2014 年 7 月 1 日発行
第 10 巻 第 2 号（通巻 39 号）
ISSN 1349-8622

SEC journal 37

巻頭言

木村 英紀　独立行政法人科学技術振興機構（JST）
研究開発戦略センター（CRDS）　上席フェロー

所長対談

國井 秀子　芝浦工業大学大学院工学マネジメント研究科　教授
一般社団法人情報サービス産業協会（JISA）　副会長

ソフトウェア産業の活性化と変革への道筋

論文

要件定義プロセスと保守プロセスにおけるモデル検査技術の開発現場への適用

松浦 佐江子　芝浦工業大学／小形 真平　信州大学／青木 善貴　日本ユニシス株式会社
谷沢 智史　株式会社ボイスリサーチ／西村 一彦　株式会社ボイスリサーチ

特集

SEC2013 年度活動概要

＜システムグループ＞
ソフトウェア障害情報の収集・分析及び対策／重要インフラ等システム障害対策（製品・制御システム）
重要インフラ等システム障害対策（IT サービス）／定量的プロジェクト管理による信頼性・生産性向上
コーディング作法ガイド：ESCR［C 言語版］Ver. 2.0 の発行／ソフトウェア・エンジニアリング成果の普及展開

＜ソフトウェアグループ＞
ソフトウェア信頼性の見える化／ソフトウェア品質説明力の強化の促進
品質説明力の強化に向けた「制度ガイドライン」の活用／先進的な設計・検証技術の適用事例
コンシューマデバイス機能安全規格化の提案のコンセプトと取り組み

海外レポート

米国 NIST、SEI、WVU を訪問して
IPA/SEC における国際連携の推進

事例紹介

宇宙システムにおける上流工程仕様の妥当性確認技術

片平 真史　独立行政法人　宇宙航空研究開発機構　情報・計算工学センター　安全・信頼性推進部 上席開発員 技術領域総括

組織の活動紹介

実践的情報教育協働ネットワーク：enPiT

春名 修介　大阪大学大学院情報科学研究科／楠本 真二　大阪大学大学院情報科学研究科
井上 克郎　大阪大学大学院情報科学研究科

報告

ソフトウェア工学分野の先導的研究支援事業について

Column

組込みソフトウェアの時代と日本の「ものづくり」

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

SEC journal No.37
Vol.10 No.2 Jul. 2014

普遍的な課題としてのシステム構築：ソフトウェアを超えて

独立行政法人科学技術振興機構（JST） 研究開発戦略センター（CRDS）
上席フェロー
木村　英紀

ソフトウェアはシステム構築と切っても切れない関係にある。ソフトウェアの世界で直面したシステム構築の難しさ、それを解くために培われた手法やノウハウを、一般的な社会システムの構築に展開していくべきと考える。

科学技術振興機構（JST）の研究開発戦略センター（CRDS）に、ナノテク、情報通信、ライフサイエンスなどと並んで、日本の科学技術の世界では聞きなれない「システム科学」を担当するユニットが誕生し、私がそのリーダを仰せつかったのは 2009 年 10 月である。着任した当時は JST の中でシステム科学について知見を持っていると思われる人は、このユニットの生みの親である CRDS の吉川弘之センター長を除けばゼロであったと言ってよい。「システム化」の重要性を説いてもけげんな顔をする人ばかりであった。もちろん「システム科学」やその関連領域がこれまでの JST の予算化の対象になったことは皆無に近く、「忘れられた科学」であった。我々のささやかなユニットはわけの分からないことを主張する異質のグループとみなされ、大げさに言えば四面楚歌の状態にあった。

それから 5 年近くの悪戦苦闘を経て状況はかなり改善された。2011 年に始まった第 4 期科学技術基本計画では「システム科学技術」が注力すべき融合分野の一つとして挙げられ、総合科学技術会議の「科学技術イノベーション総合戦略」ではシステム化がイノベーションを推進する研究開発の三本柱の一つと見なされた。システム化は技術の主要なトレンドの一つであり、それを支える科学の振興がとくに日本では必要であることはようやく産・官・学の共通認識になりつつある。しかし、それがどのような科学であり、どのような課題をどのように解決すべきであるかについては大方の賛同を得られる形で提示されているとは言い難い。

ソフトウェアの世界はシステム構築の手順が一応確立されている唯一の分野である。「一応」と書いたのはまだ完全ではないという意味であるが、それでも、作業の流れが明示的に示され、さまざまな開発手法や概念が提示され、形式手法を含む幾つかの学問的基盤も整備されつつある。既に長い間の実践を通じて成熟した分野といってよい。システム構築はソフトウェア開発の最上流に位置しており、そこではソフトウェアに限らない一般的なシステム構築の課題が現れる。IPA/SEC では既に大分前からソフトウェアとシステム構築の接点に注目していて、さまざまのプロジェクトを通じて多くの成果を上げている。私たちのユニットは、松本センター長の発案で外部の研究者も招き「システム科学検討会」を月一回のペースで開き、IPA/SEC の成果を勉強させていただいている。その中には要件定義や形式手法、アジャイル開発手法など、一般のシステムを構築する上で参考になる成果は多い。システムを社会に実装する上で隘路となっているのは、①システムがますます複雑化し、人間のマネジメントできる範囲を超えつつあること、②社会の変動が激しく、作られたシステムがすぐに陳腐化、劣化してしまうこと、③システムにかかわる利害関係者が拡大、多様化し、その調整のための共通手法が確立されていないこと、などである。これらはソフトウェアの場合にも共通する面が多いと思われる。私たちシステム屋はシステム構築の先達であるソフトウェア技術の経験と知見を学びつつ、これらの課題を解決する努力を続けて行きたい。

ソフトウェア産業の活性化と変革への道筋

芝浦工業大学大学院工学マネジメント研究科 教授
一般社団法人情報サービス産業協会（JISA）副会長
國井　秀子

×

SEC 所長
松本　隆明

IT が社会を支える時代になり、ソフトウェアの果たす役割は非常に大きくなっている。IT を使ったイノベーションにも期待が寄せられている。しかしソフトウェア開発の現場を見ると、まだ手工業的な面が残されており、そこに人が回せない、リソースが割けないという状況もある。そこで改めてそのあたりをひもとき、ソフトウェア産業を活性化し、イノベーションを生み出す活気ある現場とするためにはどうしたらいいのか、JISA 副会長で技術強化委員会の委員長もされている國井先生と一緒に考えていきたい。

松本：JISA は今年で 30 周年を迎えるとうかがいました。今とくに力を入れている取り組みはどういったものですか。

國井：私が委員長を仰せつかっている技術強化委員会のもとには「技術企画部会」「情報セキュリティ部会」「ソフトウェアエンジニアリング部会」「要求工学推進部会」「標準化部会」という５つの部会があります。各部会でいろいろな取り組みを進めていますが、「要求工学推進部会」ではちょうど『要求工学実践ガイド』を完成させたところです。ソフトウェア開発は上流工程が大事といわれながら、なかなか知識集約的になっていません。日本独特の「多重下請け構造」に根深い問題があるのではないかと思っていますが、とにかく要求工学の所をしっかりやっていかなければいけないということから、実践面を強く押し出して制作しました。今後技術強化と共に、構造的なところから変えていかなければならないと思っています。

また、今年とくにしっかり取り組みたいと思っているのは、新たなクラウド時代のシステム開発をどう進めるかということです。アジャイル開発手法やデブオプス（DevOps）に関する調査レポートを今編集中です。これは大きな期待をいただいています。変化の激しい今の時代では、やりながら色々考え、また、フィードバックを掛けて直していくということが必要です。開発と運用を連携して同時にやっていくことが重要になっている。従来のウォーターフォールでは厳しいですからね。

松本：アジャイルはソフトウェア開発の変革に向けてかなり重要なキーワードになりますね。ただこれは開発側だけでなく、利用者側つまり発注者側も巻き込んでいかないとうまくいかないのではないでしょうか。さきほどの「多重下請け構造」にも関連するかもしれませんが、発注者と一体になって開発し、仕様を決めていくということが非常に難しくなっている。これがいちばんの問題ではないかと感じています。

國井 秀子（くにい　ひでこ）

1970 年お茶の水女子大学理学部物理学科卒、テキサス大学コンピュータサイエンス学科にて Ph.D. 取得。
1982 年株式会社リコー入社以来 2008 年まで同社のソフトウェア分野の研究開発責任者。同社常務執行役員を経て 2008 年からリコー IT ソリューションズ株式会社取締役会長執行役員。
一般社団法人情報サービス産業協会副会長、日本学術会議連携会員、財務省関税・外国為替審議会委員、株式会社産業革新機構産業革新委員、内閣府男女共同参画推進連携会議議員、日本データベース学会副会長などを務める。

ソフトウェア開発の構造的な問題点

國井：「多重下請け」は、日本のソフトウェア開発の構造的な問題だと思います。情報システム側のエンジニア

だけでなくて会社のマネジメントをしている法学部や経済学部出身の人などに、ソフトウェアのパワーがどんなものなのかを知って欲しい。ソフトウェアの開発プロセスはハードウェアとは違います。ハードウェアは製造工程がかっちりしていて、製品開発の後の製造工程が非常に大事です。そして、できた物を具体的に示すことができる。しかし、ソフトウェアは説明してもピンと来づらいですね。中核となるアーキテクチャや技術は簡単には説明ができません。事例を色々見てもらうとか、技術をある程度学んでもらう必要があります。そこからやっていかないと、ソフトウェア業界だけでこの構造を変革できるかというと、それは無理だと感じます。

松本：ハードウェアはスペックが決まればあとは歩留まりをどう上げるかだけですね。物をお客様に納入してそれで終わりになる。ソフトウェアは設計以降の工程もお客様と一緒に考えていく必要がある。ここを発注側にどう理解してもらうかが重要です。

國井：とくに日本はモノカルチャーで“あうんの呼吸”の世界です。かつ、属人性が高い。しかし、どういうものを作るかということについては、ビジネスの人とエンジニアが一緒になって検討していかなければならないし、両側からお互いに分かることが必要です。コミュニケーション能力も過去と違って非常に高度なものが要求されていると思う。とくにビジネスリーダに、この点を理解してもらわなければいけないですね。

松本：単に開発の品質を上げ効率化していくためだけではなく、イノベーションを起こすという意味でも、ビジネスサイドに近い人が「あ、ソフトウェアを使えばこんなことができるのか」と分かってくれないといけないですね。このビジネス側と開発側の距離感を縮めていくことが大きな課題だと思います。

“減点主義”から脱却する

國井：イノベーションに関していうと、日本の企業はすごくかっちりとした工場的なマネジメントをしています。これは“行き過ぎた管理主義”といわざるを得ない。管理の形態をもうちょっと柔軟にすることが必要です。新しい世界というのは「こうすれば必ずこうなります」というのではない。トライアルしてみないと分かりません。最初に計画を立てて「計画通りできましたか？」「はい」「いいえ」とやっているのでは、つまり“減点主義”ではだめなのです。イノベーションは“加点主義”でないとできません。

ハードウェアの世界で、品質を守るためにきっちりした管理をするフェーズと、ソフトウェアを作りあげるために自由にトライアルするというフェーズでは、やり方が違うんです。柔軟に“加点主義”でトライアルできる体制や文化を企業の中に作りあげていかないと、イノベーションは起こらないと思います。

松本：そういう土壌もそうですし、ビジネス側もソフトウェアで何ができるか、IT で何ができるかということを勉強してもらいたいと思いますね。逆に IT の側も、様々な産業、事業の勉強をしなければならない。IT 側からの接近も必要なのかなと思います。

國井：そうですね。もはやかっちりした要求仕様があって、それを作っていきましょうというミッションではない。一緒に「そこはこうすればいいんじゃないか」と議論しながら作りあげていく。つまり両方から、技術者の方も業務を学んでいく必要があるし、ビジネスリーダも技術に関心をもって学ぶ必要がある。ところが日本では「それは技術の話だから」とどんどん現場に落としていって担当に任せてしまい、全体最適で判断できなくなってしまうという傾向があります。ここを変えていかなければいけませんね。

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高信頼化センター（SEC）所長。博士（工学）。

産学の連携をいかに進めるか

松本：JISA でも産学連携の重要性について議論されていると思いますが、学の成果をいかに産業界にフィードバックしていくか、その

正のスパイラルをうまく回していく必要があると思います。ここをどう進めればよいのか、お考えがあればぜひ伺いたいのですが。

國井：企業の中でも研究所と事業部の間には「死の谷」があるといわれます。同じ方向を向いている企業の中ですらそうですから、産学の連携はそう簡単ではありません。二つのことが考えられると思います。一つは一緒に密着してやっていくプロジェクトを増やすこと。もう一つは、人の流動性を高めることです。ドイツのフラウンホーファーなどでも、研究者と企業の技術者が一体になってプロジェクトを進めています。「開発しました、どうぞお使いください」ではないんですね。最初から学と産業界が連携するテーマを重視する形にもっていくというのも、ひとつの方法だと思います。

松本：企業の側から学に対して、こういう研究テーマで考えて欲しい、場合によっては一緒にやって欲しいという進め方をしたほうがいいのかなと思っています。

國井：産学連携がうまく進まない要因に、産業界の学に対するアクションが非常に弱いという問題もあると思います。過去、産業界の学に対する問題意識は、優秀な人材を提供してもらうことがメインで、一緒に何かしましょうというアプローチではなかった。大学がアカデミアに寄りすぎていて実践的な研究が少なかったということも災いしていると思います。産業界も期待していないからなおさら何も発信しない、という悪循環に陥っている。2004 年にパルミサーノのレポートが出て、米国の競争力はイノベーションだ。そのイノベーションを起こすためにはエコシステムが必要で、産官学が連携することと、異分野が一緒になってやること、そして多様な人材が必要だ、ということをいっています。正にそのエコシステムを作っていくためにも、産が学にもっと声をかけていかなければならないし、期待しなければ次のステップもないと思いますね。単純に点と点としてつながっているだけでは、新しいものは生まれない。

松本：確かに点と点で、そこがシステム化されていないから広げていくことができない、ということがありますね。企業の方も学に対して何を求めるかというところでギャップがあります。共同研究はやるのですが、おっしゃるように狙いは人の獲得ですね。

國井：日本は企業から大学に行く人は少しはいるんですが、大学から企業に行く人が少なくて、行っても研究分野で事業部には行きません。しかし事業部にも行かなければイノベーションはできない。ここを何とかしないといけないですね。

松本：インターンシップなどで学生さんに来てもらうこともありますが、それもやはり開発の現場ではないケースが多いようです。

國井：期間も短いですね。2 週間程度でしょう。私がリコーにいた時、ソフトウェア研究開発本部が海外からのインターン生を受け入れていましたが、ドイツのインターン生は 1 年、フランスでも半年という長期間でした。しかもマスターコースに組み込まれているので、単位がとれるんです。企業側からすれば新しい分野のことを研究してもらう、あるいは技術開発してもらう、ということでメリットがある。学生も色々経験できるし単位も取れる。大学は学生を企業に長く派遣して実践的に学ばせるということをゴールにしているのですね。日本でも大学を卒業した学部生も修士の学生も、場合によっては博士の学生も、多くは産業界に行くわけです。にもかかわらずもっぱら研究者を養成するために精力を使うというのはバランスが悪い。将来産業界に行くのなら、そこで研究ができる能力を付けることも重要ではないでしょうか。

評価にもっとリソースを投入する

松本：IPA では産と学の橋渡しができればと思っています。更に官としてやれることがあるのではないか、と考えているのですが。

國井：ドイツでは“インダストリ 4.0”ということが語られています。産業革命のインダストリ 1.0 に始まって、今はソフトウェアで新しい価値を作っていく「サイバーフィジカルシステムの時代」と位置づけ、ハード系の人とソフト系の人が融合して、次の世代の物作りを考えるということを進めている。産学連携の密度を高め、新たな方向性を見いだしてしていく、というところにフォーカスしているわけです。日本でも今、色々融合分野が育ちつつあるとは思いますが、まだ点と点ですね。そこをつなぐために、その音頭を取るところでは官が大きな役割を果たすことができるのではないかと思っています。それをつないでいくのは産業界には難しいからです。産

業界は自分の企業の利益も考えなければいけないですからね。例えば組込みソフトでも、ここは業界横断の統合的なプラットフォームを作っていかなければいけないという分野があったときに、一企業が手を挙げても「誰のためにやるんだ」と社内では抵抗がある。全体最適を図る意味でも、官のミッションとして進めることが必要だと思います。

松本：官にはそういう大きな方向性を出していくという役割がありますね。

國井：そこで大事なのは、評価に対する予算付けだと思います。評価にはきちんとリソースを投じなければいけない。アメリカの場合、NSF（国立科学財団）の色々な研究評価をする人として、例えばスタンフォード大学のある先生が数年間その任につくということになると、フルタイムでその間の経費の面倒を見るんです。評価委員として雇うわけですね。ところが日本では兼務が多い。私も委嘱を受けることがありますが、なかなか大変です。夏休みをすべてつぶしたりしてそれなりに時間は使っていますが、海外の評価委員と比較して私の評価はどうだろうと、申し訳なく思うことがあります。評価のためには最先端でなければいけないし、ニーズも知っていなければいけない。高い能力が求められ、時間も使います。こういうところに第一線の研究者は入りたがらないかもしれない。しかし、何らかのメリットを付与するといったことも考えていくべきだと思います。

松本：評価がきちんとできないから、どうしても効果があいまいになってしまって、なんのために金を付けているんだという話になってしまう。確かに、評価にリソースを投じるのは必要なことですね。

課題発見能力を高める訓練が必要

松本：人の育成という点について少し議論を進めたいのですが、産業界から見れば実践的な人材を育てて欲しいというニーズがあるでしょうし、学は学で研究者という形での育成の課題がある。そのあたりはどうお考えですか。

國井：私が思うのは、博士コースの勉強の仕方です。私はアメリカでドクターをとりました。アメリカではまずドクターの資格試験で、基本ができているかどうかをみる。そして次に研究テーマを自ら見つけるというフェーズがあります。その審査が終わって本格的な研究が始まる。この研究テーマを見つけるというところが、日本はおざなりなのではないかと思うんです。私がアメリカに行った時に、まず、問題発見にドクター論文に費やす時間の３分の１くらいは使わなければいけない、といわれたのを覚えています。

松本：３分の１ですか。それはウェイトが大きいですね。

國井：自ら問題発見するというプロセスがないと、企業に入った時に苦労するんです。そこがやれるかどうかで、ドクターを取ってきたレベルの研究者として通用するかどうか決まります。

松本：今の日本では指導する先生から研究テーマを与えられて、それを着実にこなしていくという状況になっていますね。

國井：問題発見のフェーズを持たないと研究における上流工程が貧弱になるんです。現場で受け身になってしまう。「先生」を探すことになってしまいます。リーダシップを取るためには、問題発見ができなければいけない。課題を見つけて、次にどうするか。研究の方向性を示せなければいけません。

松本：企業がドクターを採用するときも、その研究分野の取り組みや技術力を評価してというよりも、自分でテーマを考えて、どうやって進めるか、そういう“ドクター取得プロジェクト”みたいなものをやってきた、その能力を評価して、ということもあると思います。

國井：変化が激しくなっているので、今はとくにその能力が重要です。自ら方向を打ち出せるかどうか。ベースとなる技術力があり、課題発見能力があり、ターゲットのビジネスについて学んで新たなバリューを作っていこう、ということになれば、うまくまわっていくと思います。これはチームでもいいんです。それが重要だと気付いてパートナーシップが組めればいい。

その場合一番のポイントは「関心」です。それについて興味が持てれば勉強もするし話もする。何とか連携しようとすると思う。一番問題なのは「そこは私の世界ではありません」と、切り捨ててしまうことなんです。先ほどの話につながりますが、ドクターのコースできちんとやっていくということは、問題発見してどうするかという社会的な課題を見つけて、新しい価値をどう提供する

のか、とアプローチしていくということです。マーケティングの世界でも 1.0 は物志向だった。それが 2.0 で顧客志向になって、3.0 ではバリューをどう出すか、という追求に変わっています。その価値に関心を持ってくれる人がいないと困る。ドクターでそういうタイプの人がいれば企業は採用しますし、貢献できると思う。企業にとって価値を生んでくれる人材はいつも不足していますからね。

行き過ぎた管理主義

松本：IPA も「未踏プロジェクト」でイノベーションを起こせる若手の人材を育成しようとしています。実際「未踏」で出て来る人はかなりユニークで、色々なアイデアを持っている。ただそういう人をどう活かしていくか。企業が活かしきれていないという現実があります。

國井：そこに、先ほども触れましたが“行き過ぎた管理主義”の問題があるのですね。自由度が少なすぎる。枠の中に入るように教育してしまう。日本の組織マネジメントをもっと柔軟なものにしていくことが必要だと思います。人月計算であなたは何時間働いたからいくら払います、というのが浸透しきってしまって、トライアルしたくても、じゃあそれでいくら儲かるんだという話になってしまう。“とんがった”人材が管理の中にうまくフィットしない。おもしろくないから、結局モチベーションが上がらなくて辞めてしまうんです。

松本：裁量労働制を採り入れるところもあるようですが、研究開発部門だけですね。

國井：全社にはなかなか波及しないですね。ヘッドクォーターにはどうしても全社統一のルールでマネジメントしたいという意識がある。しかし、新しいことをやろうという部署と工場管理を同じルールでマネジメントするのは無理です。既存の業務についてはきっちりとマネジメントしても、新規分野は、あるリソースと期間の範囲内で自由にやってもらうという選択が必要です。事業のフェーズによってマネジメントのスタイルを変えていく必要があるんです。

松本：社内ベンチャー的にある程度認めてやっていくケースもあるでしょうね。「未踏」でも自分で自立してベンチャーを興してやっていこうとする人たちに、もっと支援ができるようにする仕組みに取り組んでいます。

國井：ベンチャーの場合は、技術は持っていても、経営的なところが弱かったり財務に関して疎かったりすれば破綻してしまう。そこをうまくサポートしないとベンチャーとして伸びない。チームワークがよくて、新しい価値を作るところと財務の連携がいいとか営業がバランスを取っているとか、トータルに企業として成功する要素を持っていないといけませんね。

