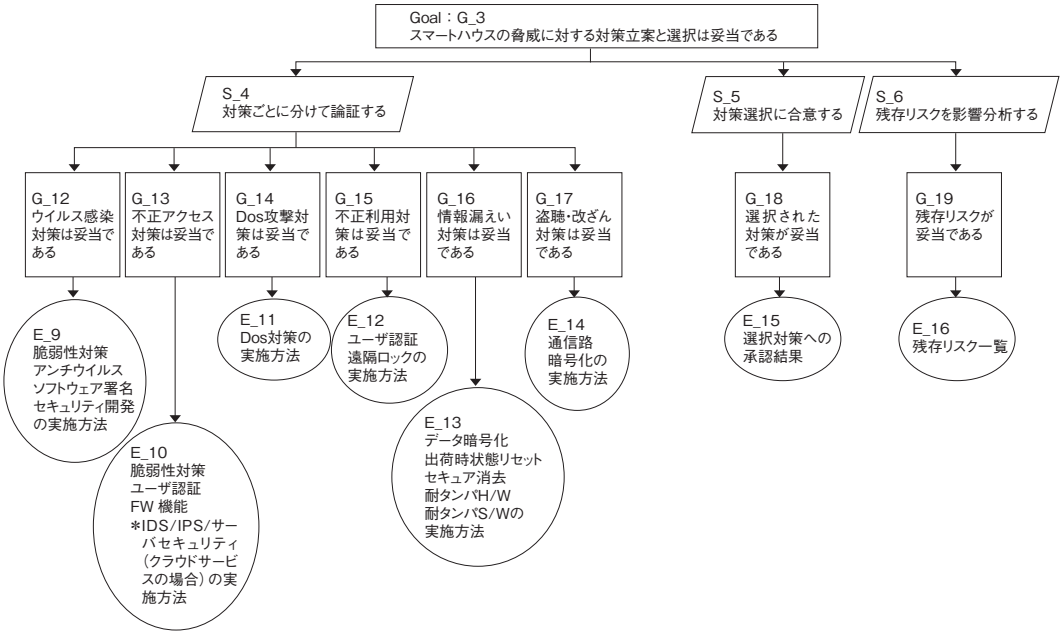


図６　スマートハウス事例へのアシュアランスケースの適用例（対策立案と選択部分）

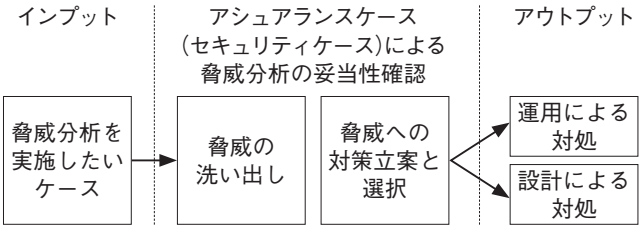


図7　脅威分析の妥当性確認

4　ステークホルダとの設計情報共有

セーフティとセキュリティの対応は企画、設計開発、販売・サポート、廃棄までライフサイクル全体において必要である。

さらに、図8のようにライフサイクルの各段階において関係する自社内の他の部門、特に品質管理部門、経営層、利用者といったステークホルダとの設計情報共有は設計品質の「見える化」のメリットの1つであり、アシュアランスケースは有効な手段となる。例えば次のような活用例が考えられる。

- セーフティとセキュリティの両部門において、設計内容を共有するためにそれぞれの部門で作成したアシュアランスケースを共有し「見える化」に利用できる。セーフティ設計とセキュリティ設計のすり合わせにも活用できる。
- ソフトウェア設計や再利用時の設計内容の理解において、新製品開発やバージョンアップ時のソフトウェア再利用時に、設計内容を理解するために活用できる。
- 設計者間での内容の理解だけでなく、経営層や品質管理部門等のステークホルダとの設計情報共有にも利用可能である。
- トレーサビリティ、説明責任のツールとして問題が発生したときに設計内容を確認したり、問題と設計との関係を説明するために活用できる。

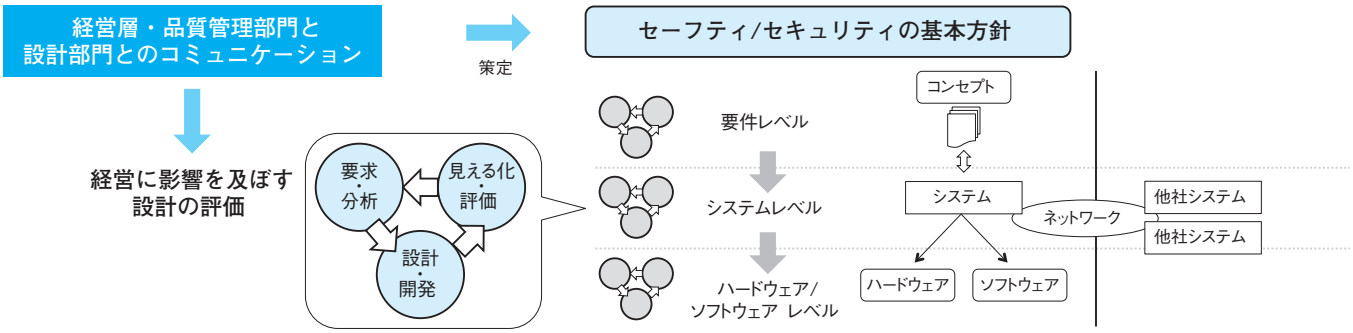


図8　ステークホルダとの設計情報共有

5　まとめ

本稿では、セキュリティ・バイ・デザインとは「情報セキュリティを企画・設計段階から確保するための方策」であり、開発の早期段階からセキュリティを考慮していくことの必要性を解説した。またつながる対象の増加によってセキュリティ上の脅威がセーフティを脅かす可能性の高まるIoTの開発において、セキュリティをセーフティと共に考慮していくことによりセーフティを守れるセキュリティとすべきであることを示した。

更に、設計品質のロジカルな説明とは、「その設計によって目標が達成されることが、事実に基づき、論理的に説明されていること」であり、その有効な手段の1つとしてアシュアランスケースが利用されていることを説明し、セキュリティ上の脅威分析にアシュアランスケースを用いた事例を示した。

IoT時代の設計は、ステークホルダとの設計情報共有がカギを握る。ここにアシュアランスケースの利用を通じて、設計品質のロジカルな「見える化」が普及していくことを念願している。

参照文献

[1] NISC, http://www.nisc.go.jp/conference/seisaku/dai15/pdf/15siryou02.pdf
[2] NISC, 安全なIoTシステムのためのセキュリティに関する一般的枠組, http://www.nisc.go.jp/active/kihon/res_iot_fw2016.html
[3] IPA, つながる世界のセーフティ＆セキュリティ設計入門
[4] IPA, セーフティ設計・セキュリティ設計に関する実態調査結果
[5] IPA, アシュアランスケース入門, 2015, http://www.ipa.go.jp/files/000043906.pdf
[6] 松野裕、高井利憲、山本修一郎「D-Case入門 ～ディペンダビリティ・ケースを書いてみよう！～」, 2012
[7] T P Kelly & J A McDermid, “Safety Case Construction and Reuse using Patterns”, in Proceedings of 16th International Conference on Computer Safety, Reliability and Security (SAFECOMP'97), Springer-Verlag, 1997
[8] OMG, ARM, http://www.omg.org/spec/ARM/1.0/Beta1/
[9] Tim Kelly and Rob Weaver, The Goal Structuring Notation – A Safety Argument Notation, Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, 2004
[10] 松野裕、山本修一郎：実践　D-Case ～ディペンダビリティケースを活用しよう！～, 株式会社アセットマネジメント、2013
[11] The Adelard Safety Case Development (ASCAD), Safety Case Structuring: Claims, Arguments and Evdence, http://www.adelard.com/services/SafetyCaseStructuring/index.html
[12] IPA, IoT開発におけるセキュリティ設計の手引き, 2016
[13] 金子朋子、髙橋雄志、勅使河原可海、田中英彦：CC-Caseを用いたIoTセキュリティ要件の可視化, 2016

組込みシステム セーフティ・セキュリティ 検討WGの取り組み

SEC調査役　石田 茂

1 はじめに

社会の重要インフラを担う組込み製品・システムでは、障害発生時に及ぼす人命並びに環境に与える影響の大きさから、ライフサイクル全般にわたる安全性を重視したものづくりが行われてきた。一方、急速に進む事業のグローバル化は国際機能安全規格認証などの新たな対応を要求しており、またネットワークによる相互接続の進展により日々増大し続けるセキュリティ脅威への対処といった新たな取り組みが必要不可欠となってきた。

例えば近年話題となっている自動車の自動運転などに代表されるような新しい機能、サービスの実現に際してはインターネットを介したシステム連携が欠かせないが、同時に車載システムのハッキングなど、セキュリティ上の脅威がセーフティに及ぼすリスクへの対応が危急の命題となっている。

2 セーフティとセキュリティの課題

従来のセキュリティ対策は情報システム部門が担い手となり、企業が内部で利用するITインフラへの対応を中心に進められてきたが、個別の組込み製品・システムのものづくりにおいてはどうなのか、開発の現場ではどのようなニーズがあるのか把握する必要があると感じ、2015年度より社会の重要インフラに使用されるセーフティな制御システムの開発製造を行っている国内メーカ並びにこうした事情に明るい大学等の総数20以上の組織よりヒアリングを行いつつ、業界を取巻く技術動向などの調査を行ってきた。この結果、

- セキュリティ対応の必要性は理解しているが、従来セーフティとセキュリティの接点はなく、開発・運用の組織やエンジニア同士のコミュニケーションも図られていない。

- セーフティに比べセキュリティの歴史は新しく、脅威の拡大スピードにその対応が追い付いていない。

などの状況にあることが分かってきた。（詳細は図1を参照）

	Safety	Security
動向・要件	プロセスなど確立しているが、IoT時代に向けた新たなサービスや機能への対応が必要になっている（自動運転、生産系と事務系システム連携など）	セキュリティ脅威は日々増加、変化しており、将来にわたる脅威の全体像をあらかじめ網羅することは不可能である
規格	IEC61508を親としたドメインごとの子規格が存在している	組込みシステムのセキュリティ規格は現在ドメインごとに作成中である
プロセス	ドメインごとに確立されたプロセスが定義されている	モデルとなるようなプロセスは現状未定義である
課題（国内企業、大学よりヒアリング）	事業の国際化を考えるとグローバルスタンダードへの適合が不可欠だが、Safetyの認証取得同様にSecurityでも多大なコスト、工数が必要となると負担は大きい	
	Safety, Securityの双方に詳しい技術者はおらず、Security要件がSafetyに及ぼす影響を同時に評価・すり合わせてゆくことが難しく、連携させる枠組みもない	
	Security要件の抽出において、脅威分析の具体的なやり方などが規格にも明示されていないため、人による差が大きくなり網羅性が十分であるかどうか判断できない	
	Security脅威を定量的に把握、評価することが難しく、また年々新たな脅威が発生するため対応の十分性がはっきりしない	

図1　セーフティ・セキュリティ状況

3 活動の狙いと進め方

これらはいずれも難しい課題でありそのすべてへの対処は容易ではないことを承知しつつ、IPA/SECでは変化する時代の要請に対応した活動が必要であると考え、セーフティとセキュリティが連携し双方の要件をすり合わせる枠組みを提示することを目的とした「組込みシステムセーフティ・セキュリティ検討WG」を設立した。この際、以下のような考え方をベースとした。