松本：経営的な支援も必要ですし、営業的にはマーケットとのパイプもすぐには持てないでしょう。マッチングの場を提供するということもあるでしょうね。

國井：最初はイノベータータイプのお客様のところにいかなければ、新しいことはなかなか理解されません。理解してくれるお客様を紹介できる、あるいは多様な人が集うコミュニティがある、そういうことがインフラとして重要だと思います。

松本：これだけ IT が様々な場面で使われるようになって来ると、IT を分かる人間があらゆる事業分野に精通することは難しい。農業も分かる電力も分かる。そんなスーパーマンはいません。例えば農業の人と出会う場を作るといったことが、イノベーション創出のために必要になってきますね。

國井：少し先行しているニッチマーケットで色々トライアルできると、アプリケーション開発についてはやりやすいと思います。

松本：そこに官が実験的にお金を付けていく、という形で支援することも考えられますね。

國井：今はビジネスモデルが大きく変わっていて、ストレートでシンプルなものではなくなっている。技術だけではだめなんです。例えば検索サービスですごく技術力がある会社が広告収入で利益を出しているといったことが生まれています。バリューチェーン全体を把握して、総合的なビジネスについて考えられる人材を育成しなければいけない。これを情報工学のところでするのかといったら、やはりビジネススクールのほうで議論していただかなくてはいけない。そちらの人材育成についても投資をしていく必要があるのではないですか。

松本：先ほど関心を持つことが大切というお話がありましたが、正にそうですね。アンテナを自分で常に高く保って、広く関心をもつことが大事です。そこで IT で何ができるか、ソフトウェアで何ができるかと次に落として

考えていく、そういう癖を付けていかなければなりませんね。

女性管理職育成のプログラムを

松本：最後に女性の活躍というテーマについて考えてみたいのですが、ソフトウェア産業界の現状はいかがですか？

國井：JISA 会員企業に対しては、指導的立場の女性比率を 30%を目標に、女性が働きやすく、また、柔軟なワークスタイルを追求することを呼びかけています。それによって発想も豊かになるでしょうし、そもそも IT はそのサポートができるはずなんです。私たちが模範にならなければいけない。柔軟な働き方を追求し、その中で女性の活躍を進めたいと思っています。

しかしワークライフバランスの問題だけでなく、女性が働きにくい、あるいは活躍できない理由があるのです。色々調査を進めて分かってきたのは、仕事のアサインメントの仕方です。とくに育児休職をして戻った後などは、重要な仕事はアサインされない。ある時間内で高度なレベルの仕事ができると思っている女性は多いんです。ところが「心優しい管理職」が、大変だろうからというので細切れの仕事を与える。仮にそれがうまくいかなくても企業経営にはそれほど影響がない軽度の仕事を与えるわけです。当然チャレンジングな仕事ではないので、モチベーションが上がらないし能力も高まっていかない。

更に、エドガー・シャインが『キャリア・ダイナミックス』という本を書いていて、そこで「キャリアアップしていく３つの軸」として、専門性を高める、職能を広げる、小さな組織から徐々に大きな組織をマネジメントしていく、という３点を挙げています。職能を広げるというのは、研究開発をしていたら次はマーケティングをするとか、企画をするとか、そういうことですが、この点について、日本の場合は、女性にはほとんど配慮がされていないのですね。

キャリア・ダイナミックスに沿ってどう育成するか ── それをきちんと考えていかなければならない。私がかつて企業在職中に女性社員の上司にヒアリングしたときも、どうしてこの女性たちが管理職に就けないのかと聞いたら、経験がないとか、視野が狭いという返事が返ってきました。事実として確かにそれはある。しかし、ではその解決のために過去にどれだけのことがなされてきたのか。会社として、女性を管理職に育成するんだという流れがないのです。企業の側から、女性に期待している、こういうことをやって欲しい、ということを強く言って、仕事も与えていかないと、トライする気持ちも湧いてきません。

松本：ソフトウェア開発の仕事は非常にクリエイティブな世界で、女性の視点が役に立つのではないかと思いますし、もっと女性に活躍していただきたいですね。

國井：モノカルチャーではイノベーションは起きにくいわけです。世の中は、物からことへ、すなわち、サービス中心へとどんどん変わっています。女性の活躍する舞台はいっぱいあります。

松本：そうですね。今日お話をしてきて、ソフトウェアがイノベーションの源泉にならなければいけないという時代の中で、しかしその実現のために、まだまだ克服しなければならない多くの課題があることが改めて浮き彫りになったように思います。産学官、それぞれに大きなテーマがありますね。

國井：日本はハードウェアに関してはブランド力がありますが、ソフトウェアに関してはありません。トータルに国としてレベルを上げ、それを世界に示していくことが重要です。ソフトウェア開発を通じたイノベーション創出に向けて、国全体の最適化ということを考えたい。それにはやはりエコシステムの中でとらえていく、ということが重要だと思います。

松本：そうですね。IPA もそのための施策を考えていきたいと思います。本日はありがとうございました。

要件定義プロセスと保守プロセスにおけるモデル検査技術の開発現場への適用





松浦　佐江子[†1]　小形　真平[†2]　青木　善貴[†3]　谷沢　智史[†4]　西村　一彦[†4]

システム構築の上流工程における仕様の実現可能性や，保守工程における仕様及び仕様の誤解に起因する不具合現象に対し，形式検証技術であるモデル検査技術を一般的な開発者がそれぞれの工程において有効利用可能なシナリオを想定し，その支援方法を提案する．

Effective Model Checking Techniques Usage for Requirements Analysis and Maintenance Process

Saeko Matsuura[†1], Shinpei Ogata[†2], Yoshitaka Aoki[†3], Satoshi Yazawa[†4], Kazuhiko Nishimura[†4]

Abstract:　To verify the feasibility of the specification in the requirements analysis process of a system development and detect the defects caused by misunderstanding of the specification in the maintenance process, we propose an effective usage of model checking technique for non-specialized developers based on several reasonable development scenarios.

1.　はじめに

近年，システムの利用シナリオの多様性と広がりに加えて，ハードウェアやアーキテクチャの多様性が増加しており，手戻りを防ぎ，高品質なソフトウェアを開発するために，開発工程における検証プロセスの重要性が増している．モデル検査技術は，システムの振舞いを状態遷移システムとみなし，システムの満たすべき性質を，状態空間の探索により検証する技術である．テストでは実現できない網羅的検査に特徴があり，システム構築の上流工程において，その仕様の妥当性を検証するための形式検証技術として注目を集めている．しかし，実際に開発現場で用いるためには，開発現場での適用シナリオを想定して，検査対象システムのモデルとその検証したい性質を現場の開発者が容易かつ適切に定義できるようにすることが必要である．本研究では，要件定義プロセス，運用時の保守プロセスといった開発現場でのモデル検査技術利用のシナリオを想定し，それぞれの場面で，現場の技術者が利用可能な検証方法とその支援ツールを研究開発した．

2.　モデル検査技術適用の課題

モデル検査技術をシステムの振舞いの検証に適用するためには，一般に，振舞いを正確かつ少ない状態数と遷

【脚注】
† 1　芝浦工業大学　Shibaura Institute of Technology
† 2　信州大学　Shinsyu University
† 3　日本ユニシス株式会社　Nihon Unisys, Ltd.
† 4　（株）ボイスリサーチ　Voice Research Inc.

移数でモデル化し，そのモデルに対して検査したい性質を検査式として定義しなければならない．

VDM [VDM] や B-method [B-Method] のような形式仕様化技術は設計工程において，ドキュメントを形式化する有望な技術ではあるが，一般には，あいまいな要求から，要求抽出，要求定義を行う混沌とした要求分析工程において，はじめから厳密な形式手法を用いることは困難である．一方，UML（Unified Modeling Language）[UML] は要求仕様を記述するための自然言語を含む柔軟な記述を許す道具として広く多くの開発者に使用されている．クラス定義のように識別子を構造的に定義することは可能であるが，例えばアクティビティ図においても，アクションの記述や分岐の条件を表すガードの記述の自由度は高く，そのままでは形式仕様のように機械的な検証はできないという問題がある．

一方，ソースコードに対する不具合や仕様を満たしているかを検証する際には，大規模で複雑なソースコードをモデル化しても，状態爆発により検査ができないことがある．更に，検査したい性質の定義は一般にはソースコードの識別子と対応付けなければならず，ソースコードを仕様と対応付けて読み解くことは困難な作業であるという問題がある．

このように，開発工程に検証プロセスを導入するためには，ソフトウェア開発の上流工程では，初めから厳密な振舞いモデルを定義することが，下流工程では，ソースコードとして定義された振舞いと，その満たすべき性質である仕様を結びつけて，検証したい性質を定義することが難しい．

本研究では，我々がこれまで研究開発してきたUML を用いたモデル駆動要求分析手法 [Ogata2010a, Ogata2010b] 並びに，モデル検査ツールの 1 つである UPPAAL[UPPAAL] を用いたソースコードの欠陥抽出手法 [Aoki2011a,Aoki2011b] に基づき，ソフトウェア開発の上流と下流の両面から，開発者が想定した振舞い定義モデル（要求仕様書やソースコード）を特定の性質を満たす抽象的なシステムの振舞いモデルとして捉え，その特定の性質が取り得る状態の遷移モデルと結び付けることで，検査したい性質を表す論理式を自動生成する方法を研究する．

3. 開発現場での検査へのアプローチ

3.1. 業務セオリー

ソフトウェア開発工程には要求定義・設計・実装・テスト・運用の工程がある．これらの工程において，作りたいシステムを利用する際の作業手順，システムを利用してできることの条件，期待される状態，あってはならない状態，期待される効果，といった様々な形で，システムに対する要求が存在する．要件定義段階では，これらの全てを考慮するわけではないが，これらのレベルの異なる性質は最終的にはすべてのソースコードが満たさなければならない性質であり，違反することがあってはならない．そこで，本研究では，ソフトウェア開発工程で考慮される，ソフトウェアの満たすべき性質を「業務セオリー」と呼び，様々なシステムの性質の検証を開発現場で行えるように，整理することを目指す．例えば，個々のアプリケーション依存のセオリーだけでなく，ドメイン依存のセオリー，セキュリティ要件のセオリー，ソフトウェア構造依存のセオリー，ドメイン依存のフレームワークによるセオリー，ハードウェアアーキテクチャの性能制約によるセオリー，無限ループやデッドロック等プログラムで生じてはならない一般的なセオリー等の観点から検査したい性質の容易な定義方法を研究する．

3.2. 利用シナリオ

開発現場で検証技術を利用するためには，どのような場面であるならば，開発者が検査の実効性を享受できるかを考える必要がある．検証技術の利用が有効な場面を「シナリオ」と呼び．シナリオを実現する検査方法とその支援ツールを研究する．

要求をシステムの要件として定義する段階では，試行錯誤的に要求を確認しながらモデリングしなければならない．この段階で要求の妥当性を確認するには，我々のモデル駆動要求分析手法で行うように，要件定義から生成される最終形態を模したプロトタイプにより，利用者の観点からシステムの振舞いを確認することが有効である．しかし，要件定義は 1 つ 1 つのユースケースという機能的要件を部分的な観点から定義したものであり，入力から期待される出力のデータは本当に生成できるのか，ユースケースをつなげたトータルなサービスは，必要なデータを供給しながら，うまく実現できるのかといった要件の実現可能性や，非機能要件に関する妥当性や整合性を検査することは，人手では確認に手間がかかり，確認漏れや誤解が生じやすい．そこで，モデル検査を利用する第一の利用シナリオは「要件定義プロセスにおける要件定義の不整合の早期発見」とする．

テスト開発者は業務の知識を用いてテストを行なうことができるが，すべてのケースを網羅的にテストすることはできないため，実装者がその業務について十分理解

していない場合に，仕様の誤解によりテストでは発見できなかった不具合が運用時に生じることがある．このような場合，不具合現象は分かるが，ユーザからは不具合原因を特定する情報をあまり得ることができずに，検査者は経験に頼って，不具合の発見と修正を行わなければならないことが多い．こうした不具合は，特定するために，しらみつぶし的なテストを繰り返さなければならず，再現を保証することも経験の少ない検査者には困難である．こうした「運用時の想定外の使用による不具合の特定」を第二の利用シナリオとする．

3.3. 検査方式

業務セオリーとモデル検査による基本的な検査手順は図 1 に示す通りである．検査対象は各フェーズにおけるシステムの振舞いを定義したドキュメントである．上流工程のドキュメントに対し，開発工程が進むにつれ，システムの満たすべき性質である業務セオリーは開発者により具体化され，段階的にシステムの振舞いモデルに導入されていると考えられる．そこで，第一のシナリオでは，この導入作業において，検査対象の振舞いモデルと検査したい性質である業務セオリーの接点に着目する．一方，第二のシナリオでは，導入されているはずの業務セオリーを検査者が接点を確認しながら検査したい性質を定義する．この作業により，モデル検査の入力となるシステムモデル（有限オートマトン）と検査式を生成し，振舞いモデルが，検査したい性質である業務セオリーに違反することがないことをモデル検査ツールを用いて網羅的に検査する．振舞いモデルをシステムモデルに自動変換することで，反例の得られた個所を振舞いモデル上で特定できることから，これを修正する．

モデル検査を開発現場で用いるためには，モデル検査ツールを直接操作しないことが一つの解決策となる．「検査したい性質」を「検査対象の振舞いモデル」と対応付けて，開発者が理解しやすい形式で定義できるかということを解決しなければならない．また，モデル検査における状態爆発を回避できるように振舞いモデルの自動変換にも工夫が必要である．

第一の利用シナリオでは，モデル駆動要求分析手法による成果物である UML モデルを検査対象とする．2 節で述べた通り，UML は VDM 等の形式仕様記述言語と比べて，あいまい性はあるが，一般の開発者にも受け入れやすいという利点があり，開発現場への適用には向いていると考えられる．

本稿では，つぎの業務セオリーに着目し，データの基本的な性質に違反しないことを検査する．

- 要件定義の実現可能性の観点から，入力から出力を保証するエンティティ・データは，すべてのユースケースを継続して，複数のユーザに同等のサービスを提供できるように，その基本的な性質であるデータの CRUD（生成・参照・更新・削除）に関する振舞いに矛盾しない．例えば，まだどこでも生成されていないデータは参照・更新することはできない．このような性質は入力から出力を得る処理フローを分析している段階では，厳密に定義することが可能になっていると考えられる．

第二の利用シナリオでは Java のソースコードを検査対象とする．ソースコードはすべての要件が定義されているが，その表現が多様であり，アプリケーションとして満たすべき性質，使用しているハードウェアの制約，使用しているソフトウェアアーキテクチャの性質，言語特有の性質，プログラム一般の性質等，満たすべき性質，起きてはいけない現象は分かっていても，それをソースコードの識別子と対応付けることは困難である．本稿ではプログラム一般の性質として「停止しない」という不具合現象を業務セオリーとして，モデル化する．「停止しない」という現象は，個々のアプリケーションに依存しない振舞いであり，一般的にモデル化が可能である．

本稿では，それぞれの工程でのデータの振舞いの制約と現象の振舞いをモデル化し，二つのモデルの接点を段階的に定義・設定することで，接合したシステム（有限オートマトン）モデルを生成し，モデル検査によって，起こってはならない状態に到達しないことを検査するプロセスを開発現場でも利用できるように支援する．このために，モデル検査ツール UPPAAL を用いて，要件定義段階の支援ツール UML2UPPAAL と保守段階の支援ツール Source2UPPAAL を開発した．

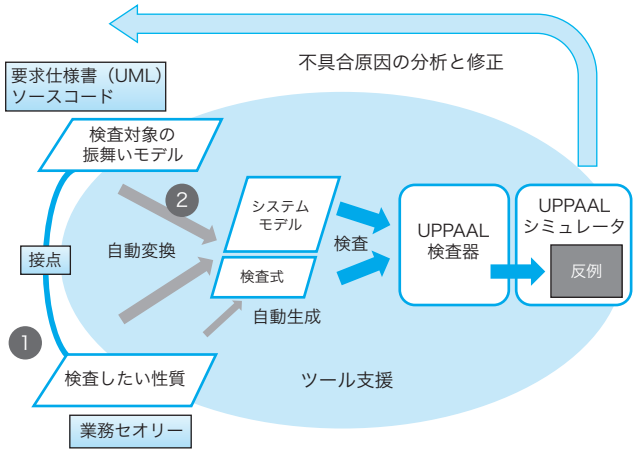


図 1　業務セオリーと検査方式

4. 要件定義プロセスにおける検査

4.1. UML2UPPAAL による検査プロセス

図 2 は，UML2UPPAAL を用いた要件定義プロセスにおける検査のプロセスを示している [Ogata2012,Aoki2012a,Aoki2012b]．要求分析モデルは，ユースケース分析に基づき，システムの提供するユースケースの振舞いを UML のアクティビティ図で，システムの利用するデータを UML のクラス図で定義したものであり，ユースケースの呼び出し関係もアクティビティ図で定義し，システム全体の振舞いを規定している．これを Navigation モデルと呼ぶ．アクティビティ図には振舞いの対象となるデータがオブジェクトノードとして定義されており，業務セオリーは，このデータの満たすべき性質をクラス毎に UML のステートマシン図で定義したものである．要求分析モデルは，各ノードと遷移に対応した UPPAAL モデル（有限オートマトン）に変換される．ステートマシン図は，すべてのアクティビティ図内で同一視されたオブジェクトノード毎に，状態と遷移に対応した UPPAAL モデルに変換される．このとき，アクティビティ図のアクションとステートマシン図のイベント，及びアクティビティ図のガードとステートマシン図のステートがこれらのモデル間の接点となり，UPPAAL のチャネルに変換され，モデル間の同期が定義される．検査式はステートマシン図において到達し得ない状態を解釈することにより，自動的に生成される．これらのモデルを UPPAAL モデル検査ツール上で実行した結果から検査式を満たしていないフローを UML 要求分析モデル上にフィードバックし，このフローと検査結果から要求分析モデルの問題点を発見して，モデルの修正を行う．

UML2UPPAAL は UML モデルの UPPAAL モデルへの変換，検査式の生成，検査の実行，反例のモデルへの反映の機能を持ち，UML モデリングツール astah*[astah] のプラグインとして実装されている．

4.2. CRUD に関する業務セオリー

要求分析モデルでは，登場するオブジェクトノードで表されるデータに対して，様々な処理を行って，期待される出力への変換手続きが定義される．ここで，処理が適用可能な最低条件は，そのデータが存在すること，すなわち，オブジェクトが「値あり」の状態になっていることである．存在しないデータに対しては何ら処理を行うことはできないため，対処方法を定めておかなければならない．そして，一般に，オブジェクトが「値なし」の状態から「値あり」の状態になるのは，そのオブジェクトの生成や取得の振舞いによってである．また，「値あり」のオブジェクトに対しては参照や更新，削除を行うことが可能である．このようにオブジェクトが適切に処理されるためには，その基本的なライフサイクルの性質である CRUD（Create・Read・Update・Delete）に関する普遍的な性質を満たすことが必要である．そこで，要求分析モデルのアクティビティ図が表す「入力データから出力データを生成する振る舞い系列」において，各アクションを CRUD 機能とその対象データによって整理することにより，開発者が定義した「実現するに足るシステム内部データ」が，要求分析モデルのすべての経路において，そのデータライフサイクルの性質を満たすことを検査することができる．

まず，アクションに記述する動詞を CRUD 機能と対応付けて定義する．これらの動詞は，アクティビティ図のシステムのパーティションにおいて，開発者が慣習的に使用する動詞を参考に決定した．例えばアクションの動詞が「取得する」場合，その行為は Read であると認識する．アクションノードに記述される「動詞」とその振

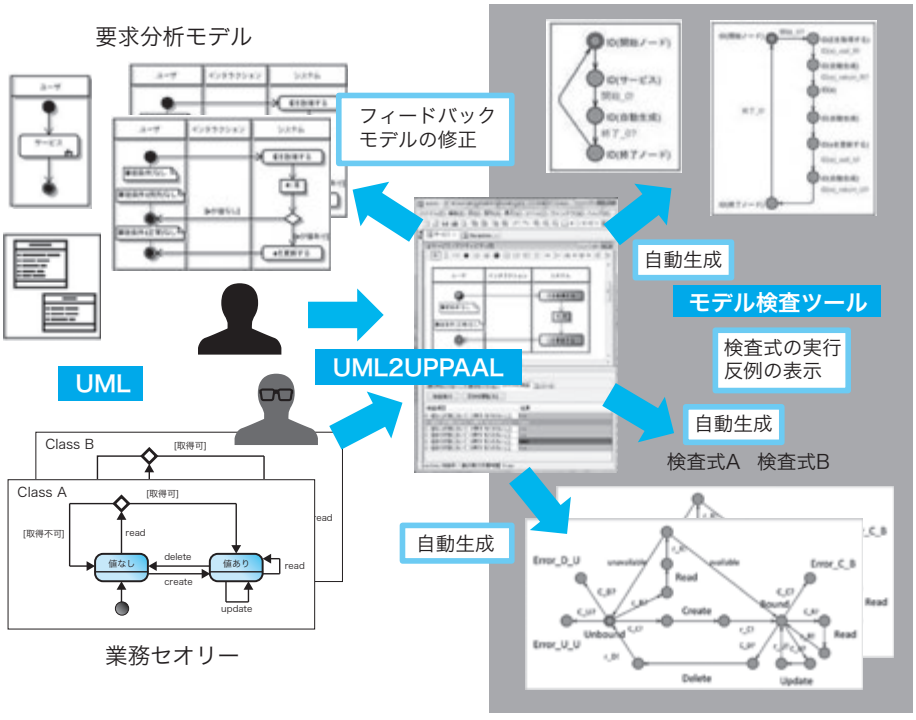


図 2　要件定義プロセスにおける検査プロセス

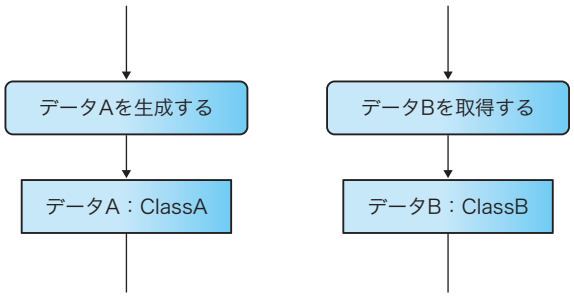


図３　Create・Read アクションの動詞とオブジェクトノードの関係

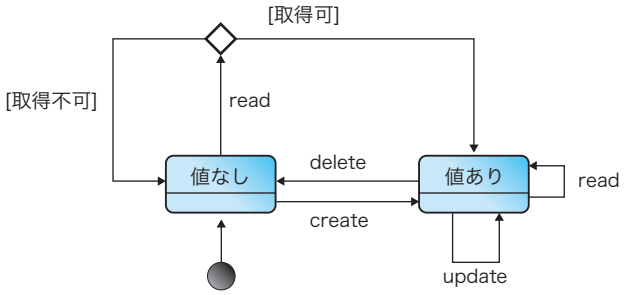


図４　クラスの CRUD に関する業務セオリー

る舞い対象である「目的語」と，それに対応する「オブジェクトノード」の位置から読み取れる意図を解釈する事で，振舞いにおける CRUD 機能の呼び出しが，図３のようにアクティビティ図に定義される．

これらの動詞により，アクティビティ図上のオブジェクトノードが「値なし」状態と，「値あり」の状態において，上記の CRUD の振舞いによって遷移可能なモデルを図４のようにステートマシン図を用いて定義する．これが CRUD に関する業務セオリーである．図４は一般的なライフサイクルであり，更新や削除を許可しないデータの場合には，これらの遷移を削除する．

ここで，重要なことは要求分析モデル内の定義要素と満たすべき性質を表すモデルの定義要素が，この段階において厳密に定義できる事柄に基づいて，対応づいたことである．これにより，要求分析モデルにおける性質を検査することが可能になる．

ステートマシン図は CRUD アクションに関するクラスのデータライフサイクルを定義しており，そのクラスに属するすべてのインスタンスの振舞いを限定するものである．言いかえると，この定義は「想定できない悪いことは決して起こってはならない」というモデル検査で検査できる性質の１つである「安全性」を示している．図４のステートマシン図は「インスタンスが値なしの状態であれば，そのインスタンスに対する Update 及び Delete の操作は決して起こってはならない」ことを定義しているわけである．そこで，この決して起こってはならない状態を UPPAAL モデル内に明示的にロケーションとして定義し，このロケーションを用いて「安全性」に関する検査式を自動的に生成する．ここで，ロケーションとは有限状態モデルである UPPAAL モデルの状態を表す要素の名称である．

4.3. 事例による検証

本手法の有効性を確認するために，適用実験を行った．本学で運用されているシステムや演習で作成されたシステム等４つのシステムの UML 要求分析モデルを５人の大学院生が，CRUD アクションの記述ルールを用いて整理した．次に，被験者はステートマシン図を用いて，要求分析モデルに登場するエンティティクラスのデータライフサイクルを各クラスの業務セオリーとして定義した．ここでは，図４に示す基本的な CRUD モデルを参考に，データ毎にカスタマイズしてモデルを作成した．被験者は UPPAAL の知識をほとんど持っていないが，UML2UPPAAL を用いて定義ミスの発見も含めて，83 個の問題点を発見することができた．適用実験で発見された主な問題点は次の通りである．

- 非決定的な Read アクションに対して，アクティビティ図内に正しくガードの条件を設定できていないケースが 10 個発見できた．これは，そのデータの想定される性質が不適切であるか，アクティビティ図のフローにおいて，データが取得できないケースを見逃していたことに起因する．
- Update と Delete 操作が「値なし」のオブジェクトに適用されていることを２個所で発見した．これは，振舞いモデルにおける「値なし」のケースの処理についての検討不足であった．
- Create 操作を定義する場所を間違えていたために．このサービスの間にはあるオブジェクトが全く生成できないという問題が１つ発見できた．このような問題が生じた１つの要因は，ユースケースの記述が複雑になり，目視のレビューでは発見が困難であったことであると考えられる．

事例による実験の結果，要求分析モデルを UPPAAL モデルに変換した結果，状態数が増加し，モデル検査が終了しないケースがあった．変換した UPPAAL モデルの状態数を抑えるため，モデル定義に工夫が必要であることが分かった．要求分析モデルでは，複数のユースケースを Navigation モデルにより統合する．この統合において，繰り返し処理を行うと，呼び出される実行モデル

が大きい場合，ロケーションへの到達可能性の検査において，out of memory になる可能性がある．そこで，Navigation モデルでは繰り返しが起こらないようにモデルを定義することで状態数の削減を行うことができた．CRUD の振舞いの検査を行うには，各ユースケースのつながりが分かれば良いので，検査上は問題ないが，要求仕様として考えた際には，自然な記述ではなくなる可能性があり，要求分析モデルの定義と検査の役割について検討が必要である．

5. 保守プロセスにおける検査

5.1. Source2UPPAAL による検査プロセス

図 5 は，Source2UPPAAL を用いた保守プロセスにおける検査のプロセスを示している．[Yazawa2012, Aoki2012c]

ソースコードは Source2UPPAAL によりソースコードモデルへ変換される．また仕様及び不具合の現象から検査者は独自のモデルを定義する．ソースコードモデルと独自モデルは Source2UPPAAL 上で，検査者の指示により結合されてシステムモデルとなり，生成された検査式により，モデル検査が実行される．開発者は UPPAAL の反例からソースコードの不具合原因を分析する．

ソースコードは AST パーサーにより抽象構文木（AST）に変換される．Source2UPPAAL が抽象構文木に基づきモデル化候補となるメソッドを表示するので，変換対象のメソッドをドラッグ＆ドロップ操作で選択することで，モデルの状態数を制限して検査する．選択されたメソッド定義において，ステートメントは UPPAAL モデルのロケーションに，整数と真偽値の変数は UPPAAL の変数に，変数への代入は UPPAAL の更新式に変換される．メソッドや API の呼び出しに対しては，図 6 に示すように，検査者が必要に応じて非決定性を与える値を提示されるパターンの候補から割り当てる．また，検査者が UPPAAL モデルとして定義した独自モデルをツールに登録することでメソッドや式にその振舞いを割り当てることができる．Source2UPPAAL はこのようなソースコードのモデル化個所の指定や，非決定性や独自モデルの割り当てのためのドラッグ＆ドロップ方式のユーザインターフェースを提供する．そして，検査者の指定が終了すると，UPPAAL モデルと検査式を自動生成し，検査を実行する．

5.2. 事例による検証

下記の事例を用いて，Source2UPPAAL により，モデル検査で検査できる性質の 1 つである「到達可能性」と「停止しない」という，起こってはならない不具合現象の検査を行い，その原因を発見することができた．

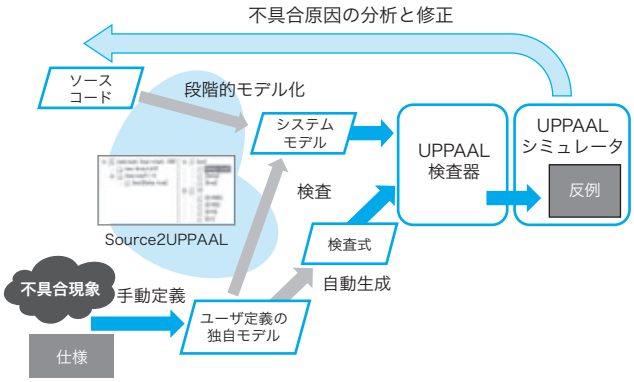


図 5　保守プロセスにおける検査プロセス

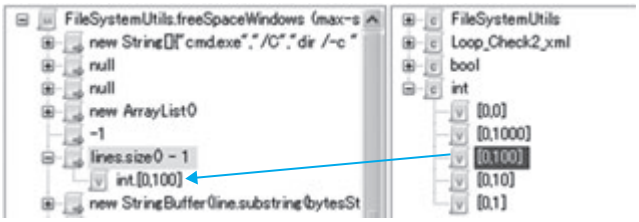


図 6　非決定性の値の割り当て

1）規定の走行体を用いて，難所を含むコースのライントレース自律走行を競う ET ロボコンにて使用したロボットの走行制御プログラムにおける制御フローの正当性の確認

2）オープンソース Apache Commons の既知のバグ（無限ループ）

3）会計伝票発行における停止しない現象（無限ループ）

1）の事例では，制御フローの正当性を確認するために，すべての制御フローがすべて通りえる状態であるかを到達可能性により検査する．検査式は，生成されたすべてのロケーションへの到達の検査であり，「ロケーション ** にいつかは到達する」という意味の検査式を自動的に生成する．検査結果が「満たされない」の場合は，そのロケーションへの到達はできないことになり，その反例がトレースファイルとして記録される．これを分析した結果，到達できないソースコードへの分岐条件式に問題があることがわかった．その条件式に登場する変数の変化をトレースファイルから確認し，その値を計算している部分を特定して，修正を行い，再度検査をして，すべての検査式が満たされることを確認した．

2）及び 3）の事例では，「停止しない」という観測できる状態に対して，無限ループのモデルを独自モデルとして定義し，制御の条件式に対して設定することで，無

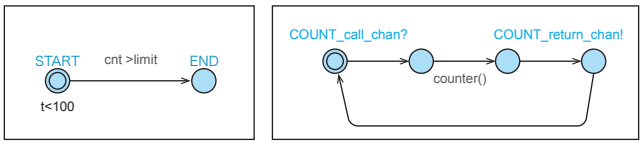


図 7　無限ループ判定用モデル

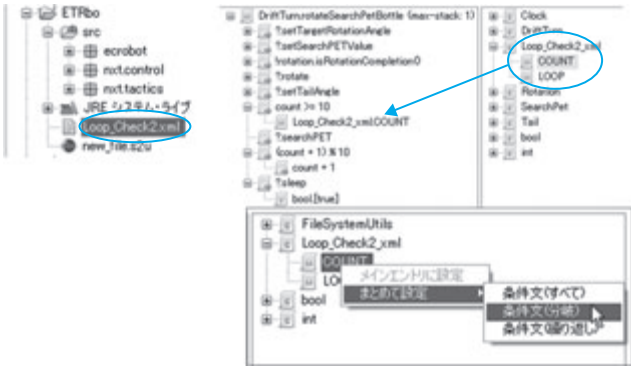


図 8　独自モデルの割り当て

限ループの発生に起因するこの現象を検査する．検査式は無限ループ状態のロケーションへの到達の検査であり，これも自動的に生成することができる．

この検査は，制御構造の未到達を検査するのではなく，既知の不具合の現象をモデル化して，それにより不具合の原因の特定を行うものである．ここでの「不具合の現象」は無限ループであるため，無限ループ判定用のモデルを定義する．これはすべてのプログラムが犯してはならない 1 つの業務セオリーである． 無限ループの判定モデルは図 7（左）のように定義し，閾値（limit）を決めておき，それを超えた場合に無限ループと判定する．無限ループは条件分岐で抜けられない可能性が高いため，条件分岐している部分にループをカウントするための関数 counter() を配置する．counter() を呼び出すモデルは図 7（右）のように定義する．ここでは，このモデルは UPPAAL を直接使用して作成している．

定義した無限ループ判定用のモデルをつぎのようにツールに追加する．まず，ツール上で検査対象と同じプロジェクト内に追加モデルのファイルを配置すると，図 8 のように表示される．ここでの追加モデルのファイルは図 8 の四角形で囲われた Loop_Check.xml である．ここでは，無限ループの発生を検査するため，条件文の式に対して，まとめて独自モデルを割り当てている．検査式は「図 7（左）のモデルのロケーション END に到達することは決してない」という安全性の検査式が自動生成される．検査式が満たされなかった場合，反例が示され，遷移の過程がトレースファイルに記録される．このトレースファイルを用いて無限ループの発生状態の分析を行う．全ロケーションの出現頻度をグラフで表すことで，その頻度が高い個所から，ループしているロケーションが特定でき，これにより，対応するループしているソースコードが特定できる．その部分を含むループ構造の条件式または break によるループの脱出に至る条件式のどちらかに問題があるので，これらに登場する変数の変化をトレースファイルから確認し，修正を行い，再度検査をして，すべての検査式が満たされることを確認した．

現実のソースコードは複雑であり，これをすべて UPPAAL モデルに変換しても，現実的な検査は行えない．本研究でのアプローチの特徴は，アプリケーションコードに依存しない，不具合現象をモデル化し，開発者が，関連するソースコードのメソッドを段階的に，非決定性を持つ値を想定しながら，検査できる機能を持つ支援ツールを提供していることである．到達可能性と無限ループ以外にも，ライントレースロボットのコース逸脱現象やデータベースロックの不具合といった不具合現象を検出できることが分かっているが，例えば前者ではロボットの走行環境であるコースのモデル化が，後者では，システムが利用している特定言語のモジュール拡張機能の仕組みを独自にモデル化する必要がある．

6.　関連研究

モデル検査ツールの効果的な利用方法に関する研究がある [Tracka2009, Bose1999, Jing2009]．Trcka[Tracka 2009] はペトリネットを利用して，read, write, delete といった振舞いを特定する予め用意した 9 つの検査式を用いて検査を行う方法を提案している．この研究も我々のアプローチと類似しているが，我々の方法ではステートマシン図を用いて，データの性質の着目点を限定して容易に定義することのみによって検査式を自動生成できるため，固定した CRUD だけのルールではなく，他の性質への拡張が可能である．[Bose1999, Jing2009] の研究では，UML モデルを PROMELA へ変換し，SPIN[SPIN] を利用して，モデル検査を行う方法を提案している．しかし，検査を行うためには，一般の開発者が直接モデル検査ツールを操作する必要があり，開発者は UML と SPIN の両方の知識を要求される．UML2UPPAAL は UML の知識だけで検査を実行することができる．

ソースコードの不具合検出にモデル検査を使用する研究は多い．Java Pathfinder[Pathfinder] は実行可能な Java バイトコードよりプログラムを検証してデッドロックとアサーションエラーを検出する．Pathfinder は複雑

で大規模なソースコードに対しては「状態爆発」の注意が特に必要である．

Bandera[Corbett2000] は．抽象化とスライシングにより，Java プログラムのソースコードから，コンパクトな有限状態モデルの自動抽出が可能である．出力モデルは，SPIN や SMV[McMillan1993] 等の既存のモデル検査ツールで検証することができる．このアプローチは，「状態爆発」の抑制には有効であるが，検査者にはモデル検査の深い知識が必要となる．

7. まとめと今後の課題

モデル検査を実施する上で，第一に克服しなければならない課題は，検査対象の定義と検査したい性質の定義をその構成要素間で容易に対応付けることができることである．本研究では，要求仕様とソースコードに着目し，段階的に検査対象と検査したい性質としての業務セオリーを結び付けることにより，要求定義段階では，モデル検査技術の知識がなくても検査が可能な UML2UPPAAL を実現した．保守段階でも，少ないモデル検査技術の知識で，検査を支援する Source2UPPAAL を開発し，開発現場での検証技術の利用が有効な場面であるシナリオを実現する検査方法と支援ツールを提案した．

要件定義プロセスにおける要件定義の不整合の早期発見に関しては，UML2UPPAAL として，開発者がモデル検査技術の知識を持たなくても，網羅的な検査を実施できることがわかった．しかし，コレクションとその要素に関する振舞い，オブジェクトとその属性であるオブジェクトの連動等の定義方法は検討中である．要求分析モデルを段階的に形式化することの見通しはあるが，定義方式が複雑にならないよう，検討する必要がある．

今後は，本手法を情報システムのセキュリティ保証要件を定めた国際標準（ISO/IEC 15408）である Common Criteria [Common Criteria] に基づき，要求仕様がセキュリティ機能要件を満たすことを検査すること [Noro2013] に応用する．

運用時の想定外の使用による不具合の特定に関しては，無限ループ以外の不具合現象例のモデル化を要件定義プロセスと同様に仕様をステートマシン図で定義し，検査式を自動生成することを検討する．また，検査によって出力される反例の解析により，修正を支援する必要もある．成功事例と失敗事例の対比により，不具合を特定する方法を検討する．

謝辞

本研究は，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（SEC: Software Reliability Enhancement Center）が実施した「2012 年度ソフトウェア工学分野の先導的研究支援事業」の支援を受けたものである．

【参考文献】

[VDM] VDMTools, http://www.vdmtools.jp/

[B-Method] K. Lano, H. Haughton: Specification in B: An Introduction Using the B Toolkit, Imperial College Press, 1996.

[UML] OMG Unified Modeling Language, OMG, http://www.uml.org/

[Ogata2010a] Shinpei Ogata, Saeko Matsuura, Evaluation of a Use-Case-Driven Requirements Analysis Tool Employing Web UI Prototype Generation, WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, Issue 2, Volume 7, pp.273-282, February 2010.

[Ogata2010b] 小形，松浦：UML 要求分析モデルからの段階的な Web UI プロトタイプ自動生成手法，日本ソフトウェア科学会，コンピュータソフトウェア，Vol.27，No.2，pp.14-32，2010.

[UPPAAL] UPPAAL, http://www.uppaal.com/

[Aoki2011a] Y. Aoki and S. Matsuura, Verification of Embedded System by a Method for Detecting Defects in Source Codes Using Model Checking , IEEE Symposium on Computers & Informatics, pp. 530-535, 2011.

[Aoki2011b] Y. Aoki and S. Matsuura, A Method for Detecting Defects in Source Codes Using Model Checking Techniques, Proc of the 34th Annual IEEE International Computer Software and Applications Conference, pp.543-544, 2010.

[astah] astah*, http://www.change-vision.com/

[Tracka2009] N. Trcka, et al., “Data-Flow Anti-Patterns: Discovering Dataflow Errors in Workflows,” Proc. of the CAiSE 2009, pp.425-439 Amsterdam, The Netherlands, 2009.

[Bose1999] P. Bose, “Automated translation of UML models of architectures for verification and simulation using SPIN,” Proc. of the ASE, pp.102-109, Fairfax, VA, 1999.

[Jing2009] L. Jing, et al. “Model Checking UML Activity Diagrams with SPIN,” Proc. of the CiSE 2009, pp.1-4, Qingdao, China, 2009.

[Pathfinder] Pathfinder, http://javapathfinder.sourceforge.net/, 2013

[Corbett2000] Corbett, J., Dwyer, M., Hatcliff, J., Laubach, S., Pasareanu, C., Robby, and Zheng, H., "Bandera: extracting Finite-state models from Java source code," Proc. the 22nd Int'l Conf. on Softw. Eng. (ICSE 2000), pp. 439–448, 2000

[SPIN] SPIN, http://spinroot.com/spin/whatispin.html, 2013

[Ogata2012] S. Ogata, Y. Aoki, H. Okuda and S. Matsuura, An Automation of Check Focusing on CRUD for Requirements Analysis Model in UML, Proc of ICSCE 2012, pp.1095-1103, 2012.

[Aoki2012a] Y. Aoki, S. Ogata, H. Okuda and S. Matsuura, Quality Improvement of Requirements Specification Using Model Checking Technique, Proc of ICEIS 2012, Vol.2,pp401-406, 2012.

[Aoki2012b] 青木，小形，奥田，松浦，要求分析における CRUD 観点のモデル検査技術の適用，ソフトウェア工学の基礎 XIX，日本ソフトウェア科学会 FOSE 2012，pp. 75-80.

[Yazawa2012] 谷沢，西村，青木，小形，松浦，Source2UPPAAL: ソースコードの効率的な検証へ向けた開発者支援ツールの検討，ソフトウェア工学の基礎 XIX，日本ソフトウェア科学会 FOSE 2012，pp. 241-242，2012.

[Aoki2012c] 青木，松浦，開発現場を想定したモデル検査に基づくプログラムの不具合検証，電子情報通信学会 , 信学技報 , vol.112,no.496, KBSE2012-69, pp. 1-6, 2013.

[McMillan1993]K. L. McMillan: Symbolic Model Checking. Kluwer Academic Publishers, 1993

[Common Criteria] Common Criteria, CC/CEM v3.1Release4, http://www.commoncriteriaportal.org/cc/

[Noro2013] A. Noro and S. Matsuura, UML based Security Function Policy Verification Method for Requirements Specification, Proc of 2013 IEEE 37th International Conference on Computer Software and Applications, pp.832-833,2013.