【活動方針】

- **セーフティファースト**

  セーフティとセキュリティの要件検討は、セーフティゴール（安全性、可用性などの確保）からセキュリティを考える。

- **グローバルスタンダードとの連動**

  プラント、鉄道、自動車など重要インフラの国際市場展開状況に照らし、ISO/IEC国際規格及び業界スタンダードの動向、日本を含む国際的な検討活動との連携性などを念頭に置く。（例. IEC/TC65/WG20※1活動）

- **本格検討に先立つ準備フェーズの設定**

  2017年からの本格検討に向けた準備段階として、2016年度はフレーム（検討のための叩き台）を検討する。

【進め方】

- **仮想システム**

  検討にあたってはターゲットシステム（検討のための仮想システム）を想定する。

- **国際規格準拠**

  機能安全はIEC61508※2、セキュリティはIEC62443※3を用い、上流プロセスを検討する。

- **実務者レベルの検討**

  国際機能安全、組込みシステムセキュリティ対応が特に重要となる分野の実務経験メンバーによる現場目線を意識した検討を行う。

4 今後の取り組み

前述の通り2016年度は本格検討に先立つ準備フェーズとして、フレームの作成を目標に数回の検討活動を行う予定である。

脚注

※1　IEC/TC 65 Industrial-process measurement, control and automationの中で安全とセキュリティのフレームワークをWG20で検討中。
※2　IECが制定した基本安全規格であり、電気・電子・プログラマブル電子にかかわる国際規格。
※3　IECが制定を進めている制御システムにおけるセキュリティに関する国際標準規格。

最先端ICT技術の実証プラットフォーム
～ NICT総合テストベッド～

国立研究開発法人　情報通信研究機構　総合テストベッド研究開発推進センター
テストベッド連携企画室長　**水落 祐二**

ICT分野における国際競争が激しくなる中、研究開発から社会実装までの加速化を図ることが更に重要となってきている。そのため、基礎研究段階の研究開発においても、研究開発と並行して技術の検証を実施することにより、研究開発成果を早期に実用段階へ橋渡しすることのできる環境が求められている。また、研究開発成果の検証においては、技術実証だけでなく社会実証にも一体的に取り組むことで、実用化へのスピードを加速することができると考えられる。
国立研究開発法人情報通信研究機構（NICT）では、こうした実証環境を整備するため、IoT分野を含む多様なICT技術に関する技術実証が可能であると共に、社会実証にも活用することが可能な実証プラットフォーム「NICT総合テストベッド」を提供している。NICTは、NICT総合テストベッドの提供を通じて、産学官の利用者による多様な実証ニーズに応えることにより、ICT分野における研究開発成果の最大化を図り、オープンイノベーションの創出に寄与していく。
また、NICT総合テストベッドは、セキュリティ分野においても様々な検証に利用が可能である。これまでにも、テストベッドの仮想ネットワーク上に構築した隔離環境を利用して、幾つか特徴のあるセキュリティ検証実験が行われてきている。これらの事例についても併せて紹介する。

NICT総合テストベッドの概要

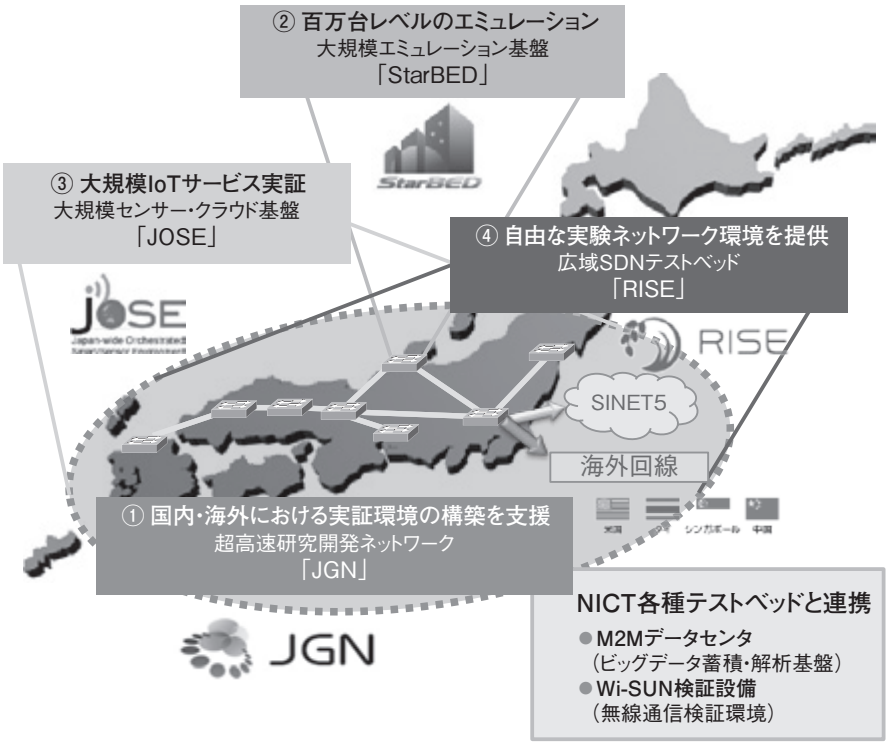


図1　NICT総合テストベッドの概要

NICT総合テストベッド研究開発推進センターでは、IoT技術など最先端のICT技術に関する各種実証を支援するため、NICTが運営する様々なテストベッドを連携させた「NICT総合テストベッド」を構築し、産学官の研究開発機関等に提供している。

NICT総合テストベッドは、超高速研究開発ネットワーク（JGN）、大規模エミュレーション基盤（StarBED）、大規模センサー・クラウド基盤（JOSE）、広域SDNテストベッド（RISE）の４種類のテストベッドから構成され、それぞれを自由に組み合わせて利用することが可能である。また、ビッグデータ蓄積・解析基盤（M2Mデータセンタ）、無線通信検証環境（Wi-SUN検証設備）等のNICT各種テストベッドとの連携利用も可能である。

超高速研究開発ネットワークJGN

超高速研究開発ネットワーク（JGN）は、最先端のネットワーク技術に関する検証を可能にすると共に、総合テストベッドの実証環境構築の支援ネットワークとしても活用可能なテストベッドである。

JGNは、国内、海外のアクセスポイントを最大100Gbpsの広帯域な回線で接続し、セキュアな広域L2接続サービスを提供している。また、仮想化サービス（仮想マシン、仮想ストレージ）による柔軟な開発環境を併せて提供している。これらにより、高速かつ広域なネットワーク環境を用いた次世代バックボーン・ネットワーク技術の検証などを可能にすると共に、StarBED、JOSE、RISEなど、他のテストベッドを用いた実証におけるネットワーク環境としても活用が可能である。

更に、JGNは、国立情報学研究所（NII）が提供する学術情報ネットワーク（SINET5）や各地方の地域情報ハイウェイ等と相互接続している。そのため、これらのネットワーク経由でJGNを利用することが可能であると共に、遠隔地の大学や研究機関等と共同して研究することも可能である。

JGNを利用したセキュリティ検証の事例としては、技術研究組合制御システムセキュリティセンター（CSSC）により、重要インフラシステムにおけるサイバーインシデント発生時に、緊急時における現地への移動という時間的ロスを伴わないよう、インシデントを安全に隔離しながら、制御システム・制御機器に対して遠隔からフォレンジック作業を行うための技術の検証が行われている。これは、JGN上に仮想ネットワークを構築し、この仮想ネットワーク上でフォレンジックを実施することで、通常トラフィックに悪影響を与えずにフォレンジックを実施するための技術の検証・開発を実施したものである。

【様々なネットワークと連携した利用が可能に】
JGNの持つ全国規模の**アクセスポイント（AP）**に加えて、**連携するネットワークの接続拠点**を利用することにより、全国各都道府県からの利用が可能。

これまでより一層、ご利用しやすくなりました。

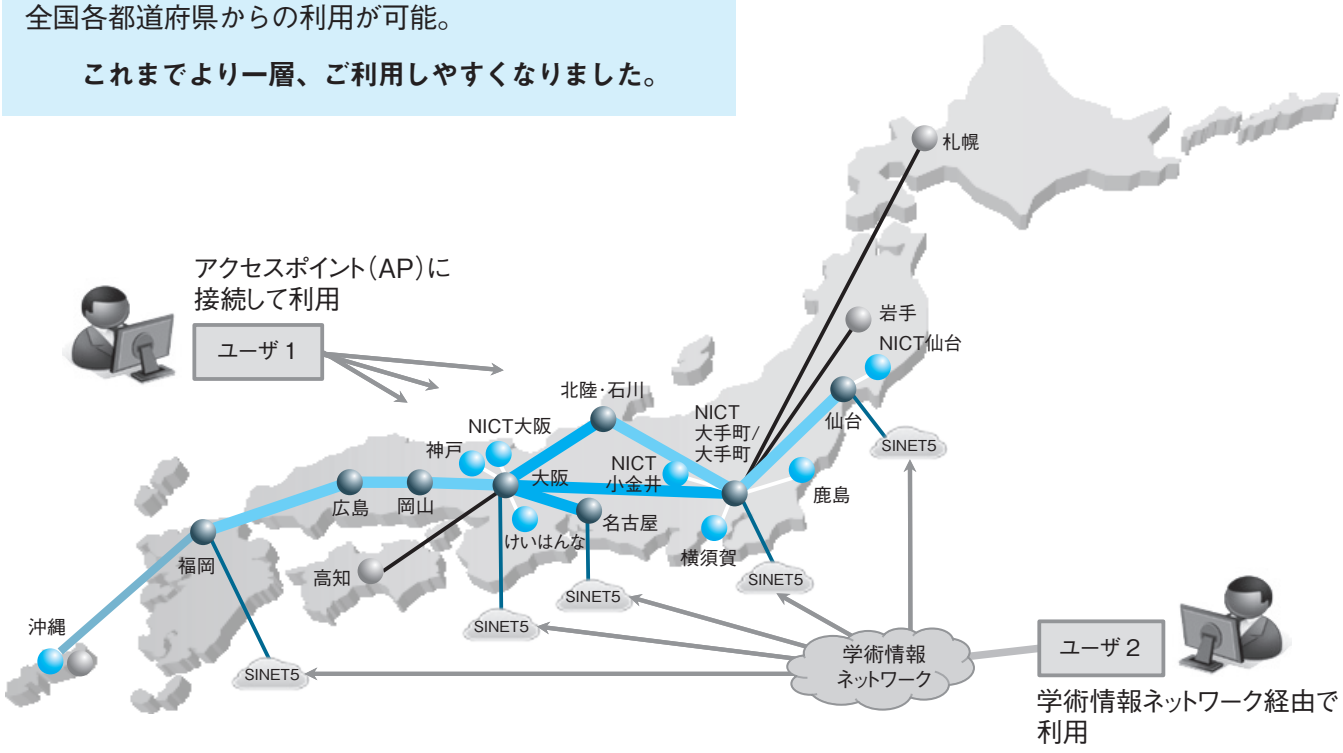


図２　JGNの概要

大規模エミュレーション基盤 StarBED

大規模エミュレーション基盤（StarBED）は、千台以上のPCサーバーを自在に組み合わせることにより、実際のハードウェア・ソフトウェアの動作を仮想環境で検証することのできるテストベッドである。