SEC2013年度活動概要

SEC 副所長 杉浦　秀明　SEC 次長 杉原井　康男　SEC 企画グループリーダー 石川　智　SEC 企画グループ主幹 江野村　亮輔

2013 年度は、IPA 第三期中期計画（2013 年度～ 2017 年度）の初年度として、中期計画で掲げた「重要インフラ分野の情報処理システムに係るソフトウェア障害情報の収集・分析及び対策」、「利用者視点でのソフトウェア信頼性の見える化の促進」、「ソフトウェアの信頼性に関する海外有力機関との国際連携」の３つの事業の目標達成に向けて、着実に活動を進めてきた。本稿では、2013 年度の主要な成果概要を紹介し、本稿以降で詳しい事業内容を紹介する。

1 重要インフラ分野の情報処理システムに係るソフトウェア障害情報の収集・分析及び対策

（1）システム障害情報事例を収集し、教訓集として取りまとめ

SEC では、国民生活や社会・経済基盤を支える重要インフラ分野等における情報処理システムの信頼性向上のため、システムの障害事例情報の分析や対策手法の整理・体系化を通して得られる「教訓」を業界・分野を越えて幅広く共有し、類似障害の再発防止や影響範囲縮小につなげる仕組みの構築に着手した。

2013 年度は、一定の機密保持ルールのもとに重要インフラ分野等の企業からの情報提供や有識者・専門家からのヒアリング等により、27 件（製品・制御システム※1 18 件、IT サービス※2 9 件）の障害事例を収集するとともに、これまでの産学官の連携のもとに蓄積されたソフトウェア・エンジニアリングの幅広い知見を基礎として、収集した障害事例情報の分析と対策の検討を行い、それらを教訓として一般化・抽象化して、「情報処理システム高信頼化教訓集」として取りまとめた。併せて、先進的企業等の取組み事例を収集し、「障害分析手法・事例集」、「障害対策手法・事例集」として取りまとめた。

また、障害事例情報を収集・分析し、社会で共有する仕組みの構築に向け、障害事例ヒアリング、共有グループでの原因分析／対策検討、及び教訓の公開時において必要な、障害情報を記録する共通様式の設計、障害情報提供に関する機密保持・情報提供ルールを作成した。

2 利用者視点でのソフトウェア信頼性の見える化の促進

（1）「ソフトウェア品質説明のための制度ガイドライン」の公開・普及

製品・システムの品質を第三者が確認する制度を設ける際の要求事項等を、公正性・整合性確保の観点からまとめた「製品・システムにおけるソフトウェアの信頼性・安全性等に関する品質説明力強化のための制度構築ガイドライン（通称：「ソフトウェア品質説明のための制度ガイドライン」）（以下、「制度ガイドライン」）を公開した。さらに制度ガイドラインの適用第一号として、2012 年度から SEC の実証実験プロジェクトチームにて実証確認を進めてきた CSAJ※3 が PSQ※4 認証制度の運用を開始した。

また、製品・サービス等の異なる 25 の業界団体・機関等に対して、制度ガイドラインの紹介と、制度構築に対するニーズのヒアリングを実施したところ、既に継続して連携している CSAJ に加え、新たに 3 団体（IIOT※5、DEOS 協会※6、SVA※7）と制度化に向けた検討体制を構築し、SEC は 3 団体との制度検討に積極的に参画し、制度ガイドラインに基づいた制度設計の具体化に貢献した。

（2）ソフトウェア開発におけるサプライチェーンの業界横断的な課題の抽出に着手

新たな取組みとして、ソフトウェア開発におけるサプライチェーンに係る課題を把握するため、製品・サービス等の異なる 20 の業界団体に加えて、組込み系、エンタプライズ系、クラウドサービス基盤系、モバイルサービス系の計 12 企業を対象にヒアリングを実施した。さらに、ヒアリングにおける課題を詳細化するために、「ソフトウェア開発の取引構造（サプライチェーン）の実態に関わる課題の調査」

【脚注】

※1 重要インフラ分野等の情報処理システムのうち、組込みシステム（製品において各種センサなどを用いて制御を行っているシステム）に関する事例を「製品・制御システム」として分類。

※2 重要インフラ分野等の情報処理システムのうち、エンタプライズ系システムや IT を利用し、または提供して行っているサービスに関する事例を「IT サービス」として分類。

※3 CSAJ(Computer Software Association of Japan)：(一社) コンピュータソフトウェア協会。

※4 PSQ(Packaged Software Quality)

※5 IIOT(international internet of things international interoperability testing)：（一社）IIOT。

※6 DEOS(Dependable Embedded Operating Systems) 協会：（一社）ディペンダビリティ技術推進協会。

※7 SVA(Smart System Verification and Validation Technology Association)：（一社）スマートシステム検証技術協会。

を実施し、その結果から新たな課題を抽出し、2014 年度以降の取組みの方向性として整理した。

(3) 先進的な設計手法・信頼性検証手法・技術等の取組み事例を収集し、適用事例報告書として取りまとめ

複雑化・高度化する情報処理システムを実現するソフトウェアについて、その高信頼性を確保するため、SEC では、先進的な取組みを実施している企業・団体・大学から、設計手法・信頼性検証手法・技術等の取組み事例を 24 件収集し、「先進的な設計・検証技術の適用事例報告書」として取りまとめた。

(4) 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※8) を改訂

近年、開発現場で使用されることの多くなった C 言語の最新 JIS 規格「C99」に適応させるべく、ESCR［C 言語版］を改訂し、Ver.2.0 として 2014 年 3 月に発行した。今回の改訂では、C 言語の最新 JIS 規格（C99）に準拠し、新機能などに対応したほか、ESCR と相互に引用を行っている、欧州組込み業界標準規格の「MISRA C※9」（MISRA C:2012）との整合を図った。

3 SEC 成果の国際的情報発信、国際連携

(1) 日本が得意とする“すり合わせ開発”とも融和性の高い開発方法論の国際規格を産学官連携で提案

自動車、サービスロボット、スマートハウス、スマート家電等、一般消費者が使用する、組込みシステムにより高機能化された機器である「コンシューマデバイス」の高い安全性・信頼性を実現するため、日本が得意とする“すり合わせ開発”とも融和性の高い開発方法論の国際規格を産学官連携で OMG※10 に提案した。

(2) IT プロジェクトベンチマーキング・プロセス評価の SEC 成果に基づく国際規格が 2 件発行

IT プロジェクトベンチマーキング・プロセス評価等の SEC 成果の国際標準化活動を推進し、2 件の国際規格が発行された。本発行に際し、それぞれの活動に SEC から 2 名エディタとして参画しており、(一社) 情報処理学会からも貢献が認められ表彰された。我が国の企業にとって馴染みの深い手法が国際標準になることで、中小企業等の海外進出や日本と同等品質の海外オフショア開発実現等の一助として、我が国産業の国際競争力向上が期待される。

(3) 海外有力機関との関係強化

これまで連携をしている海外代表的機関の米国 NIST※11、米国 SEI※12、独国 IESE※13、英国 MISRA に加え、米国 NASA IV&V Facility※14、米国 WVU※15、蘭国 TNO-ESI※16、韓国 NIPA※17 との関係を構築した。

NIST とは第 4 回定期協議をワシントンで開催し、今回は特に「自動車の自動運転やロボット」の安全性に関する指標に関する意見交換を行うとともに、ソフトウェア信頼性指標等の取組み状況についても有用な情報を得た。

SEI とは両機関の連携の一環として、「IPA グローバルシンポジウム 2013」(2013 年 5 月開催) に SEI 所長を講演者として招聘するとともに、2014 年 3 月に SEI を訪問し、共通するテーマであるソフトウェアサプライチェーン等に関する情報交換を実施した。

IESE とは 2014 年 1 月に SEC にて意見交換を実施し、SEC の先進的な設計手法・信頼性検証手法等技術の適用事例収集に関して、今後の協力を依頼した。

MISRA に 2014 年 3 月に訪問し、SEC の ESCR［C 言語版］の改訂ポイントや今後進める ESCR［C++ 言語版］の改訂について説明し、MISRA からも C++ 言語版のコーディング規約（プログラミング・ルール）改訂に関する検討状況の説明を受け、今後の協力関係について議論した。

WVU に 2014 年 3 月に訪問し、NASA IV&V Facility も交えて IV&V※18 の最近の取組み内容や、民間企業との共同研究などの最新動向について意見交換を実施した。

TNO-ESI に 2014 年 3 月に SEC として初めて訪問し、両機関の取組みのモデルベースによるソフトウェア分析と検証について意見交換を実施した。

NIPA とは 2013 年 8 月に SEC において両機関の取組み状況を共有し、ソフトウェア品質説明力強化の取組みやプロセス改善活動について意見交換を実施した。

次頁からは、これらの内容について詳しく紹介する。

【脚注】

※ 8 ESCR(Embedded System development Coding Reference)。

※ 9 MISRA C：英国 MISRA（The Motor Industry Software Reliability Association）が作成した C 言語のためのソフトウェア開発標準規格。MISRA は自動車メーカ、部品メーカ、研究者からなる欧州の自動車業界団体。

※ 10 OMG(Object Management Group)：国際的な標準化団体、本部は米国マサチューセッツ州。

※ 11 NIST(National Institute of Standards and Technology)：米国商務省国立標準技術研究所。

※ 12 SEI(Software Engineering Institute)：カーネギーメロン大学ソフトウェアエンジニアリング研究所。

※ 13 IESE(Institute for Experimental Software Engineering)：フラウンホーファ協会実験的ソフトウェア工学研究所

※ 14 NASA IV&V Facility (National Aeronautics and Space Administration Independent Verification and Validation Facility) とは、米国航空宇宙局（NASA）の宇宙機ソフトウェア独立評価機関（IV&V Facility）。運用・整備業務を WVU が行っている。

※ 15 WVU(West Virginia University)：ウエスト・バージニア大学。

※ 16 TNO-ESI(Netherlands Organization for Applied Scientific Research-Embedded Systems Innovation)：応用科学研究機構組込みシステムイノベーション。TNO-ESI は 2002 年にオランダ政府のファンドと民間等のファンドにより設立され、2013 年 1 月より TNO 傘下の組織。

※ 17 NIPA(National IT Industry Promotion Agency)：韓国の政府機関である情報通信産業振興院。

※ 18 IV&V（Independent Verification & Validation）とは開発組織やその委託組織から独立した組織が高度なソフトウェアの信頼性を確保するため、正しい仕様のソフトウェア (Validation) が正しく動作すること（Verification）を、客観的に評価する活動又は組織。

システムグループ

ソフトウェア障害情報の収集・分析及び対策

SEC システムグループ リーダー
山下　博之

国民生活や社会・経済基盤を支える重要インフラ分野等における情報処理システムの信頼性向上のため、システムの障害事例情報の分析や対策手法の整理・体系化を通して得られる「教訓」を業界・分野を超えて幅広く共有し、類似障害の再発防止や影響範囲縮小に繋げる仕組み（図 1 参照）の構築を目指した取り組みを、平成 25 年度より開始した。

まず、重要インフラ分野等の企業からの情報提供や有識者からのヒアリング等により障害事例情報を収集した。並行して、重要インフラ分野等の有識者・専門家の委員を中心とする委員会を設置し、所定の機密保持ルールの下で、収集情報の分析と対策の検討を行い、それらを教訓として一般化・抽象化した。これらの教訓を多方面の業界・分野の視点から普遍化し、「情報処理システム高信頼化教訓集－ 2013 年度版－」として取りまとめた。（図 2 参照）また、障害事例情報収集、委員会での原因分析／対策検討、及び教訓の公開時において必要な、障害情報の記録様式、その機密保持・開示条件等のルールを作成した。

これらの活動は、システムの構築や運用の形態が大きく異なることから、重要インフラ分野等における情報処理システムを構成する『製品・制御システム』と、重要インフラ分野等の『IT サービス』そのものとの 2 種に分けて実施した。以降では、それぞれの活動内容について具体的に説明する。

今後、上記教訓集の活用に向けた普及展開活動を行うと共に、障害事例の分析に基づく教訓の共有の仕組みを幅広い業界・分野に展開していく予定である。

最後に、本テーマに関するセミナーの受講者アンケート結果を紹介しておく。

- 88％が「本取り組みは有用」と回答し、その 67％が「成果は自社に役立つ」と回答。
- 87％が「基準以上の障害事例に基づく教訓は、広く社会で共有されるべき」と回答し、その 67％が「所定のルールの下で、関連公的機関が取りまとめるのがよい」と回答。
- 障害事例情報を公開する企業に対して、83％が「積極的に情報公開することから、公開しない企業に比べて信頼できる」と回答し、「障害を発生させていることから、信頼できない」との回答は無し。

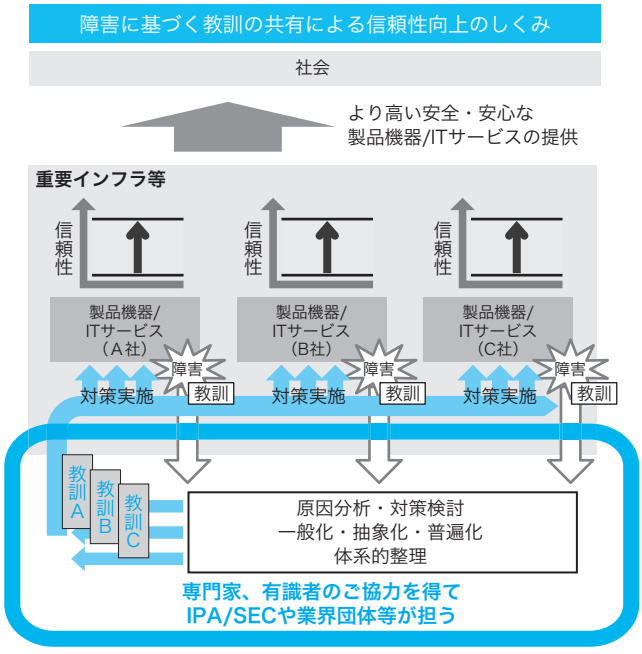


図 1　障害事例に基づく教訓の共有による信頼性向上の仕組み

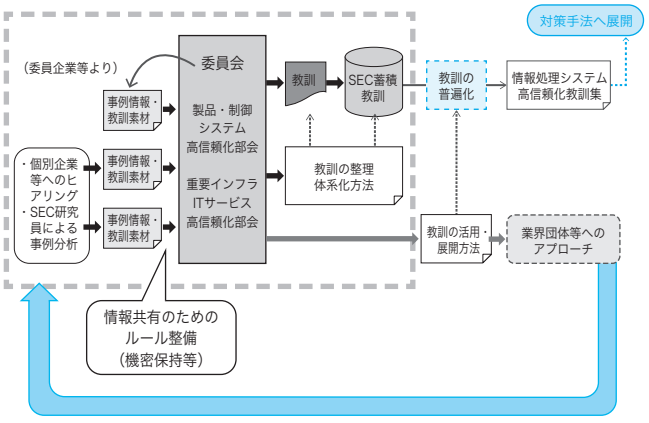


図 2　障害事例に基づく教訓の導出の流れ

システムグループ

重要インフラ等システム障害対策（製品・制御システム）

SEC 調査役 三原　幸博　SEC 研究員 松田　充弘　SEC 研究員 石田　茂　SEC 調査役 十山　圭介

交通機関や電気・水道の制御等、製品に組み込まれた機器の制御を行う「製品・制御システム」（組込みシステム）の障害情報の収集・分析と対策の検討を行い、その結果を普遍化した「教訓」として取りまとめ、「情報処理システム高信頼化教訓集（製品・制御システム編）」として公開した。

1 製品・制御システム分野におけるソフトウェア障害情報の収集・分析

近年、コンピュータを利用して制御や機能実現を図る機器や製品（以下、製品・制御システム）が増加している。これらの製品・制御システムの中には社会生活のインフラとして重要な役割を担うものも多く、それらには高い信頼性が求められる場合が少なくない。しかし、製品・制御システムは、実現する機能規模が肥大化すると共に複合化する傾向にあり、システム全体として信頼性を確保するための技術面での工夫や運用管理での工夫が求められている。

一方、グローバル化に伴う企業間競争の激化により、低価格・短納期の製品開発が主流となり、システム高信頼化のための技術やノウハウの伝承がうまく行われないといった問題も顕在化している。

システム高信頼化に関する技術やノウハウは個々の企業の中で自らの経験から習得した技術資産として認識されてはいるが、知としての共有と活用は必ずしも十分であるとは言い難い。近年、複雑化が進む製品・制御システムにあって、その信頼性やユーザにとっての安全性を考えた場合、システム高信頼化に関する技術やノウハウを企業・業界を越えて共有し活用しようとする意識の醸成とそれを可能にする仕組みの構築が重要になっている。

製品・制御システムのシステム信頼性に関する上記現状を鑑み、産業界におけるシステム高信頼の知見を集積し、将来に向けたシステム信頼性向上のための技術的な布石を打ち、その結果としてシステム信頼性に関する社会的な認識レベルを上げていくことを目的に、「製品・制御システム高信頼化部会」とその傘下に２つの WG を設置し、産業界の有識者を交えた議論を進めた。（図１）

（1）未然防止知識 WG：製品・制御システムの障害を未然に防止するためのノウハウや知見を分析収集し、産業界で共通に利用できる資料を作成する。

（2）障害事例検証 WG：製品・制御システムに関するシステム障害に関する事例研究を通して、システム障害発生時の対処法や障害要因分析の手法を整理する。

2 障害対策事例の収集と教訓集・対策事例集作成

2.1 背景と狙い

産業界で実践されているシステムの品質上の問題を未然に防ぐための知識をもとに、製品・制御システムの障害を抽象化、一般化することによって、組込みシステム開発企業において幅広く利活用できるための『智恵』として教訓集を作成した※１。併せて教訓を実践するための対策の事例をまとめた。

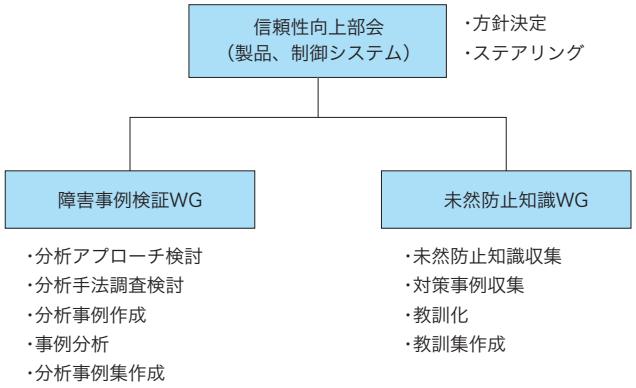


図１　製品・制御システムに関する部会・WG 構成

【脚注】

※１　http://www.ipa.go.jp/sec/reports/20140513_2.html

2.2 事例収集の方針

（1）概要

未然防止のための知識とは、他分野、他企業、他組織が起こした事故事例を抽象化、一般化することによって得られる、自分野、自組織において類似の問題を防ぐための知識である。また、未然防止のための知識には、内在的な障害を取り除くための品質向上のための知識と、内在的な障害が発現したとしても問題として外化するのを防ぐフォールトトレランスのための知識があるが、両者とも収集の対象とした。

表１　教訓の分類軸の候補

NO	分類軸候補
1	並行システム
2	複雑・組み合わせ多い
3	COTS 利用
4	ハードウェア利用
5	フォールトトレランス
6	設計思想
7	ヒューマンファクタ

表２　教訓一覧と対策が必要な工程との対応例

教訓番号	教訓タイトル	システム要求定義	システムアーキテクチャ設計	ソフトウェアアーキテクチャ設計	ソフトウェア詳細設計（変更設計）	実装（コーディング）	レビュー	システムテスト	教育	プロジェクトマネジメント	運用
1	複雑な条件式のロジック変更を行う場合は、デシジョンテーブル等による検証が有効である			○	○						
2	条件が整理されていない状態で、トータルの条件数が 100 を超えるような機能、または 10 個以上の条件を有する機能を修正する場合、関連する条件をすべて洗い出して整理し不整合がないことを確認する			○	○						
3	複数機能モジュールを統合する場合、統合前の条件数の総和と統合後の条件数を比較し差がある場合は、条件の抜けがないか確認する				○			○			
4	変数値域が広く、組み合わせバリエーションが非常に多くなる場合には、値域を適切な大きさに分割した上で境界値テストを実施する				○						
5	内蔵電池を使用する場合には、深放電時の起動シーケンスを考慮すること		○	○			○	○	○		
6	フラッシュメモリを使用する場合には、書き込み寿命回数を考慮すること	○	○							○	○
7	消費電力の多い機能を追加する場合には、一時的な電圧降下による影響（リセット、フリーズ等）や電源の種類、電池の場合は残量を考慮すること		○								
8	想定可能な例外を形式的に漏れなく分析する	○	○								
9	システムを二重化する場合は、同期すべきデータ領域を適切に設定する			○							
10	制御対象のハードウェアが同一でも、運用条件が変わるときは、ハードウェア仕様を再確認する		○		○		○		○		
11	プロセス間、スレッド間でデータを共有（引き渡し）する場合は、排他・同期処理が正しく行われているか、あるいはデッドロックが発生していないかどうか注意する			○		○			○		
12	歩留りのある製品の良品／不良品を検査する装置では、すべてが良品あるいは、不良品との検査結果は異常と判断すべきである	○									○
13	既存ソフトウェアの性能改善を実施する際には、アイドリングタイムの発生、処理の同期ずれの発生等と影響を確認する			○	○			○	○	○	
14	・大量のデータを通信経由で扱う場合、一連の処理の流れの中にボトルネックを作りこまないように注意する ・時間帯による負荷変動について考慮する	○	○	○			○				
15	納入したあと、お客様が運用するような業務システムでは、業務シーケンス中のあらゆる異常操作（リセット、電源断、放置も含め）、への対応を考える				○			○			
16	障害解析時の保守メンテ用ログ処理であっても、仕様書を作成し、影響評価を実施すること			○							
17	判断処理は、必要条件だけでなく、制限すべき条件も漏れなく抽出する				○						
18	ログファイルの断片化に注意する			○							

（2）想定利用者

未然防止知識の利用者としては、組込みシステム開発にかかわる以下の３者を想定している。

- ソフトウェア設計者
- ベンダ側システム設計者
- ユーザ側システム設計者

なお、システムを利用する特別な訓練を受けていないエンドユーザは、知識の利用が困難であると考え、想定利用者からは除外した。

（3）収集整理手順

組込みシステムの製品分野は多岐にわたるため、同一の企業内であっても他の分野の事例や知見では実感しにくいため、なるべく利用者の分野に近い事例に書き換える必要がある。そのため、下記の手順に従って、収集した事例や知見から肝となる部分を抽出し、別の事例に書き換えたものを知識として整理した。

step1：所定のシートでの情報提供／ヒアリング
step2：抽象化、他分野、別事例への書き換え
step3：対策の整理
step4：有識者によるレビューと補足

2.3 教訓の分類

（1）分類の観点

教訓の分類に関しては、教訓の数が未だ少ないこともあり、今回は、適用可能な開発プロセスで分類した。表１に示す分類軸の候補を含め。更なる分類方法を継続して検討していく。

（2）教訓と開発工程による分類

開発工程においてどのような対策を施せば、教訓が解決しようとしている問題を未然防止できるのかという観点から整理した。プロセスモデルは ESPR Ver.2.0:「組込みソフトウェア向け開発プロセスガイド」[1] のモデルを採用した。また、直接開発にかかわるシステム・エンジニアリング・プロセス（SYP）やソフトウェア・エンジニアリング・プロセス（SWP）だけでなく、サポート・プロセス（SUP）も対象とし、ESPR では定義されていない教育に関する工程も工程別未然防止知識の一部として採用した。また、組込みシステム開発において多く見られる差分開発特有の未然防止知識についても、切り出して記載した。

教訓と対策が必要な工程との対応例を表２に示す。

3 障害分析手法並びに分析事例集

製品・制御システムに関するシステム障害の未然防止には、障害を分析して直接原因を探り出して対策するだけでなく、根本原因の分析と（恒久）対策の立案が重要である。事例研究を通して、システム障害発生時の対処手順と障害要因分析の方法を整理し、手法の解説と事例を併せて公開した。

3.1 障害発生から分析・対策立案までの流れ

障害が発生した際に行う活動を図２に示す。次節で各活動の詳細について述べる。

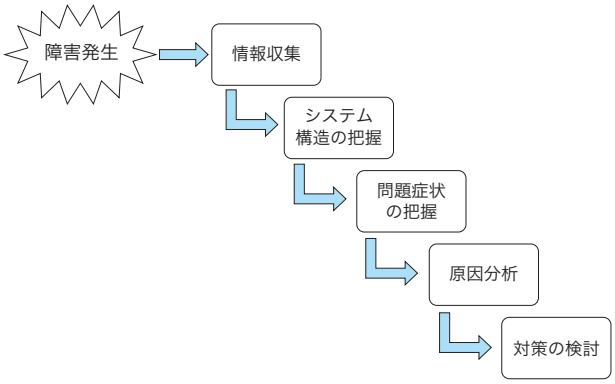


図２　障害分析・対策立案の流れ

3.2 分析の各ステップ詳細

（１）情報収集

障害の分析に際してまず行うのが情報収集である。障害の分析に必要な情報の収集は事態の収拾後速やかに行い、収集した情報は整理して文書にまとめる。システムの種類によって収集が必要な情報は異なり、収集できる情報が限られる場合もあるが、必要と思われる情報をできるだけ集めることが、以降の分析を円滑に進めるために肝要である。

（２）全体を把握する（システム構造の把握）

収集した情報を、障害発生の経緯に関する情報と、システムの動作・構造に関する情報に分類し、整理して対応付けることで、障害に至った問題症状の把握が可能になる。

とくにシステムの動作・構造に関する情報を整理して、システムの全体像を見渡せるような簡潔なシステム構造図を作成することで、サブシステム間の相互作用の関係が明確になる等、システムの構造と振る舞いの把握に繋がる。

（３）問題症状の把握（事象経過）

次に、時系列と相互作用を意識しながら事象を整理して問題症状を把握する。まず、ヒアリングなどで得た事象に関する情報を時系列に整理するとよい。このとき、観点を定めて事象を整理すると分析しやすくなる。また、（２）で作成したシステムの構造図を利用して、収集した情報に矛盾する点やあいまいな点がないことを確認する。

最後に、障害にかかわった人やシステムの構成要素の間の相互作用が分りやすくなるように事象を整理して一覧性の高い形式で図表にまとめるとよい。

（４）原因分析

原因分析は、以下の３つのステップで構成される。

①　原因個所の推定

問題症状を引き起こしたと考えられるシステム上の原因個所を推定する際にソフトウェア、ハードウェア、人、環境・想定条件の観点から原因を推定すると漏れや抜けを防ぎ易くなる。原因分析の担当者の経験なども貴重な情報源である。情報を得ることが難しい場合には、一定の仮定を置いて原因個所を推定する必要があるが、事実と区別できるように必ず記録する。

②　直接原因の特定

推定結果に基づいて直接原因を特定するために再試験によって障害の再現条件を調査するなどの作業が必要となる。ソフトウェア部分に問題があると推定される場合には、コードレビューやインスペクションを実施するとよい。

③　根本原因の特定

多くの場合は設計誤りや人為的ミスが直接原因となるが、そのような誤りやミスが行われた要因や見逃された要因を複数の視点に立って根本原因を分析するとよい。ここでは、開発担当者へのヒアリング結果等が貴重な情報源となる。

（５）対策の検討

特定された根本原因を取り除く対策を検討するが、すべての原因を取り除けない場合には影響を軽減する対策を検討する。開発プロセスや体制に関する現状や制約を踏まえた上で短期的 / 長期的対策を考えるとよい。

技術的課題か管理的な課題であるのかなど、複数の観点から検討することが重要である。

3.3　分析手法と分析事例

有用な分析手法のうち、調査した企業 の現場で利用されている８つの手法について、公開されている過去の障害事例に適用し、分析事例として整理した。その結果は、

「情報処理システム高信頼化教訓集（製品・制御システム編）」と併せて公開した。

[分析手法]

1　ブロック図
2　事故経過表
3　VTA（Variation Tree Analysis）
4　問題行動分析
5　PNA（プロセスネットワーク分析法）
6　発生源・検出漏れ分析
7　例外分析
8　なぜなぜ分析

[障害事例]

1　湘南モノレール事故
2　駒場ダム事故
3　アリアン５事故
4　カンタス航空事故

4 今後の課題

現場での教訓集の適用を促進するためのセミナー等の開催と、活用を意識した質・量両面からのブラッシュアップを進めていく。また、現場からの評価の収集にも努め、今後の活動にフィードバックしていく。

【参考文献】
[1] Embedded System Development Process Reference

システムグループ

重要インフラ等システム障害対策（IT サービス）

SEC 研究員　加藤　均
SEC 研究員　目黒　達生
SEC 研究員　平林　大典
SEC 主任　八嶋　俊介
SEC 調査役　大高　浩
SEC システムグループリーダー　山下　博之

IT サービスを担うシステムの主としてソフトウェアに起因する障害関連情報を収集し、それらの分析や対策の整理・体系化を行い「教訓」として普遍化し、類似障害の再発防止や影響範囲縮小のために業界・分野を超えて活用可能な「情報処理システム高信頼化教訓集（IT サービス編）」として取りまとめた。

1 障害事例情報の収集・分析及び対策の検討

IT システムは、今や私たちの生活や社会・経済基盤を支える重要インフラ分野※1 等のサービスに深く浸透している。その一方で、社会に大きな影響を与えたシステム障害の発生件数は、2009 年から 2012 年にかけて増加傾向にあり、以下のようなシステム障害に関するニュースを目にする機会も少なくない。

・○○システムで障害か、終日つながりにくく
　… 原因は、法律改正直前の駆け込み需要と期末の締め処理とが重なり、想定外の大量入力にシステムの性能が耐えられなかった模様。
・□□システムで障害、午前中のサービス停止
　… 原因は、システムは本番装置の故障により予備装置に自動的に切り替わるようになっていたが、その切替えが失敗したためという。

この背景には、システム障害の原因分析や発生防止対

【脚注】
※1　内閣官房情報セキュリティセンター（NISC）では「情報通信、金融、航空、鉄道、電力、ガス、政府・行政サービス、医療、水道、物流」の 10 分野を重要インフラに定義している。（平成 24 年 4 月 26 日「重要インフラの情報セキュリティ対策に係る第 2 次行動計画」）

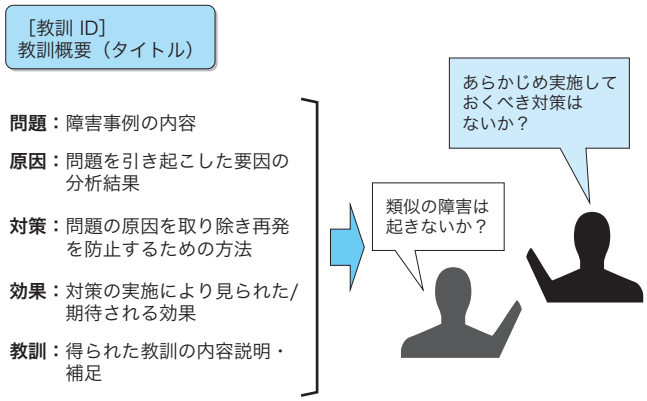


図１　各教訓の構成

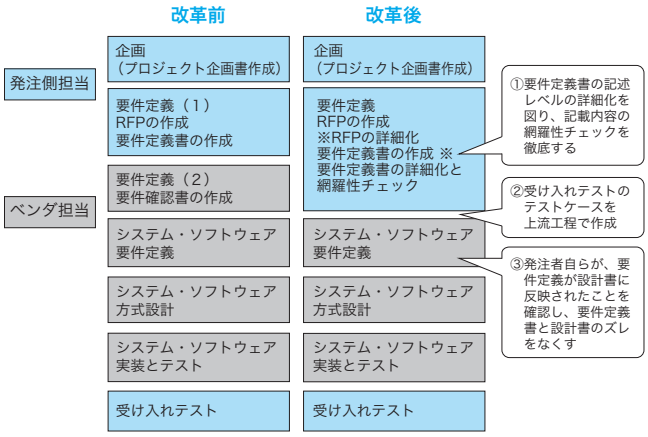


図２　教訓Ｇ２に基づく開発標準プロセスの改革例

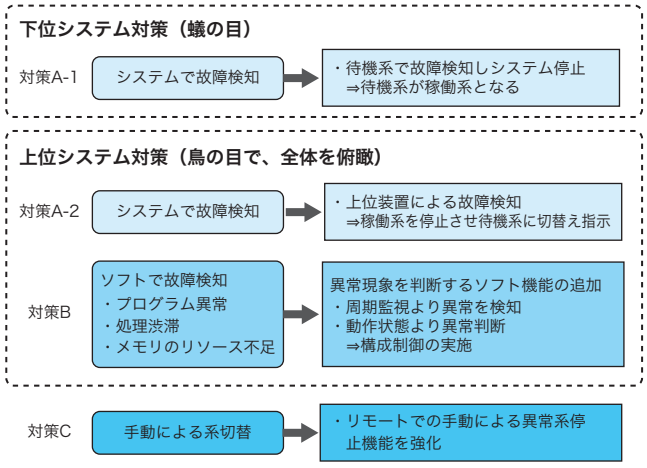


図３　教訓 T2 に基づく系切替え対策の例

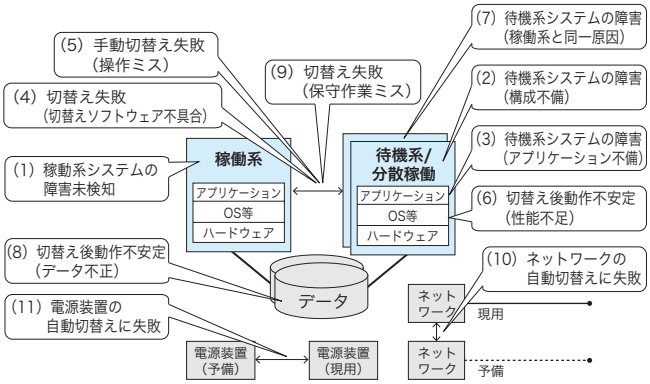


図４　教訓 T7 における切替え失敗の原因一覧

策などの情報が業界内で共有されておらず、類似の障害が繰り返し発生してしまう実状がある。障害が発生してもその詳しい情報が公開されずに当事者のみで対処されることが多く、また、一部の大規模障害では情報が公開されることがあるものの、その情報が特定の事例への対応策となっている場合が多いため、障害に関する情報が他者への参考として活かされにくいこと等が考えられる。

そこで、システムの障害事例情報の分析や対策手法の整理・体系化を通して得られる「教訓」を業界・分野を超えて幅広く共有し、類似障害の再発防止や影響範囲縮小に繋げる仕組みの構築に向けた活動を開始した。

2013 年度は、そのスタートポイントとして、以下の活動を実施した。まず、教訓化活動 として、電力、鉄道、保険、証券等の分野において、企業からの情報提供や有識者からのヒアリング等により過去の障害事例を収集した。並行して、銀行、保険、証券、電力、鉄道、情報通信、政府・行政等の多分野の CIO クラスを中心とする有識者・専門家の委員を中心とする「重要インフラ IT サービス高信頼化部会」を設置し、収集した障害事例情報を共有しつつ、その分析と対策の検討及びそれらの一般化を行った。その結果、15 件の教訓候補を精査し、下記特徴を有する教訓（９件）を導出した（表１）。

① 複数の重要インフラ分野等の有識者・専門家による「重要インフラ IT サービス高信頼化部会」において多方面から考察を行い、業界横断的に利用可能な要素を抽出。

② 所定の機密保持ルール（今回同時公開）により収集した、これまで一般には未公開の事例や情報も対象に原因や対策について考察。

③ 有識者・専門家の豊富な経験に基づく知見と、IPA/SEC の 10 年間の活動で蓄積されたソフトウェアエンジニアリングに関する検討成果に基づいて取りまとめ、技術領域に加え、ガバナンス／マネジメント領域も対象に教訓を整理。

これらの教訓を「情報処理システム高信頼化教訓集（IT サービス編）」として取りまとめた※２（図１～４）。

【脚注】
※２　http://www.ipa.go.jp/sec/reports/20140513.html

表１　IT サービスに関する教訓一覧

No	領域	ID	教訓概要
1	ガバナンス／マネジメント	G1	システム開発を情シス部門だけの仕事にせず、各事業部門が自分のこととして捉える「態勢」をつくることが大切
2		G2	発注者は要件定義に責任を持ってシステム構築にかかわるべし
3	技術	T1	サービスの継続を優先するシステムにおいては、疑わしき構成要素を積極的にシステムから切り離せ（“ フェールソフト ” の考え方）
4		T2	蟻の目だけでなく、システム全体を俯瞰する鳥の目で総合的な対策を行うべし！
5		T3	現場をよく知り、現場の知識を集約し、現場の動きをシミュレートできるようにすべし！
6		T4	システム全体に影響する変化点を明確にし、その管理ルールを策定せよ！
7		T5	サービスの視点で、「変更管理」の仕組み作りと「品質管理責任」の明確化を！
8		T6	テスト環境と本番環境の差異を体系的に整理し、障害のリスク対策を練るべし
9		T7	バックアップ切替えが失敗する場合を考慮すべし

表２　障害対策手法一覧

		障害対策手法								
領域	対策事例に対応する教訓 ID	① 超上流工程での要求 品質管理 ・ユーザ企業内の事業 部門と情シス部門との 連携 ・ユーザ企業とベンダ 企業の連携、合意形成	② トレーサビリティ管理	③「見える化」手法 ・暗黙知の整備・有効活用 ・俯瞰図	④ 要求獲得手法	⑤ 変更管理	⑥ フェール ソフト	⑦ 網羅的テスト技法 ・テスト環境のリスク管理 ・シミュレーション手法	⑧ 可用性管理 ・システムの冗長化設計 ・シングルポイントの洗出し ・障害運用マニュアルの整備と 訓練	⑨ 非機能要求グレード
ガバナンス / マネジメント領域	G 1	○								
	G 2	○								
技術領域	T 1						○			
	T 2			○						
	T 3			○				○		
	T 4				○	○				
	T 5		○		○	○				
	T 6							○		
	T 7								○	○

表３　障害分析手法一覧

No	分類	名称	開発機関
1	過程関連型	FTA（Fault Tree Analysis）	Bell Telephone Lav. 他
2		ImSAFER（Improvement for medical System by Analyzing Fault root in human Error incident）	自治医科大学
3		RCA（Root Cause Analysis）	米国退役軍人省　患者安全センター
4	リスク評価型	FMEA（Failure Mode and Effects Analysis）	US.Army が最初に導入
5		HAZOP（Hazard and Operability Studies）	英国 ICI 社（Imperial Chemical Industries）
6	基本型	なぜなぜ分析	各社（品質管理手法）
7	IT 特化型	総合的インシデント分析	富士通株式会社
8	発展型	STAMP	マサチューセッツ工科大学（MIT）
9		STPA（STAMP）	マサチューセッツ工科大学（MIT）
10		CAST（STAMP）	マサチューセッツ工科大学（MIT）

・FTA：下位アイテムまたは外部事象、若しくはこれらの組み合わせのフォールトモードのいずれが、定められたフォールトモードを発生させ得るか決めるための、フォールトの木形式で表された解析方法。
・ImSAFER：ヒューマンエラーが関係した事象分析手法であり、原因追究と対策立案を支援する。
・RCA：問題や事象の根本原因を明らかにすることを目的として使用される。
・FMEA：設計の不完全や潜在的な欠点を見出すために構成要素の故障モードとその上位アイテムへの影響を解析する技法。
・HAZOP：設計意図からの逸脱によるハザードを明示する手法。効率的な運転や操作に妨げとなる設計・運転上の意図からの「ズレ」を設定し、そこから想定される潜在的な危険性を定義し評価するための体系的な手法。
・なぜなぜ分析：問題事象から発生原因まで、「なぜ」と問いながら遡っていく分析手法。
・総合的インシデント分析：日々発生するインシデントに着目した総合的なインシデント分析手法。
・STAMP：マサチューセッツ工科大学のナンシー・レブソン教授が提唱する因果関係のモデル
・STPA（STAMP）（STAMP based Process Analysis）：マサチューセッツ工科大学のナンシー・レブソン教授が提唱する因果関係のモデル STAMP（System-Theoretic Accident Model and Processes）に基づく安全解析手法。
・CAST（STAMP）（Causal Analysis using STAMP）：STAMP を使用した要因解析手法

また、障害再発防止に向けた対策についても先進的企業等の取り組み事例を収集し、過去に IPA/SEC で蓄積されたソフトウェア・エンジニアリング手法を活用し「障害対策手法・事例集」として取りまとめ、「情報処理システム高信頼化教訓集（IT サービス編）」と併せて公開した（表２）。

また、重要インフラ分野の業界団体である、電気事業連合会の会員企業（18 企業）、財団法人地方自治情報センター（LASDEC）から推薦された地方公共団体（10 団体）、一般社団法人日本損害保険協会の会員企業（12 企業）に対するアンケート結果によれば、障害事例に基づく教訓共有の取り組みについて、「関心がある」、「成果が適用できる」との回答が高い割合を示した（図５）。

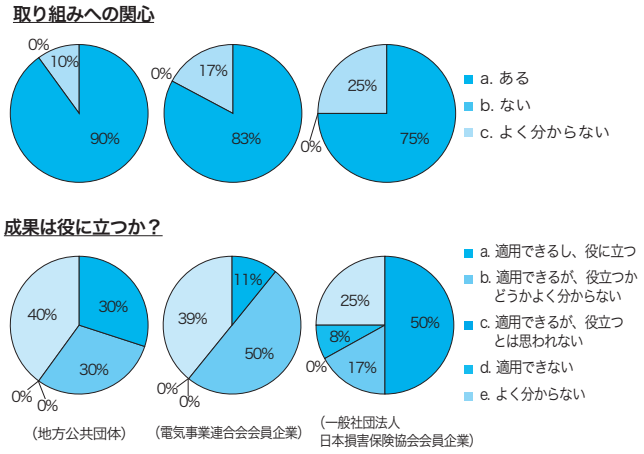


図５　業界団体に対するアンケート結果

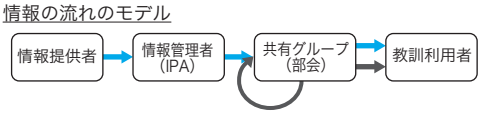
2 ソフトウェア障害の再発防止の原因分析支援

障害の原因、とくに根本原因を探ったり、その対策を導いたりするときに利用することを目的に、文献等により障害の分析手法（表３）を調査すると共に、その適用事例、障害情報分析についての各社の取り組み状況を再整理した。これらを「障害分析手法・事例集」として取りまとめ、「情報処理システム高信頼化教訓集（IT サービス編）」と併せて公開した。

3 障害情報提供に関する機密保持ルールの作成

障害事例ヒアリング、障害情報共有グループ（部会等）での議論、及び教訓の公開時において必要な、障害情報を記録する共通様式、障害情報提供に関する機密保持・情報提供ルール（図６）を作成し、公開に際しての事例情報の抽象化を明記する等、部会の意見を反映した上で、「情報処理システム高信頼化教訓集」と共に公開した。

情報の流れのモデル



No.	収集方法	ルールの適用局面（義務を負う者）			
		情報提供者から預かった障害情報の管理（情報管理者）	共有グループへの情報開示時（情報管理者）	共有グループでの議論時（共有グループメンバ）	教訓の公開時（情報管理者）
1	情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合	情報提供者との間で、機密保持のルールを適用	情報提供者との間で、加工情報の共有に係るルールを適用	情報管理者との間で、機密保持のルールを適用	情報提供者との間で、公開のための情報加工のルールを適用
2	共有グループのメンバが、障害情報をグループ内に直接開示した場合	適用なし	適用なし	共有メンバ間で、機密保持のルールを適用	情報提供者との間で、公開のための情報加工のルールを適用

（注）ここでは、情報管理者を IPA、共有グループを IPA 内に設置された部会として説明する。
また、共有グループの運用を情報管理者が行うことを前提としている。
情報管理者は共有グループのメンバでもある。
これらは、本ルールの適用先に応じ、適切な組織・会議体に置き換えることになる。

図６　障害事例情報収集・公開のモデルと機密保持ルールの分類

4 ソフトウェア障害事例に対する対策支援

ソフトウェアが関係し得る障害発生時の調査・対策支援を担える機関への発展に向け、専門家とのネットワーク構築作りと、組織としての知識の蓄積とスキルの向上を徐々に実現していくために、部会委員の協力を得て、分析する態勢を構築（試行）した（図７）。

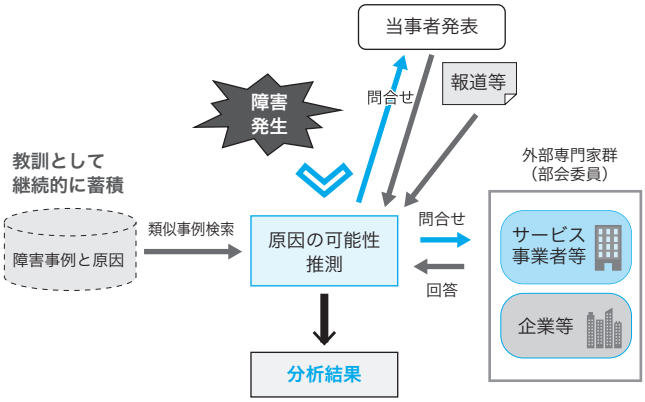


図７　障害発生時の調査・対策支援態勢（試行）

5 今後の予定

重要インフラ IT サービス高信頼化部会における検討では、各分野の障害事例に対して、他分野の委員から、自分野のコンテキストに照らして、どのような事象が考えられるか、また、どのような対策が参考となるか、といった観点から活発な議論が行われた。このように、障害事