StarBEDにおいては、仮想ネットワーク（VLAN）により任意のトポロジを構成すると共に、ターゲットシステムを実際に動作させて検証することができるため、実環境に近い条件での実証が可能となる。また、大規模な実証を短時間・低コストで行うことができると共に、実用ネットワークから隔離された環境で実証を行うことができる。

現在、すべての人、すべてのモノがネットワークに接続されるIoT時代に向けた検証基盤を構築するため、PCだけではなく携帯電話やセンサーなど常に身近にあるデバイスが動作する基盤と、それらをつなぐ温度場や電磁場までも検証環境に取り入れるための技術開発を行っている。また、様々な無線設備・ネットワークの実証環境としても活用可能となるよう機能の拡充が進められている。

StarBED上には外界から強固に隔離した環境を構築できるため、マルウェアなどを実際に動作させるようなセキュリティに関する実験を実施することができる。また、セキュリティ人材育成の重要性が叫ばれている中、座学だけではなく実際に手を動かしてセキュリティ技術を習得するための演習環境の構築にStarBEDが活用されている。enPiT Securityでは大学の授業の一環として、StarBED上に構築された演習環境に学生がログインし、マルウェアの挙動の観察や、感染した端末への対応などを学んでいる。また、Hardening Project※1が実施する競技会では、一般から公募で選ばれた参加者がチームを組み、脆弱性のある電子取引サイトを運営チームからの攻撃から守り、その売り上げを競うイベントとして継続的に実施されている。

図3　StarBEDの概要

脚注

※1　Hardening Project：「守る技術」の価値を最大化することを目指すセキュリティ・イベント

大規模センサー・クラウド基盤 JOSE

大規模センサー・クラウド基盤（JOSE：Japan-wide Orchestrated Smart/Sensor Environment）は、IoT関連技術の検証やIoTサービスのフィールド実証サポートを目的に、IoTサービスに必要な設備をSDI管理機能によって柔軟かつ迅速に提供するテストベッドである。

全国各フィールドに設置されたセンサネットワーク設備と共に、1,200ホスト：20,000VM規模のクラウドネットワーク設備を具備し、ネットワーク接続及びセキュリティ設定が完了済みの汎用Linux OSを起動された状態で提供することにより、大規模数の分散クラウド・センサによるIoTテストベッドを迅速に構築することを可能としている。

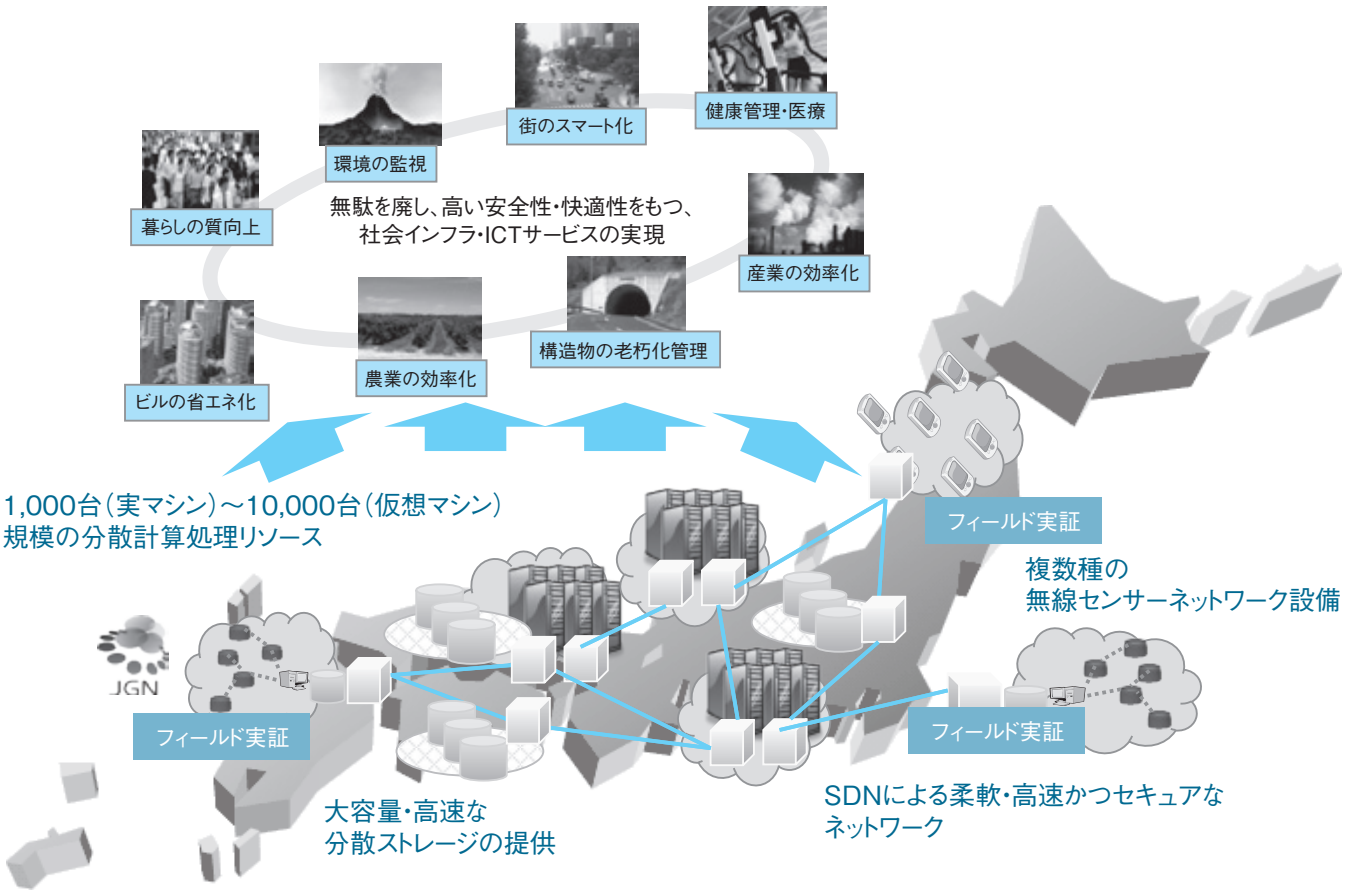


図４　JOSEの概要

おわりに

NICTでは、スマートIoT推進フォーラムのテストベッド分科会における議論や各利用者の皆様からの御意見を元に、ニーズに即したテストベッドの構築を目指していく。また、利便性の向上やコンサルティングの充実などにより、更なる利用促進を図っていく。NICT総合テストベッドに関する御意見、御要望などございましたら、筆者または総合テストベッド事務局（tb-info@jgn-x.jp）まで御連絡いただけると幸いである。

NICT総合テストベッドの御利用に関するお申込み、お問い合わせについては、総合テストベッド事務局（tb-info@jgn-x.jp）までご連絡をお願いいたしたい。また、NICT総合テストベッドの御利用方法、最新情報などについては、HP（http://testbed.nict.go.jp/）にも掲載している。ぜひとも御参照いただきたい。

つながる世界の検証

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

1 はじめに

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

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

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

2 つながる世界のシステムとは

2.1 検証対象とするつながる世界

つながる世界について、現状では以下のキーワードで表現されることが多い。

① IoT（Internet of Things）

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

② CPS（Cyber Physical System）

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

③ SoS（System of Systems）

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

本稿で検証対象とするつながる世界は、つながる経路やつながるもの（製品やシステムなど）を特定せず、上記①と②を包含した③のSoS（System of Systems）を前提に検討を進めるものとする。（以降「つながるシステム」と呼ぶ）

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

A．出荷済のもの同士をつなげて検証

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

B．出荷済のものと、開発中のものをつなげて検証

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

C．開発中のもの同士をつなげて検証

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

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

2.2 システムへの要求の変化

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

● 人の要求を実行するシステム

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

● 人に代わり要求するシステム

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

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

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

2.3 人の要求を検証する

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

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

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

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

2.4 つながるアーキテクチャ

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

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

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

つながるシステムでは、この①～③の把握が困難になりやすい。

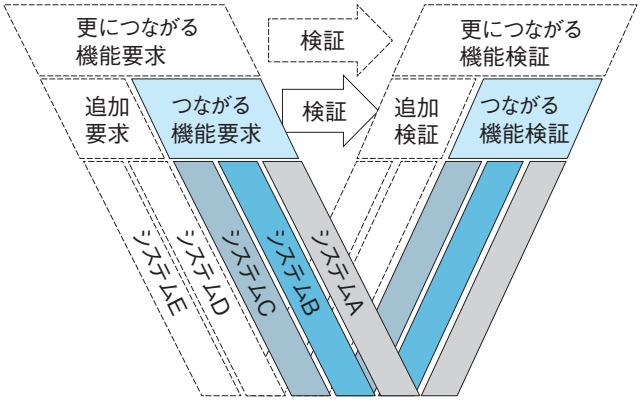


図1　つながるシステムのV字開発イメージ

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

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

2.5　つながる開発組織・プロセス

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

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

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

3　つながるシステムの検証活動

3.1　検証範囲の考え方は同じ

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

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

テストの実行についても、上位階層のシステムを組み合わせたテストは、下位階層の各システムのテストが完了してから実施するなど、単一システムのテスト実行の組み立てと基本的には同じである。しかし前述の通り、つながるシステム全体の開発をマネジメントする主体が存在しない場合があるため、テスト実行の組み立てには注意が必要で、実際には開発組織・プロセスへの関与・働きかけが必要となる。

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

3.2　検証要求分析の重要性

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

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

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

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

3.3　対象箇所を特定したテスト設計

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

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

認を目的としたものではない。

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

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

3.4 開発プロセスへの関与が有効

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

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

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

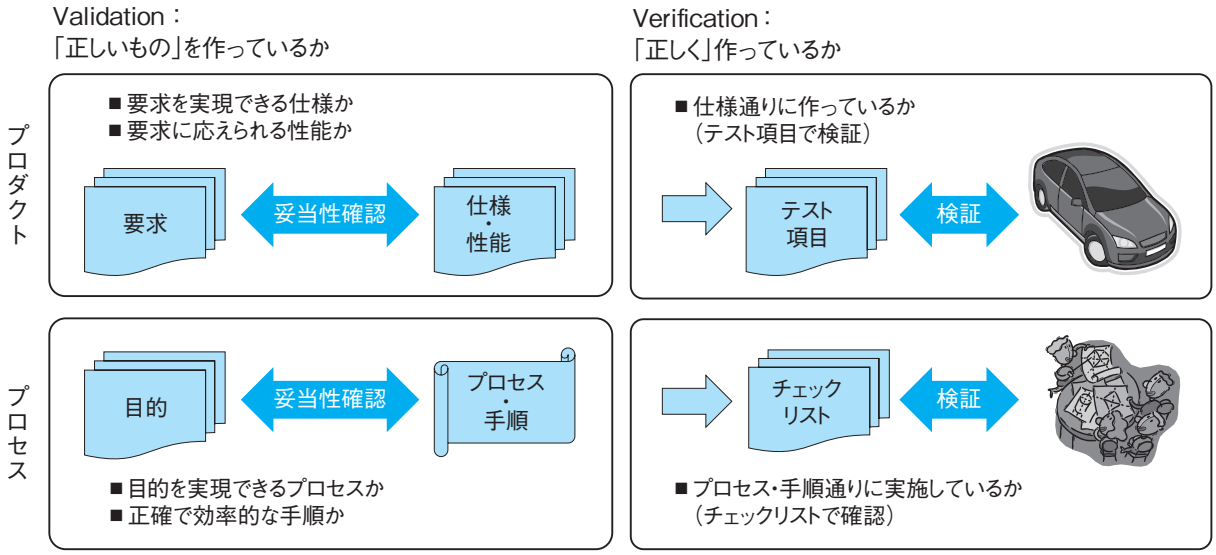


図2　プロダクトとプロセスのValidation & Verification

4 おわりに

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

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

ICTシステムのサイレント故障・予兆監視

NEC・クラウドプラットフォーム事業部・主任　西川　昌利

ICTシステムが提供するサービスの範囲の拡大に比例して運用管理コストも増大している。企業としてはビジネス拡大のために更にサービスの拡大を行いたいが、運用コストが足かせとなっており、これまでのメッセージ発生を契機に対処を行うリアクティブな運用からサイレント故障や予兆を監視するプロアクティブな運用の実現が求められている。
本稿では、サイレント監視・予兆監視の手法とそれを実現するNECのインバリアント分析技術について紹介する。