例の背景や環境にまで深く踏み込んで分析し一般化・抽象化したため、業界・分野を超えて、類似障害の発生防止対策として役立つ『社会智』が得られたと思う。また、委員からは、異分野の事例の分析過程で新たな「気づき」が得られたとの声もあった。部会での議論においては、機密保持に関するルールの下で、公開できるレベルに抽象化された本教訓集の内容より更に深い情報についても紹介され、参加者にとっては非常に有益であったと思われる。

このような取り組みは、より広く展開されることが重要である。その際には、今回の IPA/SEC における活動とその成果が参考になるであろう。今回 IPA/SEC において試行された仕組みを今後幅広く展開する方法としては、図８に示すように、分野・業界ごとに情報共有グループを設置することが考えられる。コンテキストの同じ関係者が集まる同分野・業界におけるグループでは、より精緻な議論が行われるものと期待される。そして、分野・業界を超えた情報共有の仕組みも必要となる。今後、各産業分野の業界団体等に働きかけつつ、徐々に展開を図っていきたい。

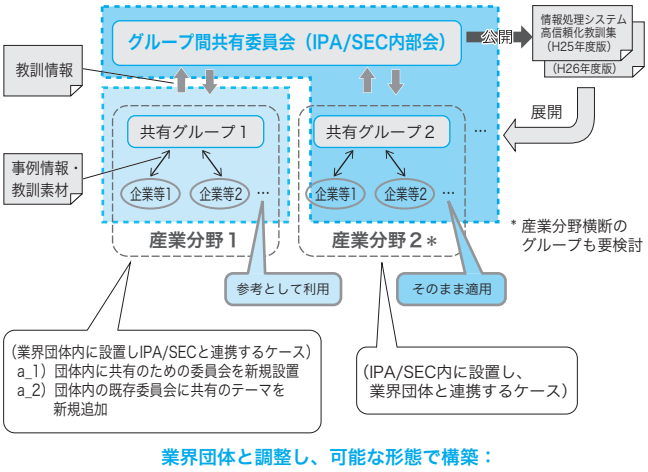


図８　情報共有の仕組みの拡大

システムグループ

定量的プロジェクト管理による信頼性・生産性向上

SEC 研究員
佐伯　正夫

SEC 調査役
三縄　俊信

SEC 研究員
森下　哲成

SEC 研究員
松田　充弘

SEC システムグループリーダー
山下　博之

ソフトウェア開発データの分析に基づくソフトウェアの信頼性・生産性向上を目指し、ソフトウェア開発データ白書の定期的発行、メトリクス分析に関する研究等への蓄積データの活用拡大等の活動を実施すると共に、組込み系ソフトウェア開発データ白書作成に向けたデータ収集を試行した。また、ソフトウェア開発データ白書に関する IPA 成果の提案等、IT プロジェクトベンチマーキングの国際標準化に貢献している。

1　ソフトウェア開発データ白書の定期的発行

ソフトウェア開発データのベンチマーキングへの活用により情報システムの信頼性・生産性向上に資することを目指し、平成 26 年度発行予定の「ソフトウェア開発データ白書 2014-2015」の素案（629 頁）を作成した。

また、ソフトウェアの信頼性向上のための定量データ分析（メトリクス分析）に関する方策の検討を目的とした高信頼性定量化部会を、平成 25 年 9 月 26 日に立ち上げた（委員 17 名）。４回の議論を経て、具体的検討作業を目的とした信頼性メトリクス WG（委員 11 名）及び IT サービス定量データ分析 WG（委員 12 名）を、平成 26 年 1 月 22 日に高信頼性定量化部会内に立ち上げた。両 WG の活動を通じて、新しい有用なメトリクス分析手法や事例を提供すると共に、メトリクス分析によって得られた信頼性向上のための新たな知見を発信して行

表１　新規分析項目一覧

分類	分析項目
信頼性	1. 開発体制と信頼性
	2. 設計工程のレビューとテスト工程のバグ
	3. 文書化の密度と信頼性
	4. 顧客の要求レベルと信頼性
	5. 短納期開発についての信頼性
	6. テストカバレッジと信頼性
生産性	7. 顧客の要求レベルと生産性
	8. 短納期開発についての生産性
	9. 顧客の体制と上流フェーズの生産性
	10. 重要インフラシステムに対する生産性
その他	11. 経年推移
	12. 機能規模、成果物量、工数の関係

表２　新規分析項目（白書追加掲載分）

分類	分析項目	分析結果
信頼性	開発体制と信頼性	品質保証体制やテスト体制、定量的な出荷品質基準の有無などの開発体制が適切な場合、ソフトウェアの信頼性が高い。
	顧客の要求レベルと信頼性	信頼性や性能に対する顧客の要求が強い場合、ソフトウェアの信頼性が高い。
生産性	顧客の要求レベルと生産性	信頼性や性能に対する顧客の要求が強い場合、ソフトウェア開発の生産性が低い（工数が多い）。
	顧客の体制と上流フェーズの生産性	プロジェクトに対する顧客の関与が強い場合、設計フェーズの生産性が高い。
	重要インフラシステムに対する生産性	重要インフラ情報システムのシステムプロファイルが高い（重要性が高い情報システム）では、ソフトウェア開発の生産性が低い。
その他	経年推移	アーキテクチャや開発言語など、収集されるデータのプロジェクト特性が変化している。

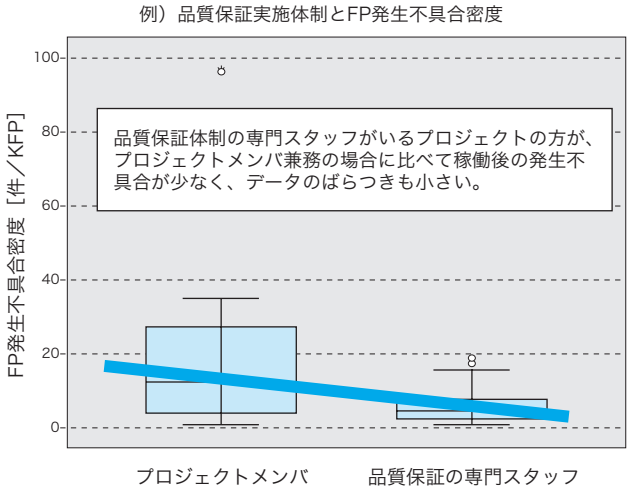


図１　新規分析結果の例（品質保証実施体制と FP 発生不具合密度）

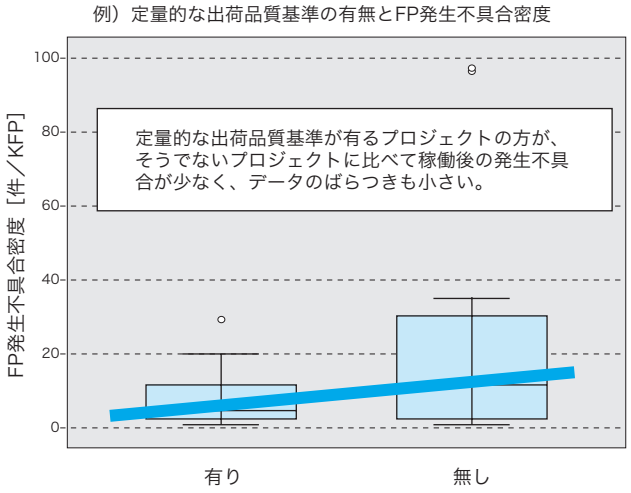


図２　新規分析結果の例（定量的な出荷品質基準の有無と FP 発生不具合密度）

く計画を策定した。

1.1 ソフトウェア開発データ白書 2014-2015

（1）データの収集と精査

平成 25 年度には、データ提供企業 23 社から 216 プロジェクトのソフトウェア開発データを収集・精査した。前年度に収集した 236 プロジェクトのデータと合わせて、前版（「ソフトウェア開発データ白書 2012-2013」）に 452 プロジェクトを追加した 3,541 プロジェクトのデータを分析対象とした。

（2）新規分析項目の検討

信頼性に関する分析テーマを重点に、白書に新規に追加する分析項目の候補を検討し、表 1 に示す 12 の分析項目を新たに設定した。

（3）データ分析及び白書素案の作成・レビュー

既存の分析項目及び新規分析項目について 3,451 プロジェクトのデータを分析し、その結果に基づいて「ソフトウェア開発データ白書 2014-2015」の素案を作成した。新規分析項目の分析結果については、高信頼性定量化部会によるレビュー・検討の結果、分析結果の有意性等を勘案して、最終的に表 2 に示す 6 項目について、63 個の図表を白書に追加掲載することとした。

信頼性に関しては、高信頼化を達成するための開発プロジェクト要因として開発体制、顧客の信頼性要件等に

着目し、開発されたソフトウェアの信頼性実績との関係を分析した。一例として、開発体制と信頼性実績との関係の分析における、品質保証実施体制と FP 発生不具合密度との関係、及び定量的な出荷品質基準の有無と FP 発生不具合密度との関係の分析結果を、それぞれ図 1 及び図 2 に示す。

1.2 ソフトウェア開発データ白書の改良

ソフトウェア開発データ白書の改良を目的とし、IT サービス定量データ分析 WG にて、次の検討を開始した。

◇ 新しい分析項目とその評価（必要に応じて試行分析して実現性を評価）
◇ 将来の信頼性関連収集データ項目（開発～運用時）
◇ 収集データ精度向上のためのデータ記入要領の改良
◇ 利用シーンに沿った掲載項目の見直し・再整理
◇ 白書掲載内容の階層化と重点化
◇ 統計的手法の更なる活用
◇ 分かりやすさの追究

1.3 新しいメトリクス分析手法の検討

新しいメトリクス分析手法及び事例を提供すると共に、信頼性向上のための知見を発信することを目的とし、信頼性メトリクス WG にて、次の検討を開始した。

◇ 新しいメトリクス分析手法（組織毎の分析結果を組織横断的に分析して共通的な知見を導出する手法）
◇ 信頼性向上に有益な新たな知見の導出（高信頼性が要求される開発にはコストがかかる、早い段階から品質をコントロールすれば結果指標が良くなる、という仮説に関する知見）

2 蓄積ソフトウェア開発データの活用促進

2.1 メトリクス分析に関する研究への活用

蓄積されているソフトウェア開発データをより一層活用し、ソフトウェアの信頼性・生産性向上に繋がる新たな分析手法の発見等を目指し、所定の守秘義務の下で蓄積データを大学に提供し、以下の研究に貢献した。

（1）東海大学

◇ 解説記事：「FP 計測手法における FP 規模と工数の相関の差」（SEC journal33 号、平成 25 年 7 月 31 日発行）

（2）法政大学

◇ 国際学会講演：
「A Note on Modeling of Quality Evaluation Based on Large Data Sets in Software Development Projects」（The 14th Asia Pacific Industrial Engineering and Management Systems Conference（APIEMS 2013））
◇ 卒業論文：
―「プロジェクトデータの充足度と不良発生数の関係性の研究」
―「プロジェクトマネージャの能力とプロジェクト成功度の関係の研究」

また、ソフトウェア工学分野の先導的研究支援事業の一環として、新規区分として D 区分（ソフトウェア工学に関する課題指定研究「ソフトウェア開発データの分析」）を追加し、公募した。（今回は応募は無かった。）

2.2 正確な参考情報の提供と適切な説明

より正確な参考情報を提供することを目的とし、ソフトウェア開発データ白書の開発規模と工数との関係、工数と工期との関係等における回帰式と定数を、平成 26 年度発行予定の「ソフトウェア開発データ白書 2014-2015」にて明記することとした。また、ソフトウェア開発データ白書に掲載しているグラフに対応する、Excel 等によるグラフ化用データもダウンロード可能とすることとした。

＜回帰式の定数公開のイメージ＞

従来：（工期）= A ×（工数）^{B}、B=0.32、R=0.73
今後：（工期）= 2.45 ×（工数）^{B}、B=0.32、R=0.73

これらの情報公開にあたり、データ提供企業と調整すると共に、提供情報の意味を解説して正しい使い方を促すための「利用上の注意事項」を作成した。同注意事項についても、「ソフトウェア開発データ白書 2014-2015」に掲載すると共に、プロジェクト診断支援ツール内に明記する予定である。

3 組込み系ソフトウェア開発データ白書

現行の「ソフトウェア開発データ白書」は、主にエンタープライズ系ソフトウェアを対象としたもので、定量的プロジェクト管理のために広く活用されている。一方で、組込みソフトウェア開発に携わる開発者からは、組込みソフトウェア開発を直接の対象とした「組込み系データ白書」の編纂を切望されてきた。IPA/SEC は 2013 年度、新たな中期計画の開始を契機に、組込み系を対象にプロジェクトデータの収集と分析を試行した。

「組込み系データ白書」の目的は、現行のエンタープライズ系のものと同じであり、自社の開発プロジェクト

の欠点や弱点をベンチマークできることにある。

組込み系と一言でいうものの、組込み系の製品や機器は、使用される環境が様々であり、ソフトウェアを実行させるプラットフォームも様々であるため、収集したデータをひとくくりに分析できない。一方で、「組込み系データ白書」では、製品や機器で分類された統計情報が公開してもらえるものと期待する傾向がある。

2013 年度の試行では、製品や機器での分類ではなく、プロジェクトプロファイル（リアルタイム性の要否や自然環境条件の影響有無等）で分類することにより、ベンチマーク可能な分析データを得られると判断し、収集項目を決定した。そして、従来から IPA/SEC の活動に協力的な企業を中心に協力を打診し、6 社から計 65 件のプロジェクトデータの提供を受けることが出来た。そのため、その分析情報を 6 社で共有し、製品・制御システム定量データ収集・分析 WG を設置し、今後の進め方を議論することが出来た。

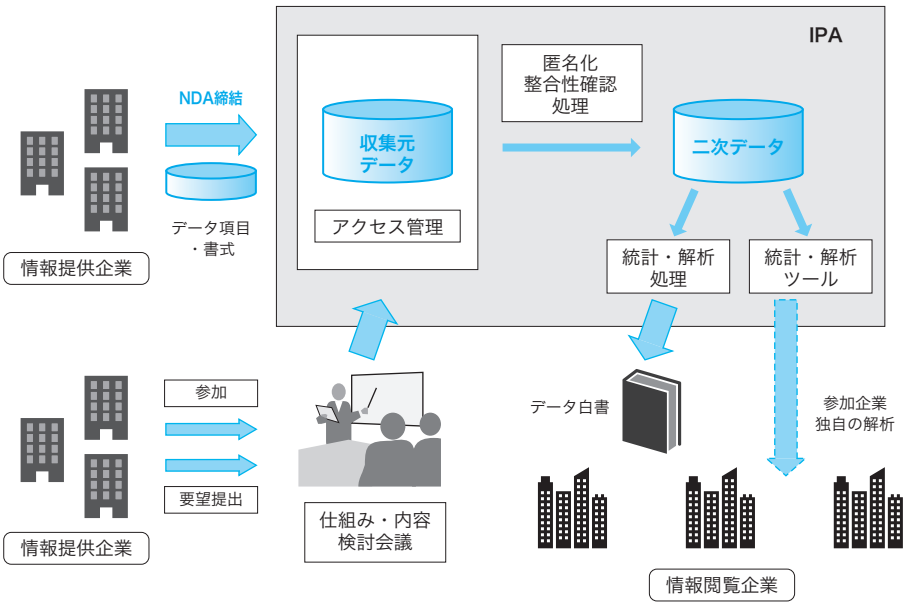


図３　組込み系データ白書作成と活用の流れ

また、「組込み系ソフトウェア開発データ白書」の将来的な公開に向けた準備を推し進めて行くためには、収集するプロジェクトデータの件数を蓄積してゆくことが不可欠である。広く協力企業を募るために、何をするために「組込み系ソフトウェア開発データ白書」を編纂しようとしているのか、あらためてその意義を下記に示す。

（1）組込みデータ白書が目指す利用され方（目標）

- 自社の開発プロジェクトの欠点や弱点をベンチマークできる
- 開発ツール、開発プラットフォーム、開発環境のトレンドが分かる

（2）組込みデータ白書が排除する利用され方（禁止）

- 特定データの提供企業名を推測し、ネガティブキャンペーンに用いること
- 下請け企業に対する査定やコスト低減の指標として用いられること

上記（1）は、利用者側の利点を示しているが、利用者側の準備として、社内に定量的管理の仕組みが構築されていることが前提になることに留意されたい。また、（2）については、公開の賛否が問われる重要な議題であるため、SEC と情報提供企業との間で十分な議論が必要である。

「組込み系データ白書」編纂への協力企業の募集は、SECjounal37 号の本記事内にて行うと共に、今後の展示会等でも行う予定である。また、賛同いただけた企業とは、IPA との間で NDA を締結していただき、参加企業独自の分析が行える等の仕組みも構築してゆく（図３参照）。

4　ベンチマーキング標準化

ISO/IEC JTC1/SC7[※1] にて進められている IT プロジェクトベンチマーキングの国際標準化に IPA 成果に基づく規格案を提案してきており、当該年度には以下の進展があった：

- ISO/IEC 29155-2（ベンチマーキング―実施手順）の国際規格発行（平成 25 年 11 月 1 日）。
- ISO/IEC 29155-3（ベンチマーキング―報告様式）の DIS[※2] 投票付議。
- ISO/IEC 29155-4（ベンチマーキング―データの収集と管理）の CD[※3] 投票付議。

【脚注】

※ 1　ISO/IEC JTC1/SC7: ISO（International Organization for Standardization, 国際標準化機構）/IEC（International Electrotechnical Commission, 国際電子標準会議）JTC1（Joint Technical Committee, 合同技術委員会 1）/SC7（Subcommittee 7, 専門委員会 7）

※ 2　DIS（Draft International Standard）：国際規格原案。

※ 3　CD（Committee Draft）：委員会原案。

システムグループ

コーディング作法ガイド：ESCR［C言語版］Ver. 2.0 の発行

SEC 調査役
十山　圭介

1 コーディング作法ガイド：ESCR［C言語版］の特徴と普及状況

IPA/SEC では組込みソフトウェアのソースコード品質をより良いものとすることを目的に、コーディングの際に注意すべき事柄やノウハウを「コーディング作法」という形で整備している。ESCR［C 言語版］はその第一弾として、組込みソフトウェア開発で広く利用されている C 言語に向けて 2006 年に Ver. 1.0 を発行したもので、コーディングの際の基本的な考え方を「作法」、作法を具体化した守るべき個々の事項を「ルール」とし、それらをソフトウェアの品質特性の観点で整理している。更にルールの選択指針と採用決定の手順によるフレームワークを提示することで、組織内のコーディングルールを決める際の参考やプログラミングの学習に用いられている。2007 年には一部修正を行った Ver. 1.1 を発行し、書籍発行部数に PDF 版ダウンロード数を加えるとこれまでに 3 万冊を超えて多くの方々にご利用いただいていることになる。

2012 年度「ソフトウェア産業の実態把握に関する調査」によると、組込み系ユーザでは 4 割が ESCR を導入済みであるか導入を検討しており、品質と生産性の向上に有用との評価を受けている。また、ソフトウェア開発ツールの国内ベンダー 4 社によって、ESCR に基づくコードチェック機能を実装したツールが提供されている。

なお、ESCR のフレームワークは JIS X 0180「組込みソフトウェア向けコーディング規約の作成方法」として標準化されている。

2 Ver. 1.1 の課題

ESCR［C 言語版］Ver. 1.1 までは、発行当時 C 言語規格として最も一般的に普及していた C90 に準拠していたが、昨今後継の規格である C99 が使われるようになりつつある。この状況に対応するべく欧州組込み業界標準の MISRA C※1 が 2013 年 3 月に MISRA C:2012 として大幅に改訂されている。

そこで、最新の JIS 規格である C99 に対応する必要があること、また ESCR は MISRA C を参照しているため、両者共通的な部分で MISRA C の改訂によって生じる食い違いを解消すること、の二点が ESCR［C 言語版］Ver. 1.1 に対する課題となっていた。

3 改訂の目的と概要

2013 年度は、以下の二点を目的に Ver. 2.0 として ESCR を改訂した。

- **準拠する言語規格を JIS 最新の C99 とし、新機能を利用しやすいルールとする**
- **相互関係にある MISRA C の改訂に合わせ、旧版（MISRA C:2004）への参照個所を見直し整合性を取る**

Ver. 1.1 からの記述の継続性を保つため、作法やルールの番号は変更せず C99 で拡張された言語仕様に関する項目について、ルールやルールの解説文、適合例 / 不適合例の追加や一部変更などを行っている。削除したルールもあり、その番号は欠番としている。追加・修正・削除した作法やルールの個数は表 1 のとおりである。MISRA C の参照個所に関しては、旧版から変更や追加のあったルールでは改訂版（MISRA C:2012）のルール番

表 1　追加・修正・削除した作法やルールの数

更新の種類	対象となった作法・ルールの数	備　考
修正	62	表現の明確化など軽微な修正も含む
新規	5	指示子付の初期化子や可変長配列など
削除	2	continue 文など

【脚注】

※1　MISRA C：英国 MISRA（The Motor Industry Software Reliability Association）が作成した C 言語のためのソフトウェア開発標準規格。MISRA は自動車メーカ、部品メーカ、研究者からなる欧州の自動車業界団体。

号を、旧版にしか存在しないルールでは旧版のルール番号を記載している。

ESCR［C 言語版］Ver. 2.0 は 2014 年 3 月 7 日に発行されており※2、この英語訳の PDF を 2014 年 5 月に公開する予定である。

4 MISRA との連携

ESCR 策定においては MISRA C を参考としており、ESCR のルールとして MISRA C のルールを採り入れたものや関連を示している部分がある。一方 MISRA C:2012 で ESCR の記述内容が引用されている。

MISRA C:2012 への改訂では MISRA から打診を受けて IPA/SEC の ESCR チームもドラフトのレビューを行った。幾つかのコメントが改訂版で反映されると共に、2014 年 3 月に実施した MISRA との打合せにおいても IPA/SEC によるレビューが記述の明確化や分かりやすさの向上に有効であったとの評価を受けている。

今後、両者とも C++ 言語版のガイドライン改訂にとりかかる計画であり、相互にガイドラインについてレビューや意見交換を実施することで連携を強化する合意を得ている。

【脚注】

※2 http://www.ipa.go.jp/sec/reports/20140307.html

システムグループ

ソフトウェア・エンジニアリング成果の普及展開

SEC システムグループ リーダー
山下　博之

信頼性の高いソフトウェアを効率よく開発するための手法の普及展開に向け、ソフトウェア・エンジニアリング成果※1 に関する首都圏でのセミナー主催、地域開催セミナーへの講師派遣、イベント出展等を行った。ここでは、幾つかのトピックスを述べる。

1 非機能要求グレード（地方公共団体版）の作成支援

地方公共団体での情報システム調達において、非機能要求を漏れなく明確化し、効率よくシステム調達仕様を作成するために特化した、「非機能要求グレード」のカスタマイズ版が、財団法人地方自治情報センター（現、地方公共団体情報システム機構）により作成・公開された※2。オリジナル版の非機能要求項目（全 236 項目）から目的実現に必要な 76 項目に集約（一部項目追加も実施）されており、この過程で、IPA/SEC は助言と協力を行った。

2 SPEAK-IPA 準アセッサ育成コースの開講

プロセス改善推進者育成の促進を目的に、「SPEAK-IPA」に規定されているソフトウェア開発プロセスのアセスメント能力を身に付けることができるセミナーコース「SPEAK-IPA 準アセッサ育成セミナー」を開始した。演習を主体とする 3 回・6 日間のコースであり、当該年度は東京と名古屋で開講し、計 25 名が修了した※3。

3 共通フレーム 2013 活用の利便性向上

「共通フレーム 2013」の活用時によく利用するプロセス一覧等の図表類を、クリエイティブ・コモンズ・ライセンス（表示 - 継承 2.1 日本）の下に、加工できる形式で公開した※4。

【脚注】

※1 ソフトウェア開発データ白書、定量的プロジェクト管理ツール、共通フレーム 2013、プロセス改善手法（SPEAK-IPA、SPINA^{3}CH 自律改善メソッド）、非機能要求グレード、アジャイル型開発プラクティス活用リファレンスガイド、高回復力システム基盤導入ガイド、GQM+Strategies、高品質な組込みソフトウェア開発標準リファレンス（ESxR Series）。
GQM+Strategies とは、組織ゴールからその実現のための戦略の整合性を体系立てて見える化し、関係者間での合意形成を図る方法論。

※2 http://www.ipa.go.jp/sec/info/20140325.html

※3 http://www.ipa.go.jp/sec/info/20140327.html

※4 http://www.ipa.go.jp/sec/publish/tn12-006.html

ソフトウェアグループ

ソフトウェア信頼性の見える化
～ 2013 年度の取り組み～

SEC ソフトウェアグループ
中村　雄三　　中尾　昌善

1 はじめに

今日、ソフトウェアが組み込まれた製品・システムは日常生活に無くてはならない社会基盤となり、ソフトウェアの不具合に起因する機器の故障やシステムの停止が社会に与える影響が拡大してきている。このように重要な役割を担うソフトウェアに関し、安全性、セキュリティ、可用性、不具合の有無などを総称して、ここでは「ソフトウェアの信頼性」、あるいは「ソフトウェアの品質」と呼ぶことにする。

利用者が安心して製品・システムを使えるようにするために、平成 25 年度から開始した第三期中期計画では、「ソフトウェアの利用者視点での信頼性の見える化」という取り組み課題を設定しており、その初年度である平成 25 年度においては、以下のような活動を行ってきた。

2 ソフトウェアサプライチェーンでの課題解決

近年、製品・サービスの多機能化・高機能化、相互接続のためのオープン化、OSS やパッケージの利用などに伴い、取引先や仕様の決定主体が異なるなどのソフトウェアサプライチェーンの変化が生じている。また、利用者が個々に購入した製品・システムが相互に連携して動作する事も増えている。このため、信頼性に関して構成要素のすべての内容を把握することが困難になって来ている。平成 25 年度は、各業界企業へのヒアリングを通じて、サプライチェーン上の課題を浮き彫りにしたが、平成 26 年度以降は相互に接続される製品・サービスに着目し、そこにかかわる事業者などが、信頼性確認のために取り組むべき事項の整理を推進することとした。

3 品質説明力強化のための制度ガイドライン

日本では、品質文化が高く、高信頼な製品・サービス、及びそれを構成するソフトウェアを製造することは得意であるが、その高い品質を説明することは苦手と言われてきた。また、利用者にとって、ソフトウェアは直接的には見えない存在であり、その信頼性を事前に確認することは不可能に近い。このため、専門知識をもった第三者がソフトウェアの品質を確認し、その確認結果を利用者に提示する制度を構築することにより、開発現場での品質説明に向けた姿勢の強化と利用者への安心の提供に繋がると考えた。

信頼性の考え方の異なる製品・サービス分野に依存せず、前述のような観点から制度を構築するための要求事項を整理した「製品・システムにおけるソフトウェアの信頼性・安全性などに関する品質説明力強化のための制度構築ガイドライン」の案をとりまとめ、平成 24 年度末からのパブリックコメントを経た後、平成 25 年 6 月 12 日に第 1 版を公開した。同時に、本ガイドラインの適用第一号として CSAJ（Computer Software Association of Japan）の PSQ 認証制度（パッケージソフトウェアに関する品質認証制度）が発足し、年度末までに 9 社 13 製品が認証された。認証された製品企業からのヒアリングでは、品質説明の強化と同時に品質向上効果も報告されている。

4 先進設計・検証技術の適用事例紹介

利用者視点での信頼性向上を図るには、生産物としてのソフトウェアだけでなく、その開発プロセスの見える化や高度化に取り組む必要がある。それは、例えば不具合が見つかった時、その不具合混入の原因追求や試験実施の十分性確認につながり、関連する不具合の残存などを含めた総合的な信頼性を把握できることを意味している。そこで、開発プロセスのうち、最近「見える化」の手法が充実してきた「上流設計」、及び「検証」に着目し、その先進技術の普及を図ることとした。しかし、まだ現場レベルでは具体的な導入方法が不明で逡巡することが多いので、その参考として資する目的で、まずは先進的な取り組みを行っている企業での具体的な「導入上の工夫」、「生じた問題点と対策」などの導入事例をと

りまとめた。平成 26 年度 5 月末に事例報告書として公開した。

5 コンシューマデバイスの機能安全規格化

自動車、家庭用のサービスロボット、スマート家電、スマートハウスなどのコンシューマデバイスは、様々な環境で様々な利用者に異なった使われ方をするため、信頼性の確保が難しい。そこで、コンシューマデバイスを開発する事業者に参画いただいた WG で検討を進め、国際的な標準化団体である OMG（Object Management Group）に対して、開発方法論の国際標準案を共同で提案した。早ければ平成 26 年度に標準化最終段階に進む見込みである。

ソフトウェアグループ

ソフトウェア品質説明力の強化の促進
サプライチェーンにおけるソフトウェアの高信頼化

SEC 研究員　伊藤　克己
SEC 主任　八嶋　俊介

1 背景

近年、製品・システムの高機能化や多機能化が加速し、その実現手段として重要な役割を担うソフトウェアも急速に大規模化・複雑化している。

この状況に対応するためにソフトウェアの開発現場では、複数の事業者による分業化、ソフトウェア部品・OSS（open source software）の導入などによる開発手法・調達手法の多様化が進んでいる。更には、今までとは異なる業種・国籍などの開発事業者との関係構築なども拡大している。

また最近は、製品・システムなどが相互に接続されるサービスが構築されることにより、既存産業の変容・異業種の融合などが拡大している。このような新たな産業分野が進展することで、事業者の想定とは異なる利用形態・動作環境の下で、従来とは異なる新たな障害が生じる可能性が高まっている。

そこで、IPA/SEC では、利用者の安全・安心にかかわる信頼性の確保のため、ソフトウェア開発におけるサプライチェーンの課題と今後の取り組みの方向性について検討を行った。

2 調査及び WG 活動

2.1 ソフトウェア開発におけるサプライチェーンの課題の調査

ソフトウェア開発におけるサプライチェーンの課題を把握するため、製品・サービスなどの異なる 20 の業界団体、及び組込み系、エンタープライズ系、クラウドサービス基盤系、モバイルサービス系計 11 企業を対象に、ソフトウェア開発に関するヒアリングを実施し、従来から言われている「要求仕様・要件化の課題、受発注コストの課題、契約上の課題」などの相対取引における課題と、「品質文化の異なるシステム間の相互接続」に起因する課題に集約した。

これらの課題の詳細化のため、「ソフトウェア開発の取引構造（サプライチェーン）の実態に関わる課題の調査」として、基幹産業・輸出産業、新たな産業として今後の国際競争力が期待される分野、グローバル化・環境変化の速い分野、IT 融合における主要な構成要素となる分野、重要インフラ分野であるなどの視点から、自動車、スマート家電、ヘルスケア機器（サービス含む）、サービスロボット（移動支援、介護支援）、モバイル端末、クラウドサービス基盤などの 8 分野と、現在、各地域で行われているスマートコミュニティ実証実験を対象に文献調査、及びヒアリング調査を実施した。各分野での過去から現在・近未来に向けたサプライチェーンの変化を類型化し、そこに共通する新たな課題の整理を行った。

2.2 サプライチェーンにおけるソフトウェアの高信頼化 WG

調査と並行して「サプライチェーンにおけるソフトウェアの高信頼化 WG」を設置し、製造業に見られる国際的な水平分業化や、OSS やソフトウェア部品の流通などに見られる異分野間での水平分業化の進展により、垂

直統合型での信頼性確保から水平分業化での信頼性確保への移行にかかわる議論を行った。また、上記の調査結果に対して WG 委員から意見を聴取し、課題整理に反映することで、次に示す 3 つの課題を平成 26 年度以降の取り組みの方向性としてまとめた（図 1）。

3 次年度に向けた課題の整理

① ソフトウェア開発の分業化（図 1 左）

同一グループ内などで行われていた従来の垂直統合型開発から、グループ外からソフトウェアを調達する水平分業型開発に変化することによりサプライチェーンが複雑化・多層化して、トレースが難しくなり修正時間が長期化。

【例】 鉄道の運行管理システムにおいて、グローバル調達を実施。

【例】 欧州自動車業界のサプライチェーン。

② 供給部品ソフトウェア仕様決定者の変化（図 1 中央）

仕様決定の主体者が部品の調達者から供給者側に移行することにより、仕様変更や不具合修正の優先順位の認識ズレ、セキュリティにかかわるリスクが顕在化。

【例】 スマートフォンを代表とする Android OS 利用。

【例】 ヘルスケア機器を連携・統合したインターネットを介した健康管理サービス。

③ 製品・サービスのソフトウェアをユーザが組み合わせて利用（図 1 右）

個々の製品・サービスが独立に提供されることから、それらの組み合わせに関する最終的な品質を明確にできない。とくに、提供側の想定外の組み合わせで問題が発生したときの責任の所在が確定できず、また、利用者がリスクの存在を十分に理解していない。

【例】 インターネット接続によりモバイル端末からスマート家電を遠隔操作。

【例】 家庭向けエネルギー管理システム（HEMS）による複数機器の接続。

4 今後の取り組み

平成 26 年度以降の取り組みの方向性として、課題①での「ソフトウェア開発の分業化」に伴う責任分担に関しては、開発フェーズにおいて事業者間で解決すべき課題と判断。同時にそこで必要とされる信頼性確認のための技術的課題に関しては、後述の「先進的な設計・検証技術の適用事例」で設計・検証技術などの紹介をして

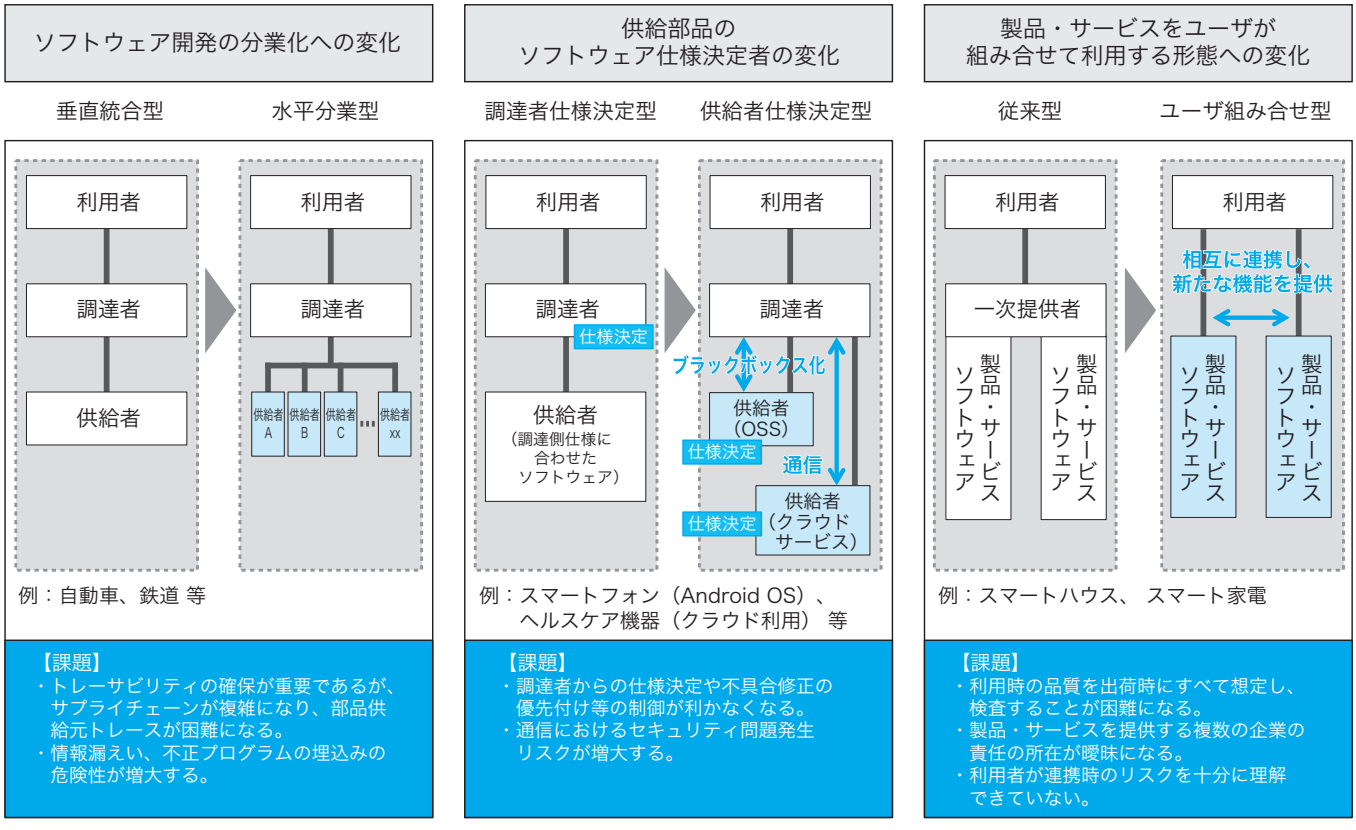


図 1 サプライチェーンにおける課題の整理

いる。

課題②及び③については、運用フェーズでの課題がとくに重要で、相互に接続される製品・サービスについて、サプライチェーンを構成する事業者などが Safety と Security の観点から信頼性確認のために取り組むべき事項を今後整理する。

ソフトウェアグループ

品質説明力の強化に向けた「制度ガイドライン」の活用

SEC 研究員
鈴木　基史

SEC 研究員
宮崎　義昭

SEC 研究員
細目　紀子

1 「制度ガイドライン」の概要

利用者が安心して製品、サービスやシステムを使えるようにするためには、その供給者が、高度化・複雑化する製品・システムの品質確保に努めると同時に、その品質について、利用者への十分な説明責任を果たす必要がある。

このための環境整備に向けた取り組みとして、平成 24 年度の活動において、第三者が客観的かつ専門的な立場から供給者の品質説明の適切性を、利用者の代わりに確認し、結果を利用者に理解できる形で提供する仕組みの導入を検討し、その成果を、「製品・システムにおけるソフトウェアの信頼性・安全性などに関する品質説明力強化のための制度構築ガイドライン」（以下、「制度ガイドライン」）として取りまとめ、その原案を公開した。このガイドラインは、製品・サービス分野に依存しない共通的な観点で、上記の仕組みを制度として構築する場合に、制度に責任を持つ組織や制度規定文書の記載内容に関する要求事項を整理したもので、制度構築を検討する組織にとっては、検討する上での指針となり、かつ複数制度の基本的な設計思想を合わせることで、制度のばらつきを抑制するという目的を持っている。

2 「制度ガイドライン」（第一版）の公開と普及活動の経緯

平成 25 年度は、「制度ガイドライン」の公開及び普及活動と、複数の団体への本ガイドラインに基づく制度構築支援活動を実施した。

■ 制度ガイドライン（案）に対するパブリックコメント（平成 25 年 3 月 29 日～平成 25 年 4 月 30 日）で寄せられた 31 件のコメントを踏まえ、正式版（第一版）を作成し公開。

■ 制度ガイドラインの公開と合わせて、制度ガイドラインの適用第一号である、パッケージソフトウェア製品

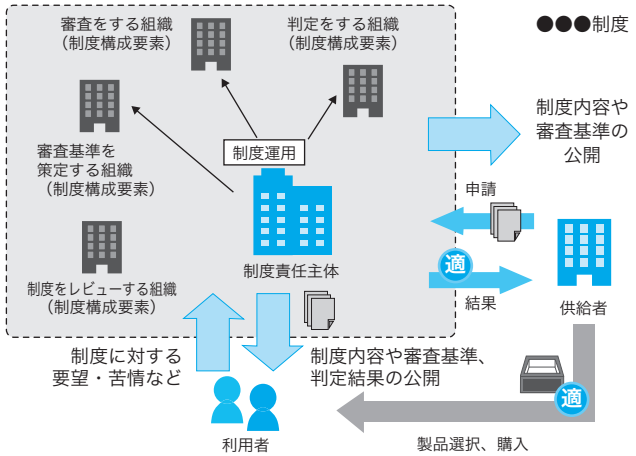


図１　個別制度開発・運用体制の例

写真１　制度ガイドライン公開（IPA）、及び PSQ 認証制度運用開始（CSAJ）の共同記者発表模様

に関する品質認証制度（PSQ 認証制度）の運用開始について、同制度を主管する CSAJ※1 と共同記者発表を実施（平成 25 年 6 月 12 日）。

■ PSQ 認証制度の強化を予定する CSAJ に加え、新たに 3 団体（SVA※2、IIOT※3、DEOS 協会※4）が計画する制度に関して検討、設計を支援。

表 1　制度ガイドラインに基づく制度化に向けた支援状況

団体名	各団体における品質説明関連の取り組み	IPA 活動（平成 25 年度）
SVA	家庭向けエネルギーマネジメントシステムの品質認証制度の枠組みを検討。モデルベース開発・検証技術に基づく審査が特徴	制度運用体制及び制度規定文書案の作成に関する支援
IIOT	モバイル機器などの相互接続性及びセキュリティに関する認証制度を検討中（平成 26 年度に認証基盤設計を計画中）	認証制度の企画及び基本設計に関する支援
DEOS	同団体の活動成果である標準規格（DEOS プロセス、D-Case、など）に関連した認証制度の構築を検討中	制度ガイドラインを紹介し、平成 26 年度に認証制度の枠組みを検討することを確認
CSAJ	パッケージソフトウェア向けの PSQ 認証制度を運用。標準規格（ISO/IEC 25051）の 2014 年の改定と同期して、PSQ 認証の SaaS/ クラウドサービスへの拡大を計画	CSAJ の制度関連委員会へオブザーバ参加

3 今後の予定

平成 25 年度、制度ガイドラインの普及促進を目的に、分野の異なる 24 の業界団体・機関などに対して、制度構築に対するニーズや課題についてヒアリングを実施した。その結果、利用者に対する品質説明や客観的な品質確認の仕組みが今後重要になるという点について賛同する意見が多かったが、一方で、利用者にとっての妥当な品質基準の定義やその検証技術の選択と適用が難しいという意見があり、今後の活動において、これらの課題の整理と解決策の検討を行う予定である。

【脚注】
※1　CSAJ（Computer Software Association of Japan）：一般社団法人コンピュータソフトウェア協会
※2　SVA（Smart System Verification and Validation Technology Association）：一般社団法人スマートシステム検証技術協会
※3　IIOT（international internet of things international interoperability testing）：一般社団法人 IIOT
※4　DEOS（Dependable Embedded Operating Systems）協会：一般社団法人ディペンダビリティ技術推進協会

ソフトウェアグループ

先進的な設計・検証技術の適用事例

SEC 調査役　室　修治
SEC 研究員　春山　浩行
SEC 研究員　藤原　由起子
SEC 研究員　佐々木　方規

1 はじめに

ソフトウェアの高信頼性を確保するためには、その開発プロセスに踏み込んで、見える化や高度化に取り組む必要がある。そのために、色々な開発方法論や手法が提案され、その有用性も報告されているが、現場レベルでは具体的な導入方法が不明であるとか、効果が見通せないとか、実システム開発における失敗が怖いなどの理由により、広範囲な実用にはつながっていないのが実情である。そのような中にあっても、先進的な手法を導入し、その改善に取り組み、実システムの開発にも適用するという実践的な事例が多く存在する。

利用者が安心して製品・システムを使えるようにするために、これらの事例を数多く収集し、それを共有することにより、我が国のソフトウェア開発の水準を高めていくことが重要であると考える。そこで、開発プロセスのうち、最近「見える化」の手法が充実してきた「上流設計」、及び「検証」に着目し、その先進技術の普及を図ることとした。これまでに、先進的な取組みを行っている団体・企業などから適用事例を収集してきた。

これらは、平成 26 年度 5 月に「先進的な設計・検証技術の適用事例報告書 2013 年度版」として公開した※1。

今後は、更に事例数を増やし、最終的には収集した事例を俯瞰して、有用性や適用領域を整理したガイドラインを作成する予定である。

【脚注】
※1　http://www.ipa.go.jp/sec/reports/20140530.html

2 今回収集した事例の特徴

① 各方面のご協力をいただき、20 件以上の事例を収集でき、多岐にわたる内容になっている。
② 設計・検証に関する先進的な適用事例を紹介したものであり、導入上の工夫や効果なども記載しており、今後の導入の参考になるものである。
③ 各社、各団体の現場で活躍されているリードオフマン的な方々に紹介していただいた事例であり、高度な内容も含んでいる。