1　ICTシステム監視の目的と課題

ICTシステム監視の目的は、ICTシステムが提供するサービスの継続性を向上し、サービスのダウンタイムを短縮、サービス低下を防止することにより、機会損失の抑止や社会的信用を維持することである。しかし、従来の運用監視の仕組み（事象を想定した単体での監視）では、サービス影響につながるすべての事象を検知できておらず、このようなサイレント故障により、サービスに影響を与えてしまっていることが多く見受けられる。

とくに昨今のICTシステムは、サービス範囲の拡大や仮想化・クラウドの普及により複雑化しており、監視項目が増大・多様化し、従来のICTシステム監視ではサービスダウンにつながる事象の網羅度は更に低くなっている。

2　サイレント故障の要因

従来の運用監視に内在するサイレント故障の要因について、以下に示す。

監視の仕組みは＜図１＞に示した通り、監視対象の情報収集、収集した情報の分析、分析結果の可視化の３つの機能に分けられるが、従来の運用監視には、それぞれの機能において、サイレント故障につながる要因が内在する。

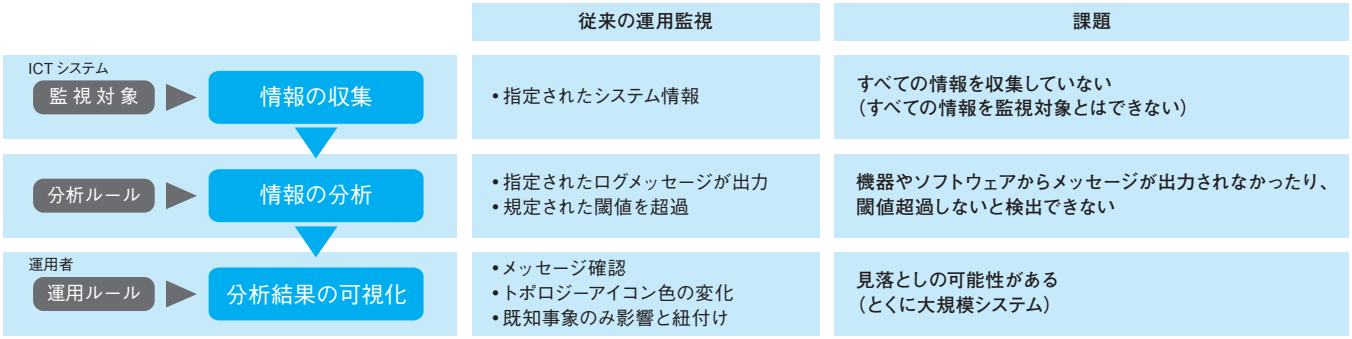


図１　内在するサイレント故障の要因

①情報の収集不足
ICTシステムのすべての部位を監視対象とはできない。

②情報の分析不足
閾値の超過や既定したメッセージが出力されなければ、故障検知できない。

③監視の表現力の不足
監視対象数が多い、メッセージラッシュなどにより、見落としが発生する。

サイレント故障を低減するためには、それぞれの機能の課題を解決することが必要となる。

3　サイレント故障監視へのアプローチ

3.1　予兆監視とサイレント故障監視の定義

ICTシステムが提供するサービスへの影響を低減するためには、サイレント故障と故障予兆について検討する必要がある。＜図２＞

●サイレント故障

サイレント故障とはICTシステムの自己診断機能（あらかじ

め準備された監視機能)で検知できない故障を指す。ICTシステムの様々な情報をもとに検知(故障前兆を含む)する手法を自己診断機能に追加することで、サイレント故障を低減することができる。

●故障予兆

予兆とは何かが起こりそうな状態を指す。ICTシステムの挙動を観察し、普段とは異なる挙動を検知する。挙動の差異の原因を確認することで、サービス影響や故障個所を特定する。

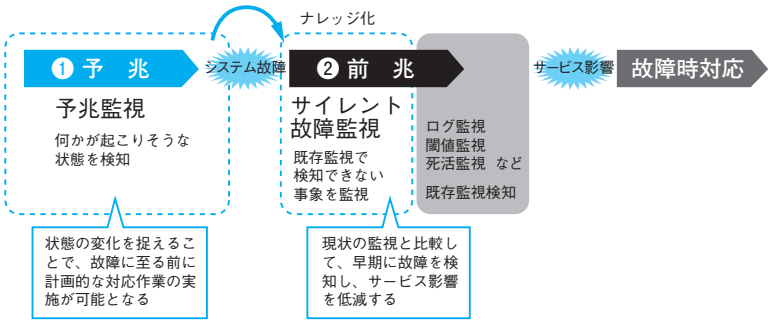


図2　サイレント故障と故障予兆

3.2　アプローチ方針

① 情報の収集不足への対応(サイレント故障の監視)

監視を行うには少なからず監視対象のリソースが利用されることから、必ずしもすべての部位が監視できるわけではない。このため、現在取得している性能値やメッセージなどから類推し、故障と紐付けて監視する必要がある。

例えば、A→B→Cと流れる処理があり、Bの監視を行っていないとする。この場合Bで故障が発生しても、サイレント故障となるが、Aの処理が行われた後にCの処理が行われないことをBの故障として定義することにより、Bの状態を監視することができる。

② 情報の分析不足への対応(予兆の確認)

①に示したような監視はシーケンスや構成を把握しているなど既知の事象に対しては、監視に組み込むことが有効であるが、逆に未知の事象に対しては定義できないという課題がある。また、ICTシステム構成やシーケンスに変更があった場合、事象の関係性を見直す必要がでてくる。

事象との紐付けやICTシステムの構成、サービスシーケンスを一旦無視して、すべての情報をもとにその関係性をモデル化し、挙動の変化を網羅的に監視する。挙動の変化を検知後にその原因を確認する運用を合わせて行うことを推奨する。

③ 監視の表現力不足への対応(可視化)

現状でも見落としはあるが、②を監視に組み込むと、監視対象が増加し、現状のメッセージとしての監視では見落としてしまうリスクが増加する。また、見せかけの関係性も取り込んでモデル化してしまい誤報となることで、運用者が注意して見なくなってしまう。このため、これまでのメッセージベースではなく、関係性の崩れ方の円グラフでの表現や、その集中度合をマップ表示させるなどの工夫が必要となる。

3.3　運用イメージ

現状のICTシステム監視は監視メッセージと事象が紐付いたものとなっている傾向にあるが、上記②の運用は紐付いたものにはならず、現状の監視運用には適合しない。このため、システムアセスメント運用を新設し、②の監視を実施。そこで故障と定義できた事象を①で監視することにより、現状の監視に適合させる。<図3>。

②での確認は、AI技術を適用し、確認手順をナレッジとして学習させることにより、故障判断もシステム化することができるようになるが、十分なナレッジが蓄積できるまでの間は、人間系で故障判断を行うことが必要となる。(故障判断のシステム化については、ICTシステム情報の内容やその標準化度合に依存するため、本稿では割愛する)

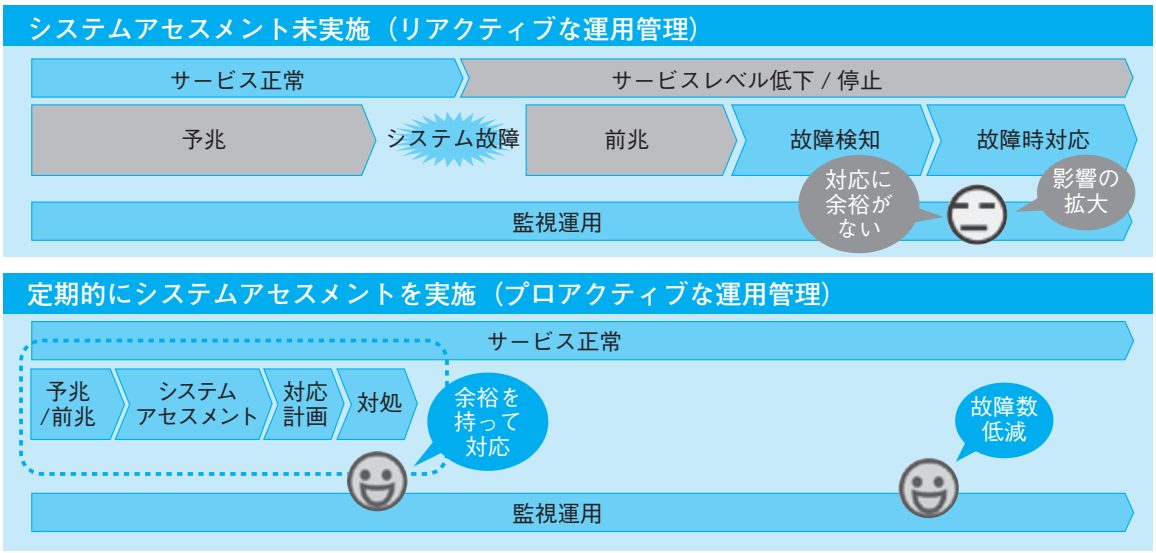


図3　サイレント故障を監視する運用

4　インバリアント分析技術

4.1　技術紹介

NECでは、ビッグデータ分析技術の1つとして、インバリアント分析技術（System Invariant Analysis Technology：以下SIAT）を開発し、ICTシステムのサイレント故障監視・予兆監視を実現するInvariant Analyzerとして製品化した。SIATの主な分析技術は以下の通りである。

●モデルの作成

性能情報などの時系列に並んだ数値データを入力することで、数値データ間にある不変的な相関関係（インバリアント）を自動的に抽出し、それぞれの関係性をy= f（x）の形式でモデル化する。＜図４＞

●挙動変化の検知

現在のデータや分析対象となる期間のデータを上記のモデルから予想される挙動と比較し、関係性の崩れを挙動変化として検知する。＜図５＞

SIATにICTシステムの性能データを入力することで、ロードバランスされた個々のサーバーのリソース使用状況の関係性（分散状態）やトランザクション量とCPU使用率などの間に成り立つ関係性の変化を監視することが可能となる。

また、Invariant Analyzerには、ICTシステムの利用特性（平日と休日など）に応じてモデル作成をアシストする機能＜図６＞や過去の挙動変化をナレッジとして登録する機能＜図７＞を具備することで、誤報の低減や原因の特定を強力に支援する。

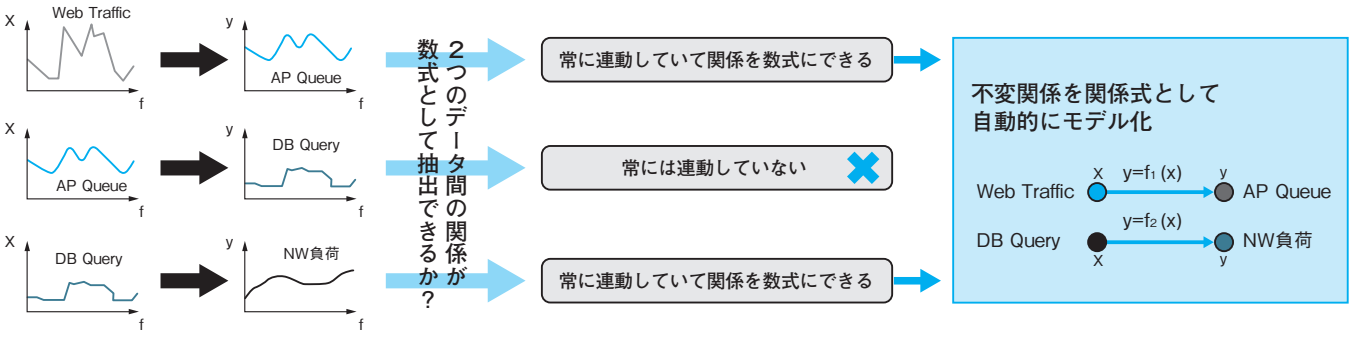


図４　相関モデルの作成

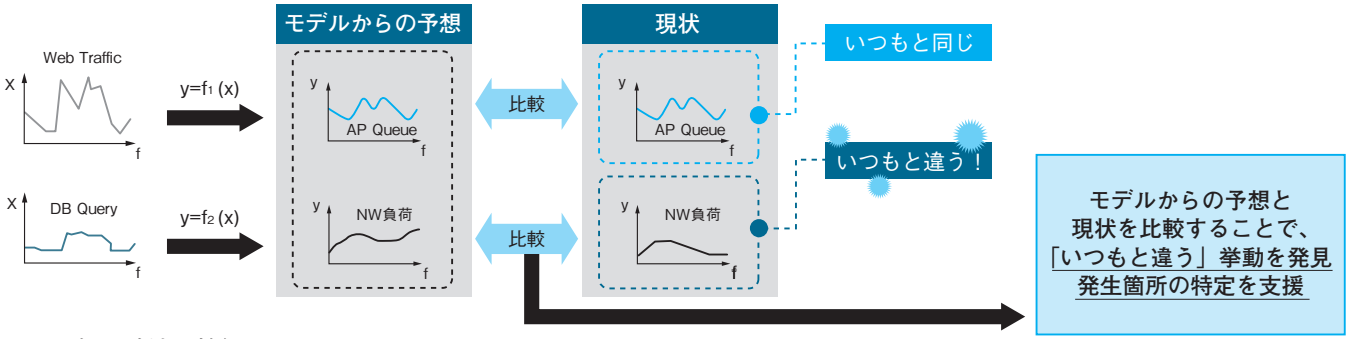


図５　相関破壊の検知

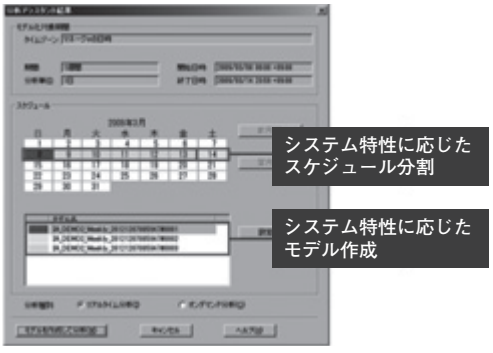


図６　モデル作成アシスト機能

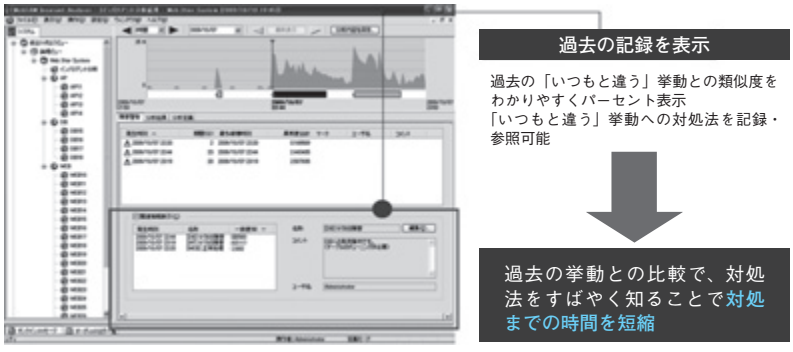


図７　ナレッジ機能

4.2 従来監視との比較

以下に従来の監視とSIATを利用した監視との比較を示す。ただし、SIATでの監視で従来の監視で検知できた事象を網羅できるわけではなく、従来の監視とSIATを利用した監視を併用することで、ICTシステムの監視品質を向上する。（性能情報をもとにした監視の比較。ログ監視や死活監視は割愛）

●ベースライン監視との比較

ベースラインは個々の測定時間ごとに取り得る値の範囲を学習するため、モデル作成までに長期間（最低でも数週間程度）を要する。対して、SIATを利用した監視は、性能情報間の関係性を見ることから、最低100ポイントのデータ（1分単位のデータであれば100分間）があれば十分に高い精度のモデルを作成することが可能であり、モデルの劣化時など、再作成の影響が少ない。

イベントなどで一時的にICTシステムの負荷が増大する場合、ベースライン監視では正常に稼働していても誤報となるが、SIATでは関係性を監視しているため、正常に稼働していれば負荷が増大しても誤報とはならない。

また、ベースライン監視は個々の監視項目の挙動を監視するものであり、監視項目以外の事象について検知できない場合がある。

●閾値監視との比較

閾値監視は上限、下限値を設定することで、危険な水準に達した場合に検知することが可能であるが、下限値に関しては、閑散時間帯を加味しなければ、誤報となる可能性がある。

また、監視項目ごとに閾値を設定する必要があり、項目に合わせて閾値を調整するような場合はメンテナンス性が低下する。

4.3 SIATを利用した監視例

例えば、個々の装置単位の受信パケット総計と送信パケット総計には強い相関関係があり、その相関関係を監視することにより、機器のリソース枯渇や間欠故障時などに発生する異常な破棄パケットの発生を検知することができる。（破棄パケットは通常時でも発生するため、そのカウンター情報を監視していても故障が検知できるとは限らない）

●従来の監視＜図8－①＞

ポート単位にパケット量の推移を監視しても閾値を超過していないため、故障を検知できない。

●ネットワーク監視ツールで、送受信パケットを集計して確認＜図8－②＞

装置単位の送受信パケットには強い相関関係があるため、グラフを重ね合わせて目視確認することで、相関関係の崩れを見つけることは可能。しかし、業務パケットと比較して、破棄パケットの量が少ないため、目視ではすぐに気づけない可能性がある。

●相関係数の変化を監視＜図8－③＞

相関度の強さを示す値として、相関係数がある。この相関係数の変化を確認することにより、グラフを重ね合わせただけでは見つけることができなかった微細な相関関係の崩れでも検知することができる。SIAT技術を導入したInvariant Analyzerにより、このような相関関係の崩れをリアルタイムで検知することが可能となる。

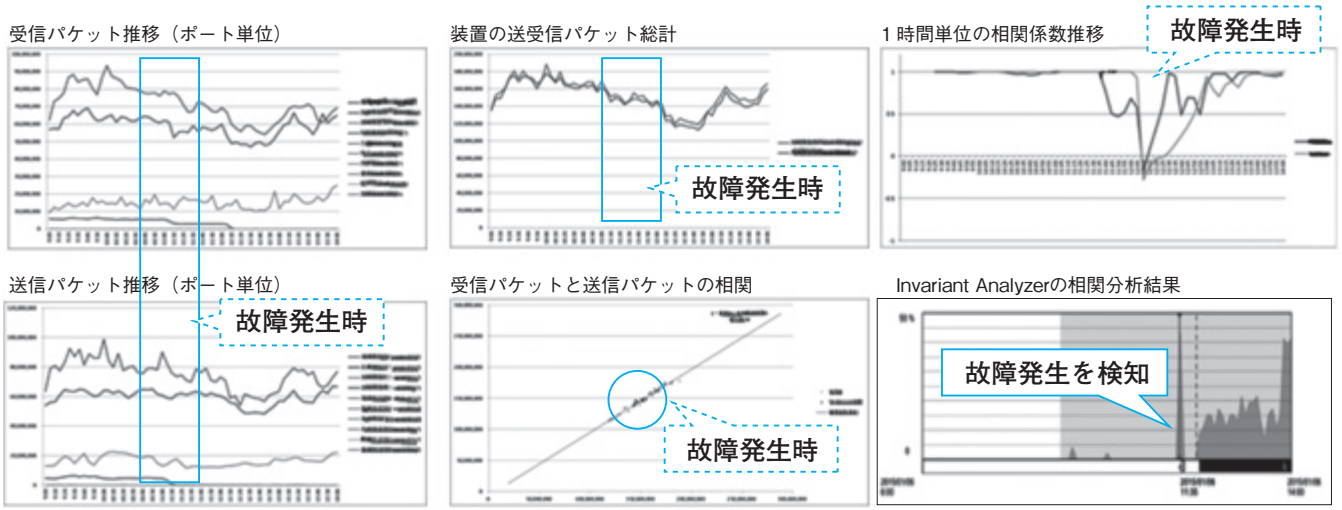


図8　相関監視の効果

5 分析事例

Invariant Analyzerでは＜表９＞にある通り、様々な業種のICTシステムにおいて既存で取得されているデータを用い、サイレント故障や障害予兆など従来の監視では見つからなかった故障を検知している。

No.1のWeb3Tier業務システムの分析事例について、詳細を以下に示す。

Web/AP/DBサーバーよりCPU、メモリ、ネットワーク、スワップ、ディスクなどの性能情報を取得。平常時をモデル化し、各性能データを分析。結果として、(ア) APサーバー処理遅延、(イ) DBサーバーへの負荷集中、(ウ)通常業務にないWebアクセスの発生を検知した。＜図10＞

6 今後の方針

6.1 分析対象拡大

ICTシステムの状態を示すデータとして、性能情報のほかにログメッセージやプロセス間通信やファイルのアクセスなどがある。NECはSIATでの性能情報の分析を始めとしてシステムから得られるあらゆる情報を活用し、ICTシステムの信頼性向上を支援する。＜図11＞

●ログメッセージの分析

とくに大規模システムにおいては、個々のICTシステム間に依存関係があるが、サイロ型に構築された結果、生成されるログメッセージは標準化されておらず、かつ大

表９　主なSIAT分析事例

No.	分析事例	分析データ	検知した異常	効果
1	Web3Tier業務システム	CPU、メモリ、ネットワーク、スワップ、ディスクの性能データ	• APサーバーの処理異常 • DBサーバーへの負荷集中 • 通常ではないアクセス発生	• 障害の検知 • 障害原因の絞り込み
2	Oracle DB分析 (1)	Oracle、OS、ネットワークの性能データ	• SQL実行回数の異常	• DB障害の検知 • 障害原因の絞り込み
3	Oracle DB分析 (2)	Oracle Statspackデータ	• Global Cacheの異常	• DB障害の検知 • 障害原因のSQL単位での絞り込み
4	NW分析	NWトラフィック	• トラフィック異常	• NW障害の早期発見 • 原因箇所のポート単位での絞り込み
5	NW・サービス性能分析	NWトラフィック、サービス要求／処理	• トラフィック異常 • 処理遅延	• サービス性能劣化の早期発見 • 原因箇所のサービス提供箇所単位での絞り込み
6	Netflow分析の検証	Netflowデータ	• スイッチ間ネットワーク負荷 • スイッチ間STP無効化 • ICMP大量投入	• NW障害の検知 • 障害原因のIPアドレス単位での絞り込み

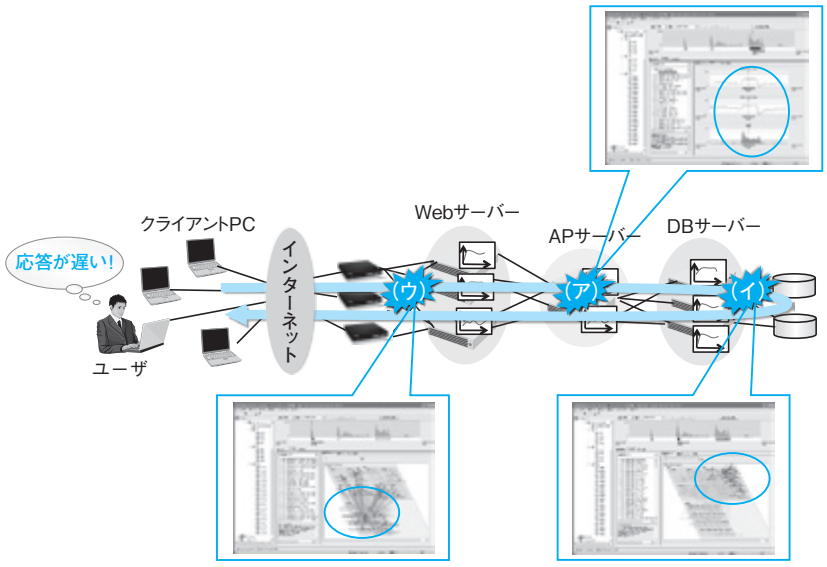


図10　Web3Tierシステムでのサイレント故障の検知事例

量であるため、予兆を含めた故障の見落としにつながるリスクがあった。

これらの大量かつ多種のログメッセージをオペレータが分かりやすい形に整理、可視化し、ログメッセージの各イベント間に成り立つ規則性を学習することで、既存の監視ルールでは見つけにくい故障の検知やその予兆を早期に検出する。

なおログメッセージの分析には以下が想定される。<表12>

●プロセス間通信、ファイルアクセスの分析

モデル対象となるログメッセージが出力されない場合や性能傾向に変化のない故障についてはSIATやログメッセージの分析を利用しても検知することが困難である。

プロセス間通信やファイルへのアクセスをモデル化し、挙動比較することで、SIATやログメッセージの分析で検知困難な故障や予兆を検知し、より高度な監視を実現する。

6.2 適用領域拡大

NECではますます複雑化・高度化する社会課題に対し、人とAIが協調しながら高度な叡智で解決する方針である。

SIATについては、インフラ／プラント・マネジメントとして発電所などの故障予兆監視を実現している。今後も様々な業種・業務へ適用領域を拡大していく。

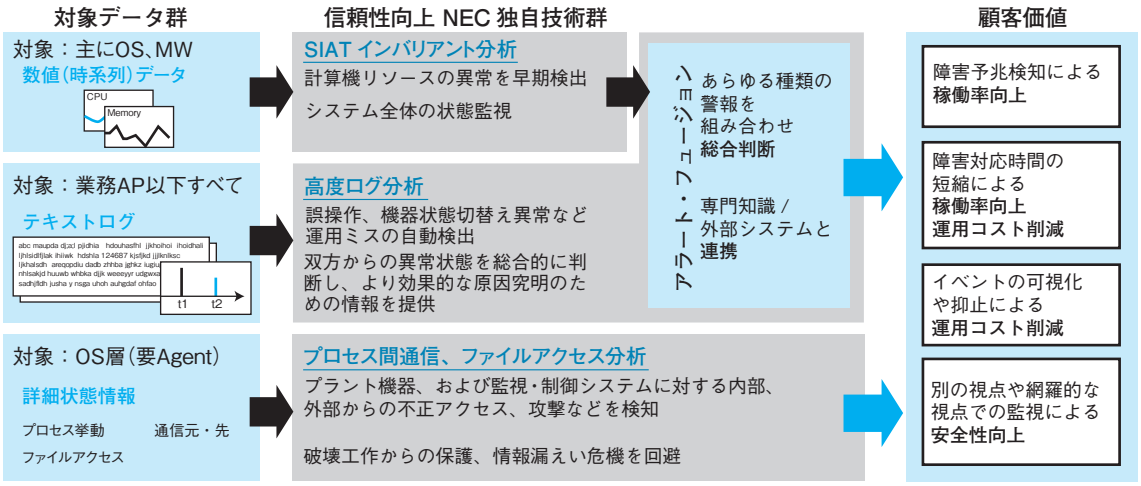


図11　あらゆる情報を活用したICTシステムの信頼性向上

表12　ログメッセージの分析例

	監視	予兆確認
エラーメッセージ抽出	エラーステータスのメッセージが出力されていないか確認する。	ワーニングを含めたメッセージの発生回数の推移を確認する。
処理抜け	通常発生するメッセージが出力されているか確認する。	同左。 メッセージが出力されていない原因を確認する。
通常異なるシーケンス	通常と異なるメッセージが出力されていないか確認する。	同左。 メッセージが出力されている原因を確認する。
戻り値	不正な戻り値がもどされていないか確認する。	同左。 戻り値の妥当性を確認する。
特定処理の追跡	シーケンスにそって処理を追跡し、異常や遅延がないことを確認する。	—
操作履歴	不正なオペレーションが実施されていないか確認する。	同左。 オペレーションによる異常がないか確認する。
トランザクション量確認（スループット）	処理の発生回数を確認する。	同左。 トランザクション量によるリソース利用状況を確認する。
処理時間の確認（TAT）	処理の開始から完了までを確認する。	同左。 処理遅延発生していないか確認する。

：一般的なログメッセージログ監視での分析　：特定部のみ監視されている分析

技術力向上を目的に全社で活用
…三菱電機コントロールソフトウェア株式会社

研修事業の教材として活用…株式会社オージス総研

SEC journal編集部

IPA/SECでは、日々の事業・研究成果を様々な形態で展開している。その一つが、ソフトウェア高信頼化を支援する書籍の出版だ。中でも「SEC BOOKS」は、ソフトウェア開発に携わる現場で求められるハウツーやインテリジェンスを網羅した実践書として、開発者から一定の評価をいただいている。本連載では、日々の業務で、実際にSEC BOOKSを役立てている企業の事例を紹介する。第一回目は、2016年４月に刊行した、「【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］」を活用している「三菱電機コントロールソフトウェア株式会社」と「株式会社オージス総研」に御登場いただく。

事例１　三菱電機コントロールソフトウェア株式会社

1980年設立の三菱電機コントロールソフトウェア株式会社（以下、MCR）は、三菱電機株式会社100％出資のグループ企業である。「ソフトウェア開発事業」「システムインテグレーション事業」「ソフトウェアパッケージ事業」「ハードウェア開発事業」を基幹事業とし、社会・公共、交通、電力や工業・産業などでの監視制御システム、更には自動車関連のカーメカトロニクスやカーエレクトロニクス機器など、社会インフラを担う重要なシステムやソフトウェア、機器開発を手がけている（表１）。

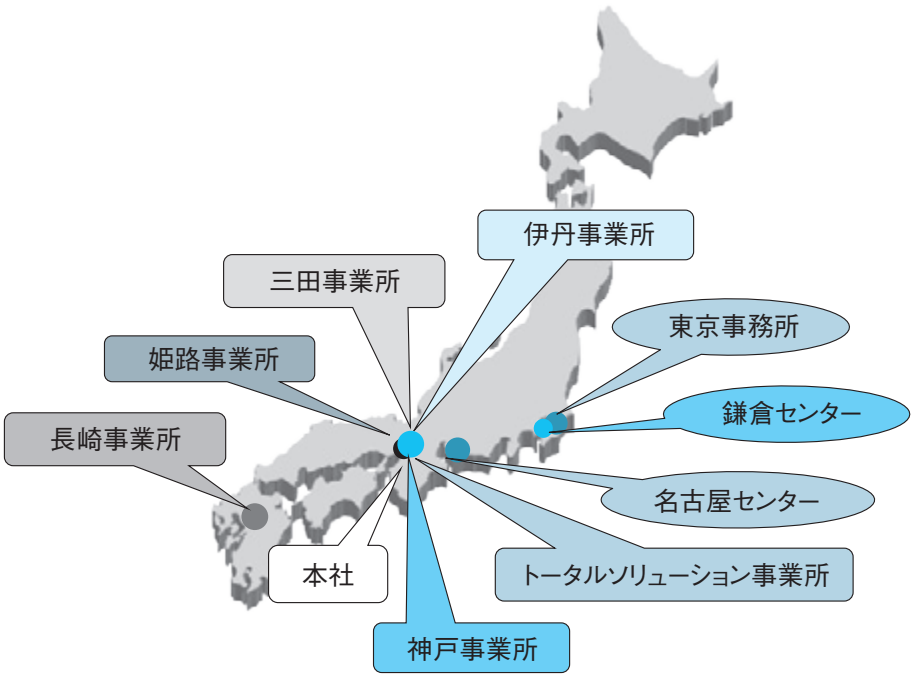


図１　事業所の所在地

表１　事業内容

＜ソフトウェア開発＞ ●社会・公共/交通/電力/工業・産業 ●カーエレクトロニクス/カーマルチメディア
＜システムインテグレーション＞ ●FA・PA/省エネ
＜ソフトウェアパッケージ＞ ●シーケンサ対応/MES対応
＜ハードウェア開発＞ ●産業用パソコン ●個別用途向基板開発

同社は、「【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］」を880冊購入し、コーディングに直接携わる全社員に配布した。また、品質管理や技術スタッフ部門にも一冊ずつ配布。今年度は新入社員研修にも利用したという。その目的についてMCR取締役社長を務める池田一成氏は、「社員の技術スキル向上が狙いだった」と語る。

かねてからMCRは、「仕組みの強化」「技術力向上」を２本柱に据え、全社を挙げて組込み系開発の品質改善に取り組んでいた。その一環として、社員の技術スキルの棚卸しを実施したところ、会社側が求めるレベルに達していない技術者がいることが判明した。入社数年目の会社の中核となるべき技術者のスキルが、想定と乖離していたのだ。

なぜそのような事態が起こってしまったのか。同社でソフトウェア生産技術・人材開発部長兼ソフトウェア生産技術課長を務める折方孝夫氏は、「最近はフルスクラッチで開発する機会が少なくなった。以前はOJT（On-the-Job Training）でスキルの向上が図られたが、そうした環境が変わってしまったのが一因だ」と説明する。

このままでは、開発者の技術力向上は望めない―。そんな危機感を抱いていたとき、ある事業所が「【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］」をテキストに勉強会を実施し、開発者のスキルアップを実現しているという取り組みが報告された。しかも、一定の成果を上げているという。池田氏は、「全社会議でこの話を聞き、すぐに組込みソフトウェア開発向け コーディング作法ガイドを利用したスキルアップ施策を全社に展開することを決めた」と語る。

現場技術者の再教育教材として活用

「【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］を全社展開する際、留意したことは２つある」と、折方氏は説明する。１つは本書を活用した勉強会で、メンバーのスキルアップが達成したことを成功事例として説いたこと。もう１つは、本書の活用方法を、各組織の課長に委ねたことだ。これが功を奏し、今では各職場で工夫しながら、それぞれのスキルアップ施策を実施しているという。

折方氏は、「本書の良い点は、質的特性（信頼性、保守性、移植性、効率性）ごとに『すべき』『べからず』が記述されていることだ。こうしたアプローチは、ほかの書籍で見たことがなかったため、目から鱗が落ちる思いだった。また、良いコーディング例と悪いコーディング例が対比して載っている点も、技術者にとっては理解しやすい」と評価する。

新人が一から学ぶ“教科書”というよりも、C言語のコーディングの知識がある技術者への再教育時の教材として効果を発揮しているという。

一方、今後の改訂版に期待する点は、「マルチコア、マルチスレッドでの留意点を取り上げること」だ。組込みソフトウェア開発の場でもマルチコア、マルチスレッドでの処理形態が主流になりつつある。「現場でのニーズを考えれば、次の改訂版でぜひ言及して欲しい」との指摘をいただいた。

現在は、生産技術部門及び品質管理部門でMCRのコーディング規約や、チェックリストにも【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］を活用している。更に今後は品質強化に向けた共通講座や、若手ソフトウェア開発者を対象とした基礎講座の中でも利用していくとのことだ（図２）。

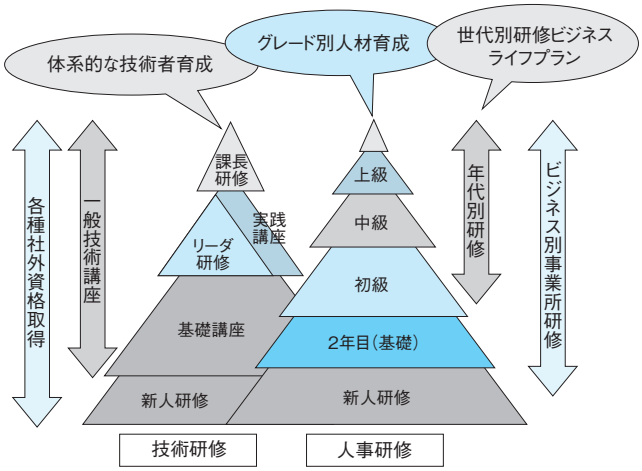


図２　技術研修・人事研修の全体イメージ

会　社　名	三菱電機コントロールソフトウェア株式会社
本社所在地	兵庫県神戸市中央区
設　　立	1980年10月１日
代　表　者	取締役社長　池田一成
資　本　金	３億円
売　上　高	248億円（2016年３月実績）
事業内容	社会・公共、交通、電力、車、産業など各種制御システム・ソフトウェアの開発・設計
従業員数	1,445名（2016年３月末実績）

事例 2　株式会社オージス総研

大阪市に本社を構える株式会社オージス総研は、大阪ガス株式会社の情報システム部門を母体とした、ベンダーフリーの情報処理サービス事業者である。コンサルティング、システムインテグレーション、プラットフォームサービスを事業の柱とし、日本でいち早くオブジェクト指向を導入・リードしてきた事業者としても知られている（図３）。

また、同社はIT人材の育成事業にも注力しており、ITスキル向上を目的に、研修・トレーニングを実施している。研修コースは、アジャイル、モデリング、超上流、品質向上などに関連するテーマが中心で、定期的に開催している「オープンコース」と、顧客企業のニーズに合わせてカスタマイズし、顧客企業先で実施する「オンサイトコース」がある。

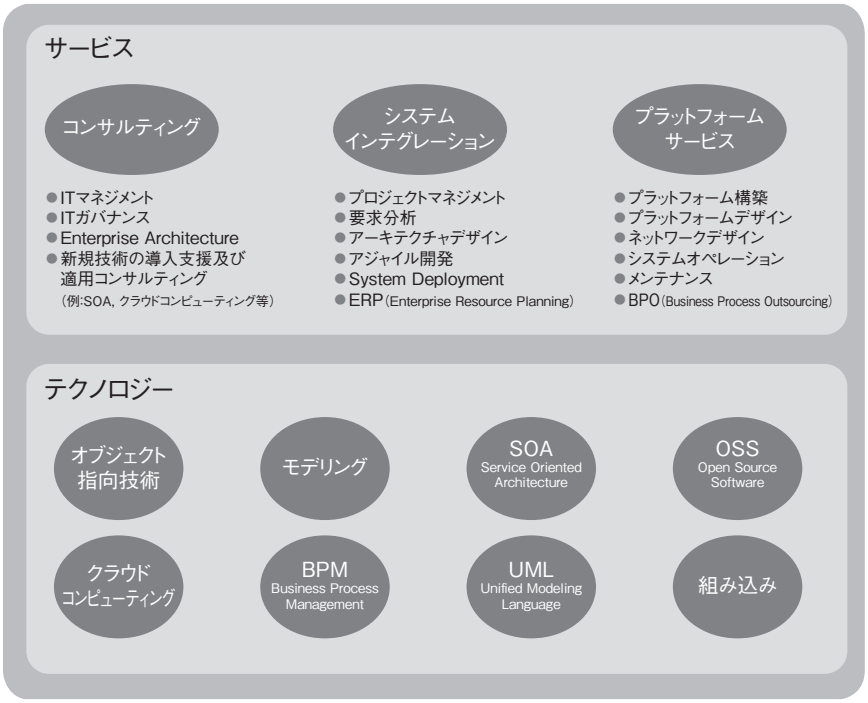


図３　株式会社オージス総研事業内容

オージス総研では、2015年より「【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］」を同社の研修コース「品質向上のためのソフトウェア設計・実装基礎」（図４）で、教材として利用している。同コースは、組込みソフトウェア技術者として知っておくべき信頼性・保守性を高めるコーディング作法と、品質の高いモジュールを設計する上で重要な凝集度と結合度の考え方を習得することが目標だ。顧客企業先で実施するオンサイトコースとして、既に４社ほどの実績がある。

図４　SEC BOOKSを用いた研修コース（株式会社オージス総研Webサイトより）

JIS X 25010の品質特性による分類、体系化が使いやすい

「【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］」を教材とした理由について、同社コンサルティング・サービス部でシニアコンサルタントを務める辻博靖氏は、「研修教材として利用する以前から、製造業のお客様に対するコンサルティングの現場で利用していた」と話す。その現場では、ソフトウェア製作において何が常識なのかわからないエンジニアが多く、その常識を学ぶための勉強会で本書籍が利用されていた。勉強会の参加者から品質に対する意識が高まったという声があったので、新規に立ち上げる研修コースでも本書籍を採用した。

同コースは、受講者に研修開催約一カ月前に本書籍を送付し、各自事前学習した上で研修に参加する反転学習形式になっている。限られた研修時間の中で実践機会をなるべく多く確保するため、作法（書き方）は本書籍による事前学習となっている。

そのうえで研修時には、問題のあるソースコードを提示し、受講者に問題点をその理由とともに指摘させ、書き改める形の演習を中心に行っている。単に作法（書き方）だけを覚えさせるのではなく、その作法の背後にある理由（解決できる問題）も理解できるように演習を用意することで、ソフトウェア開発現場での活用力を高めている。

辻氏は本書を「各作法がJIS X 25010に基づき４つの品質特性に関連付けて分類・整理されているため、数多くの作法を理解しやすいように体系立てて教えられるというメリットがある」と評価する。また、受講者からは「事前配布のテキスト（本書）がとてもわかりやすかった」「今までの研修は、既存ソフトや現状の開発と違いが大きく、受講しても活用できないものが多かったが、この研修内容はすぐに業務に活用できる」との声もあり、好評を博しているという。

社　　名	株式会社オージス総研
代 表 者	代表取締役社長　西岡信也
設　　立	1983年６月29日
資 本 金	4.4億円（大阪ガス株式会社100%出資）
売り上げ実績	64,897百万円（連結） 35,887百万円（単体）（2015年度）
従 業 員 数	3,227名（連結）　1,386名（単体）

（2016年３月31日現在）

活用いただいているSEC BOOKSはこちら

本書は、C言語を用いて開発されるソフトウェアのソースコードの品質をより良いものとすることを目的としている。コーディングの際に注意すべきことや、そのノウハウを実践に即して紹介。また、組込みソフトウェア作成時に決定する組織やグループ内のコーディングルールをはじめ、ソースコードの標準化、品質の均一化を進める方法についても説明している。また、ソフトウェア品質と同様に、コードの品質も「信頼性」「保守性」「移植性」などの品質特性で分類できるという思想に基づき、コーディングの作法とルールを「JIS X 25010 ソフトウェア製品の品質特性」をもとに体系化している。

旧版（Ver.1.1）からの主な改訂の内容は、以下の通り。

- 準拠するC言語規格をC90からC99にし、JIS規格側で変更・修正された部分を反映。
- C99から新たに規定された機能などに対応し、より効率的で不具合の起きにくいコーディングができるように一部内容を追加・更新。
- 改訂版MISRA C:2012（2013年３月公開）に対応し、本書が参照している両者共通的な部分で記述の食い違いを修正。
- 旧版からのユーザも自然に移行できるよう、旧版と共通する部分のルール番号や体裁を再構成。

【改訂版】組込みソフトウェア開発向け コーディング作法ガイド［C言語版］ESCR Ver.2.0

〈購入方法〉
http://www.ipa.go.jp/sec/publish/index.html
独立行政法人情報処理推進機構（IPA）技術本部
ソフトウェア高信頼化センター（SEC）編集・発行
ISBN：978-4-905318-40-8
B5変型判・181頁
定価1,619円（税抜）
2016年２月26日「２版１刷」発行

Column

シェアリング・エコノミーの時代、「所有」から「利用」へ

IPA顧問 松田 晃一

車は持たずにカーシェア

近所のコインパーキングに最近「カーシェア」ののぼりが立つようになった。

マイカーを持つのではなく、必要なときに必要な時間だけ借りて使えるシステムだ。

1960年代半ばの高度経済成長期には、自動車（Car）と共に、カラーテレビ（Color television）、クーラー（Cooler）の3つの耐久消費財、この頭文字をとって「3C」とか、「新・三種の神器」とも言われたが、これらを持つ家庭生活が夢であり、豊かさや憧れの象徴となっていたことを思い出す。それからほぼ半世紀、今や「多く持つ」より「賢く使う」ことのほうが「豊かだ」と考える消費者が増え、無駄な買い物をしない消費動向へ変化してきていることが背景にあるようだ。

確かに、高価な自動車を買っても平日はほとんど車庫。週末に近所への買い物やたまのドライブに使っても後は車庫。高額な資産でありながら、その稼働率は極端に低い。一説によれば、マイカーの不稼働時間は96%に上るとのことである。その上、車検や車庫、保険など持つことの手間やコストも必要である。それらを考えれば、「所有」するよりも、「利用」するほうがよほど合理的だ。もちろんレンタルサービスは以前からあったが、大きな違いは使い勝手の良さ、手続きの簡便さである。スマホなどから必要なときにいつでも直ぐに簡単に車を使えるITシステムの完備が最近の普及の原動力であることは間違いない。

サーバーは持たずにクラウドで

一方ITの分野では、既に2006年頃から「所有」から「利用」へ、「持たないIT」の流れが急速に動き出している。クラウドコンピューティングである。ハードを売り、受託ソフトを販売していたベンダーにとっては、モノが売れなくなっては大きな打撃を受けるわけで、どのITベンダーも早々に自らクラウドサービスを開始し、「製品販売」だけではなく、「サービス提供」業者としてのマーケットの確保に手を打っている。しかし、旧来のITベンダーとは全く異なる業種から参入した企業が業種の壁を壊し、新しいプレーヤーとして市場の大きな地位を占めるようになってきている。

ビジネスモデルはC to Cへ

カーシェアにしろクラウドにしろ、対象となる自動車やコンピュータはサービス提供業者が所有するものであり、企業の所有する資産を消費者が利用するB to Cのビジネスモデルだ。これに対して最近では、個人が保有する遊休資産を別の個人に貸し出すという形の仲介サービスが続々と生まれている。C to Cのモデルへの拡張だ。

例えば、今話題の民泊は自宅の空き部屋を他人に貸し出すサービス、自動車の空いている座席を同じ目的地へ行きたい人に貸し出す相乗りサービス（ライドシェア）、空いている駐車場を一時的に貸す駐車場シェアサービス、日本では規制によってサービスは始まっていないが、空いているマイカーを運転付きで借りる配車サービスなどなど。いわゆる「シェアリング・エコノミー」である。

貸主は遊休資産の活用による収入、借主は所有のコストを負担せずに利用できるメリットを得られる。しかし、このためには貸主と借主の信頼関係をいかに築き安心して貸借ができるようにするか、という問題があり、仲介サービスの最も重要なポイントとなる。

シェアリング・エコノミーのインパクト

シェアリング・エコノミーの成功は、利便性や経済性のメリットだけではなく、モノを通してその所有者と利用者の間に人としてのつながりを生み、それが新しい仲間作りへ発展する期待もある。民泊を切掛けにして借主と貸主が交流を始める、などの例も現れているようだ。古き良き時代は、米や味噌を切らすと隣近所からちょっと拝借するという付き合いができるコミュニティが当たり前にあったと聞く。現代社会では失われてしまったこのような人間関係を、シェアリング・エコノミーによって再び取り戻し、モノの貸し借りを超えた人間同士の交流が始まるきっかけになるのかも知れない。

「所有」から「利用」へ、「B to C」から「C to C」へのパラダイムシフトは、単なる経済効果だけではない大きなインパクトを生活スタイルに与えそうだ。

スマホのボタンを押せば、近くで空いている誰かの自動運転車が、自宅の玄関に迎えに来て行き先まで送ってくれる。用を済ませて乗り捨てれば、後は自動運転車が持ち主の車庫まで無人で帰る、などという“Car as a Service”もそんなに先ではなさそうだ。

書籍紹介

Safety-Ⅰ & Safety-Ⅱ
― 安全マネジメントの過去と未来

エリック・ホルナゲル　著
北村　正晴（東北大学名誉教授）・
小松原　明哲
（早稲田大学理工学術院教授）　監訳

ISBN：978-4-303-72985-1
海文堂出版刊
A5判・216頁
定価2,700円（税抜）
2015年11月刊

安全方策を考える上で「うまくいかなくなる可能性を持つこと（Things that might go wrong）」を取り除く（Safety-Ⅰ）のではなく、「うまくいくこと（Things that go right）」の理由を調べ、それが起こる可能性を増大させる（Safety-Ⅱ）ことが現代社会において必要になってきたと述べている。

悪い結果をもたらす原因をすべて除去すれば安全が達成されると考えるSafety-Ⅰであるが、IoT時代においては脅威の完全除去は困難である。かつてのようにユーザを定義し自装置や、自システムにおいて脅威を洗い出せたにしても不十分である。仮に設計時にすべての脅威に対応・除去できたとしても、製品やサービスのライフサイクル全体で脅威の変化がないとは考えにくい。更には大震災のような自然要因、また社会要因の除去は不可能だろう。

現代社会の安全を考えるに際しては、これらの実態に目を向けてみることが必要であり、これに対応した安全方策がSafety-Ⅱである。

ホルナゲル教授はSafety-Ⅰを否定はしていない。またすべてがSafety-ⅠからSafety-Ⅱに置き換わると言っているわけではない。脅威の状況に柔軟に対応して安定を保つ、レジリエンス力強化の必要性を主張している。　（遠藤秀則）

マイクロサービスアーキテクチャ

Sam Newman　著
佐藤　直生　監訳
木下　哲也　訳

ISBN：978-4-87311-760-7
オライリージャパン刊
A5判・344頁
定価3,400円（税抜）
2016年2月26日刊

Webサービス企業を中心にいち早くサービスを開始し、ユーザの反応を見ながら軌道修正や機能の追加などを次々に実施していくスピード感のあるサービスが求められている。そうした流れを実現する新しいシステム・ソフトウェア開発のアーキテクチャスタイルであるマイクロサービス（Microservices）が注目されている。ビジネス機能に沿って複数の小さいマイクロサービスに分割してそれらを連携させることで、迅速なデプロイ、優れた回復性やスケーラビリティを実現するという。本書はマイクロサービスの特徴や概念、採用する上で必要となる技術について著者の体験談やNetflixやAmazonでの事例を交えながら広く記述されている。

技術書だと思って読み進めていたら目を引いたのが10章の「コンウェイの法則とシステム設計」であった。モノリシックな世界では開発者が運用上の懸念に無関心になってしまう傾向があり、このような思考の開発者をマイクロサービスのプロジェクトにアサインするのは危険だというものである。開発組織の設計にまで解説は広がっている。

本書は、特定の言語や技術に特化していないのでマイクロサービスの採用を考えていなくても新たなシステム開発のスタイルを学ぶ書籍として、また、マイクロサービスの持つ特徴から、本書を俯瞰することにより、アプリ／インフラそして運用の境界なく働くことができるDevOpsエンジニアの参考書にも適していると感じた。　（遠藤秀則）

編集後記

今号はセキュリティ設計を特集しています。IoTが普及、浸透してくるとパソコンやスマートフォンだけではなく、様々なモノやサービスにおいて、セキュリティを今以上に意識する必要があることは誰もが感じることと思います。しかしながら、作り手側にとって将来的に予想もしなかったモノにつながる環境では、どこから手を付けたら良いか、何を考えたら良いのか難しいのではないでしょうか。本号でそのヒントをお伝えできれば幸いです。IPA/SECから発信する情報は技術が中心なのですが、後藤先生との対談では、「そもそもセキュリティは技術だけでは守れない。ソーシャル・エンジニアリングなどの、人の要素も含めて考えなければならない」ことや、「個別のしっかりした知識だけでなく全体的な視点が大切」といった技術以外の大切さも伺うことができました。技術情報と併せてお伝えしていきたいと思います。(編集長)

編集部より

次世代のソフトウェア・エンジニアリングに関して等、忌憚のないご意見をお待ちしております。下記のFAXまたはメールにてお気軽にお寄せください。

SEC journal 編集部　FAX：03-5978-7517
e-mail：sec-journal_customer@ipa.go.jp

クリスマスマーケット　撮影：K.Hasegawa

SEC journal 第12巻 第3号(通巻50号)　2016年12月1日発行

編集兼発行人　独立行政法人情報処理推進機構
技術本部　ソフトウェア高信頼化センター
所長　松本 隆明
〒113-6591　東京都文京区本駒込2-28-8　文京グリーンコート センターオフィス16階
Tel：03-5978-7543　Fax：03-5978-7517
URL：http://www.ipa.go.jp/sec/　e-mail：sec-journal_customer@ipa.go.jp

SEC journal 論文募集

独立行政法人情報処理推進機構（IPA） 技術本部 ソフトウェア高信頼化センターでは、下記の内容で論文を募集しています。

論文テーマ

・ソフトウェア開発現場のソフトウェア・エンジニアリングをメインテーマとした実証論文または先導的な論文
・ソフトウェアが経済社会にもたらす革新的効果に関する実証論文

論文分野

品質向上・高品質化技術、レビュー・インスペクション手法、コーディング手法、テスト／検証技術、要求獲得・分析技術、ユーザビリティ技術、プロジェクト・マネジメント技術、設計手法・設計言語、支援ツール・開発環境、技術者スキル標準、キャリア開発、技術者教育、人材育成、組織経営、イノベーション

応募要項

締切り　：1 月・4 月・7 月・11 月　各月末日
査読結果：締切り後、約 1 カ月で通知。「採録」と判定された論文は SEC journal に掲載されます。
応募方法：投稿は随時受付けております。応募様式など詳しくは HP をご覧ください。
http://www.ipa.go.jp/sec/secjournal/papers.html

SEC journal 論文賞

毎年「採録」された論文を対象に審査し、優秀論文には SECjournal 論文賞として最優秀賞、優秀賞、所長賞を副賞と併せて贈呈します。

IoT 時代に活躍する**【組込みシステムの腕利きエンジニア】**を目指す！

国家試験 エンベデッドシステムスペシャリスト試験

高度な実践能力の証明に！

▶ 身近な場面を想定した出題を通して、最適な組込みシステム実現のために必要となる高度な実践能力（レベル 4）を問います。
（レベル 4 の定義）：専門分野において、自らのスキルの活用によって、独力で業務上の課題の発見と解決をリードするレベル。

技術要素	開発技術	管理技術
プロセッサ、メモリ、バス、計測・制御、リアルタイム OS、プラットフォーム、電気・電子回路、ネットワーク、セキュリティ	・要求分析の実行とレビュー ・設計の実行とレビュー ・テストの実行とレビュー	・開発環境マネジメント ・知財マネジメント ・構成管理、変更管理

▶ 近年の試験では、「無線通信ネットワークを使用した安全運転支援システム」、「3 次元複写機」、「通信機能をもつ電子血圧計を用いた健康管理システム」、「非接触型 IC カードを使用した入退場ゲートシステム」などのテーマを出題しました。

▶ 自動車、家電、モバイル機器などに搭載する組込みシステムや重要インフラの制御システムを、ハードウェアとソフトウェアを適切に組み合わせて構築し、求められる機能・性能・品質・セキュリティなどを実現できる組込みエンジニアを目指す方に最適です。

試験概要

【試験区分】エンベデッドシステムスペシャリスト試験（情報処理技術者試験 高度試験の 1 区分として実施）
【日　　時】年 1 回の実施（毎年 4 月第 3 日曜日）
【申込受付】毎年 1 月中旬から 2 月下旬（予定）まで WEB・郵送で申込み受付

詳しくは、Web ページをご覧ください。http://www.jitec.ipa.go.jp/index.html
試験概要の最新情報、過去問題、活用事例などをご紹介しています。

IPA Better Life with IT

SEC journal No.47
第12巻第３号（通巻50号）
2016年12月１日発行

ISSN 1349-8622