3 収集した事例の一覧

今回収集した事例を、設計と検証に大別して、表 1 に示す。

4 普及に向けた活動

有用な技術や手法の普及のために、以下の 2 つの観点でセミナーを開催している。

① 導入への関心を高めるための事例紹介セミナー
② 具体的な技術や手法に関する技術セミナー

昨年度は、ソフトウェアの高信頼化に関連した技術（MBSE、形式手法など）に関するセミナーを開催した。

今後も事例紹介セミナーとその事例に適用されている技術に関するセミナーを企画・開催していく予定である。

知識習得の場として、ぜひ活用していただきたい。

5 事例紹介

以下に、今回収集した事例をいくつかピックアップし、その概要を紹介する。

（1）「アシュアランス技術を用いた鉄道信号の革新」（事例提供元　東日本旅客鉄道株式会社）

現行の列車運行を止めることなく、運行管理システムを刷新していく取り組みにおいて、システムの安定稼働を保証する技術（アシュアランス技術）をいかに適用したかを解説している。とくに、既に稼働中の装置と試験中の装置が混在する状況において、すべてのデータの中から自律分散的に自身が扱うべきデータを判断し、他に影響を与えることなく稼働可能としている点に特徴がある。

日本の列車制御の安全性は、世界に誇れるものであり、それを実現する技術の一端を興味深く知ることができる。

（2）「要件定義段階における信頼性向上の取り組み事例紹介」（事例提供元　ビッグローブ株式会社）

コンシューマ向けサービス開発や受託システム開発においては、短期開発、コスト削減の条件の元で品質確保を行わなければならない。この問題を軽減するのに効果的だった要件分析手法、優先度付けの方法、ツールの活用方

表 1 「先進的な設計・検証技術の適用事例報告書 2013 年度版」掲載事例一覧

	No.	標題	事例提供元
設計事例	A-1	アシュアランス技術を用いた鉄道信号の革新	東日本旅客鉄道株式会社
	A-2	XDDP 導入による派生開発の品質改善とその効果	株式会社日立産業制御ソリューションズ
	A-3	組込み系の利用品質における「HMI 品質メトリクス」開発と適用事例	株式会社 U'eyes Design
	A-4	要件定義段階における信頼性向上の取り組み事例紹介	ビッグローブ株式会社
	A-5	要件定義の品質向上に向けた取り組み	富士通株式会社
	A-6	ジェネレータツールを利用した高信頼開発、高速開発の実践	株式会社市進ホールディングス
	A-7	設計工程における TERASOLUNA DS の適用	株式会社エヌ・ティ・ティ・データ
	A-8	Grails/Groovy の適用推進	エヌ・ティ・ティ・ソフトウェア株式会社
	A-9	個人依存開発から組織的開発への移行事例	三菱電機メカトロニクスソフトウエア株式会社
	A-10	MBSE による双腕作業ロボット動作実行系のコンセプト設計	独立行政法人産業技術総合研究所（AIST）
	A-11	仕様記述言語 VDM++ を用いたシステムの仕様の記述	フェリカネットワークス株式会社
	A-12	車載 ECU 開発における上流工程での品質確保	東芝情報システム株式会社
	A-13	独自開発したモデル駆動開発プロダクトラインエンジニアリングの実践	株式会社デンソー
検証事例	B-1	宇宙システムにおける上流工程仕様の妥当性確認技術	独立行政法人宇宙航空研究開発機構（JAXA）
	B-2	鉄道の機能安全（RAMS）認証支援のためのセーフティケース	独立行政法人産業技術総合研究所（AIST）
	B-3	非機能要求グレードの大学ポータルサービスへの適用	名古屋大学
	B-4	冗長構成システム（クラウドなど）の耐故障性に対する検証技術	株式会社富士通コンピュータテクノロジーズ
	B-5	単体ランダムテスト実行／可視化ツール “Jvis” の適用事例	宮崎大学
	B-6	要求仕様明確化のための仕様記述技術（USDM）活用事例	株式会社ベリサーブ
	B-7	形式手法を用いたセキュリティ検証	アーク・システム・ソリューションズ株式会社
	B-8	形式仕様記述手法を用いた高信頼性を達成するテスト手法とその実践	フェリカネットワークス株式会社
	B-9	Orthogonal Defect Classification 分析による欠陥除去と品質の成熟度可視化	オリンパスソフトウェアテクノロジー株式会社
	B-10	モデル検査の適用による上流工程での設計誤りの発見	株式会社東芝
	B-11	上流工程の要求を効率的に閉ループシミュレーションする次世代 SILS の開発	トヨタ自動車株式会社

法などを実践的な工夫を交えて具体的に紹介している。

これまでの常識とされた考え方に捉われることなく、現場感覚が垣間見える内容であり、セミナー講演を聞いた受講者からは、「目から鱗（うろこ）であった。」という感想も寄せられている。

（3）「仕様記述言語 VDM ＋＋を用いたシステムの仕様記述」（事例提供元　フェリカネットワークス株式会社）

おサイフケータイ※2 に搭載された IC チップファームウェアの開発にあたり、形式仕様記述手法を適用した効果を紹介している。この手法を導入しても、プログラム開発効率の低下は見られず、仕様に関するコミュニケーションの向上などの成果が得られたことを示している。

自然言語による仕様記述の方が、むしろ難しく感じる人がいるという実態報告にはうなづけるものがある。技術者にとっては数学的表現と国語的表現のいずれが理解しやすいのかと考えさせられるような問題提起を含んでいる。

（4）「宇宙システムにおける上流工程仕様の妥当性確認技術」（事例提供元　独立行政法人宇宙航空研究開発機構（JAXA））

宇宙システムのようなセーフティクリティカルシステムのソフトウェア開発では、上流工程の仕様定義での問題 (不正確、不完全、想定不足など) がシステムの安全性を脅かすことが多い。本事例では、問題解決のため導入適用した「モデル検査」と「チェックリストベースレビュー」について、特徴や導入上の課題、解決策を紹介している。

宇宙システムは、我々の一般的なソフトウェアとは縁遠いものと考えがちだが、ソフトウェアの開発現場ではレベルの高低はあれども、悩みは同じであり、特別なやり方が行われているわけではない。そういう意味で、ソフトウェアの信頼性を追求する活動は、大いに参考になるものである。

（5）「形式手法を用いたセキュリティ検証」（事例提供元　アーク・システム・ソリューションズ株式会社）

組込み機器のソフトウェア開発において、セキュリティ要件を満たすために形式手法を適用した事例を紹介している。形式手法は、数学的な証明を与えてくれるため、客観的な観点でセキュリティの保証につながること、及び形式記述することにより曖昧性を排除できることに特徴がある。

適用効果を示すために、セキュリティの専門家に独立に分析してもらい、形式手法適用ツールを利用した場合の脆弱性指摘との違い（指摘箇所は同じ）までも考察しており、効果への見通しの参考になる内容である。

6 おわりに

今回の事例収集の取り組みは緒に就いたばかりであり、比較的大規模なソフトウェア開発が対象であったり、先端的な事例に着目している面がある。しかし、実際の開発現場では、もっと実状に即した対応がなされていたり、幅広い取り組みが存在すると想定される。それらを各方面のご協力を得ながら収集し、更に充実したものに仕上げていきたい。

【脚注】

※2　「おサイフケータイ」は株式会社 NTT ドコモの登録商標です。

ソフトウェアグループ

コンシューマデバイス機能安全規格化の提案のコンセプトと取り組み

SEC 研究員　春山　浩行
SEC 研究員　内田　功志
SEC 調査役　室　修治

1 はじめに

一般消費者向けのシステムを対象とした製品群を“コンシューマデバイス（消費者機械）”と呼び、自動車、サービスロボット、スマート家電、スマートハウスのようなものが該当する。

コンシューマデバイスには高いディペンダビリティ※1 が求められ、既に自動車に関しては ISO 26262 でその機能安全規格が定められている。しかし、これは自動車に特化したものであり、あらゆるコンシューマデバイスに

適用できるものではなかった。そこで、あらゆるコンシューマデバイスに横断的に適用できる体系化された効率的な枠組みとして、Dependability Assurance Framework（以下「DAF」）をOMG※2に2013年11月に初期提案した。

2 提案のコンセプト

ISO 26262が、電気・電子システムの機能安全のための規格であることと対比して今回の提案の特徴を述べると、次の通りである。

（1） 対象をコンシューマデバイス全般に広げたこと

特定な分野に偏らない様に図1の分野（自動車、サービスロボット、スマート家電、スマートハウス 他）の産官学の有識者及びディペンダビリティや標準化の専門家に集まっていただき汎用的な提案になるようにした。

（2） 規格範囲をディペンダビリティに広げたこと

代表的なコンシューマデバイスである自動車は、安全性と可用性を同時に満たす必要がある。今後、更にコンシューマデバイスはより高機能で複雑化するため、安全性に可用性や信頼性、保全性を含んだディペンダビリティを保証しなければならなくなる。今回の提案は、様々な利用者と様々な環境の下で利用されることを想定したものである。

（3） 汎用的な開発方法論にまで踏込んでいること（図2）

ディペンダブルなコンシューマデバイスを実現するためには、開発当初からいかにディペンダビリティを保証するか考慮する必要がある。そこで高品質なコンシューマデバイスを実現するために体系的な枠組み（開発方法論）を提案しており、とくに日本流の摺り合わせ開発による品質の作り込みを反映したものにしている。また、ディペンダビリティを保証しつつ、効率的に開発できる枠組みとしてモデルベース開発やシステムエンジニアリングの技法や手法も取り入れている。モデルベースの開発をすることで想定と試行の素早い繰り返しによる改善が有効になる。

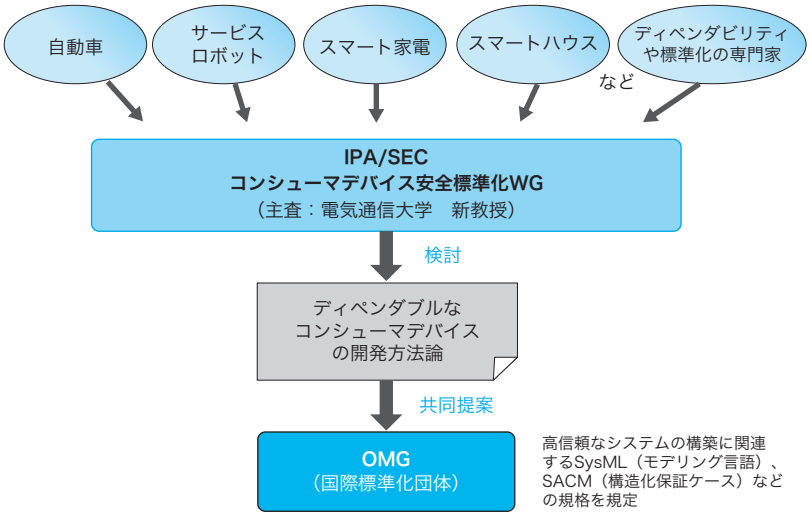


図1　コンシューマデバイス安全標準化WG

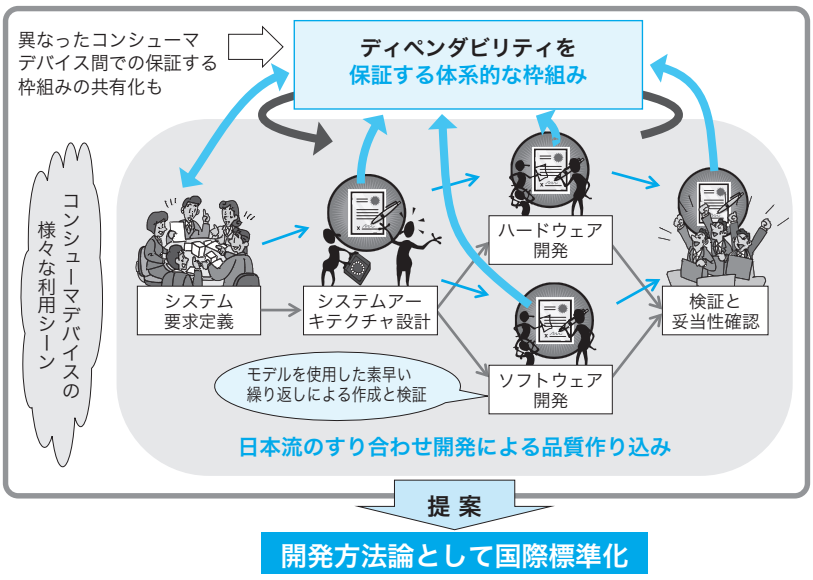


図2　ディペンダビリティを保証する体系的な枠組み

3 おわりに

平成25年度は、9月に『コンシューマデバイスの信頼性確保に向けた取組み～開発方法論の国際標準化に向けて～』を公開し、平成26年3月にOMGにて本提案の変更提案の中間レビューを実施した。

平成26年度は、6月に本提案の変更提案を提出し、採択の可否を投票によって確定し、9月に標準化最終段階に進む予定である。

【脚注】

※1　ディペンダビリティ：信頼性性能、保全性性能及び保全支援能力を記述するために用いられる包括的な用語（JIS Z 8115:2000）

※2　OMG:Object Management Group

ソフトウェア工学分野の先導的研究支援事業について

SEC 調査役
小沢　理康

IPA/SEC では大学等における、主としてソフトウェア工学分野の研究活動の活性化を図る目的で、「ソフトウェア工学分野の先導的研究支援事業」を 2012 年度より実施している。大学・公的研究機関から研究提案を広く公募しており、2013 年度は 5 件を採択し、そのうち研究期間が単年度の研究 2 件については、成果報告書を IPA/SEC が、また研究成果であるソフトウェア開発現場で利用可能な支援ツールを大学が公開した。2014 年度の公募では 21 件の応募があり、このうち 4 件を採択した。本稿ではその事業概要と、2013 年度事業の成果の一部と、2014 年度事業の公募における採択状況について報告する。本事業は 2017 年度までの継続を予定しており、次年度以降も公募を実施する予定である。

1. 研究支援事業の概要

ソフトウェアは、あらゆる産業や市民生活を支える基盤として不可欠な存在となってきており、複雑化・大規模化するソフトウェアの高信頼化や開発プロセスの高度化、またそれらの運用や保守についても様々な課題が存在している。このような課題に対して工学的なアプローチで解決策を提供しようとするソフトウェア工学や複雑な統合システム（System of Systems）へのシステム工学の適用に係る研究や、ソフトウェアの経済的効果に関する研究についての一層の振興をねらいとして本事業を実施している。

本事業では、ソフトウェアの開発・利用の現場に密着した研究を重視しており、開発現場と連携した研究を促進するため、産業界の有識者から成る「ソフトウェア工学研究推進委員会」を設置し、公募内容を決め、研究テーマの選考と研究に対する助言も行いながら実施することとしている。

表 1　2013 年度に採択した研究

区分*	期間	研究テーマ名	提案者名
A	1 年	次世代ソフトウェア信頼性評価技術の開発とその実装	国立大学法人 広島大学
B	1 年	抽象化に基づいた UML 設計の検証支援ツールの開発	公立大学法人 岡山県立大学
A	2 年	ソフトウェア品質の第三者評価のための基盤技術 ーソフトウェアプロジェクトトモグラフィ技術の高度化ー	国立大学法人 奈良先端科学技術大学院大学
A	2 年	IPA EPM-X の機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築 ―次世代の定量的プロジェクト管理ツールとリポジトリマイニング研究基盤―	国立大学法人 和歌山大学
B	2 年	形式仕様とテスト生成の部分的・段階的な活用 ～探索を通したコード中心インクリメンタル型開発の支援	大学共同利用機関法人 情報・システム研究機構 国立情報学研究所

* 公募した研究分野で A 区分は「ソフトウェア工学分野の先導的な研究」、B 区分は「ソフトウェア開発現場へのソフトウェア工学の適用に関する研究」

2. 2013 年度事業の成果

2013 年度は 7 件の応募の中から 5 件の研究テーマを採択し、これを支援した（表 1 参照）。研究期間は単年度（2013 年 6 月から 2014 年 2 月まで）と 2 年度（2013 年 6 月から 2015 年 2 月まで）の 2 種類としている。

単年度の 2 件については研究期間が今年 2 月で終了し、これらの成果報告書を IPA/SEC が公開すると共に、委託研究先である岡山県立大学と広島大学より、ソフトウェア開発現場で利用可能な支援ツールをそれぞれの大学でフリーウェアとして公開した。成果報告書のダウンロード及びフリーウェアの紹介ページへのリンクは以下の URL を参照いただきたい。

http://www.ipa.go.jp/sec/reports/20140430.html

委託研究が終了した単年度の 2 件の研究成果の概要を以下に示す。

◎次世代ソフトウェア信頼性評価技術の開発とその実装（国立大学法人広島大学）

従来のソフトウェアの定量的信頼性評価を詳細なデータ（統計量）を扱えるソフトウェア信頼性モデルへと拡張を行うことで、開発現場で獲得し得る情報水準に応じて信頼性評価の方法を分類する次世代のソフトウェア信頼性評価技術の体系化を目指した。

研究成果である信頼性評価技術は Excel の add-in として実装した。Excel インターフェースを利用しているため、企業でも手軽に信頼性評価を行うことができる。ソフトウェアメトリクス、テスト入出力情報、ソースコード情報からソフトウェアの信頼性を特徴づける情報を抽出し、精度の高い定量的信頼性評価を実現することが可能となっている。同様の信頼性評価技術を統計処理ソフトウェア R のパッケージとしても開発した。ユーザが R 言語を用いて拡張することができるため、実験的なデータ解析や大量のデータ解析を行う研究者に向いている。

◎抽象化に基づいた UML 設計の検証支援ツールの開発（公立大学法人岡山県立大学）

組込みソフトウェアの設計検証において検証モデルの作成をツールによって支援することで、モデル作成の困難さ並びに検証時間の増加といった問題を解決し、開発現場へのモデル検査の導入促進を目指した。

成果として、設計記述から検証モデルを自動的に作成するための検証支援ツールを開発した。本ツールは、組込みソフトウェアの開発において広く利用されている形式仕様記法である UML で記述された設計を対象として、モデル検査ツール NuSMV[※1] の検証モデルを自動的に作成する。本ツールを用いることで、大きな学習コストを要することなく NuSMV を用いた設計の自動検証を実施することが可能となる。また、検査する性質に関連する部分の抽出や検証に影響しない部分の抽象化を自動的に行った上でモデルを作成することで、状態爆発による検証時間の増加を回避することが可能である。

表２　公募した研究分野及び区分

区分	分野名	分野の概要説明
A	ソフトウェア工学分野の先導的な研究	要求工学、プロセス改善、高信頼性、アジャイル開発、形式手法、モデルベース開発等のソフトウェア工学分野の先導的な研究
B	ソフトウェア工学・システム工学の実践的な適用に関する研究	ソフトウェア開発現場への適用を目的としたソフトウェア工学の成果・手法を詳細化・具体化・実用化する研究またはスマートコミュニティ、ヘルスケア、ロボット、次世代自動車と交通システム等の複雑な統合システム（System of Systems）の研究開発において、ソフトウェア工学・システム工学の成果・手法を適用する研究
C	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研究	ソフトウェアが社会や組織経営にもたらす経済価値、生産性向上、競争力強化、イノベーション等の経済効果についての実証
D	ソフトウェア工学に関する課題指定研究	「ソフトウェア開発データの分析」IPA/SEC が過去９年間にわたり収集・蓄積してきたソフトウェア開発データを新たな視点や手法で分析・研究することにより、ソフトウェア開発における課題や方向性を提唱する研究

また 2014 年５月 22 日には、支援事業で実施した研究の成果を広く産業界へ普及展開するための「第２回産学連携のためのソフトウェア・シンポジウム」を開催した。このシンポジウムは、前半の各大学による研究成果の講演と、後半の各大学の個別ポスターセッションで構成した。シンポジウムの講演資料のダウンロードと講演動画の閲覧は以下の URL のリンク先から参照できる。

http://sec.ipa.go.jp/seminar/20140522.html

3. 2014 年度公募の状況と採択結果

2014 年度の公募に際しては産業界の課題等に対応するため、対象となる研究分野を拡大、見直した（表２参照）。

2014 年度の公募に際しては大学や関連学会等への周知に力を入れたことにより、前年度の３倍にあたる 21 件の応募があった。これらの提案については、ソフトウェア工学研究推進委員会において厳正な審査を行い、４件の研究提案の採択を決定した（表３参照）。区分 C は本事業開始３年目で初めての採択となった。

表３　2014 年度採択研究提案一覧

期間	区分	研究テーマ名	提案者名
1年	B	保守プロセスにおけるモデル検査技術の開発現場への適用に関する研究	学校法人芝浦工業大学
2年	A	オープンシステム・ディペンダビリティのための形式アシュランスケース・フレームワーク	学校法人 神奈川大学
2年	B	システムモデルと繰り返し型モデル検査による次世代自動運転車を取り巻く System of Systems のアーキテクチャ設計	学校法人慶應義塾 慶應義塾大学 大学院
2年	C	日本のソフトウェア技術者の生産性及び処遇の向上効果研究：アジア、欧米諸国との国際比較分析のフレームワークを用いて	学校法人同志社 同志社大学

【脚注】

※１　モデル言語の表現力と検証速度に優れたモデル検査ツール

米国 NIST、SEI、WVU を訪問して

SEC 所長
松本　隆明

SEC 研究員
鈴木　基史

SEC 研究員
石田　茂

2014 年 3 月 3 日（月）から 3 月 7 日（金）にかけて米国の NIST※1（米国商務省国立標準技術研究所）、SEI※2（カーネギーメロン大学ソフトウェア工学研究所）、WVU※3（ウエストバージニア大学）の三ヵ所を回り、新しい知見を得ることができた。しかし今回は北米東海岸が雪のため、幾つかのアクシデントがあり、読者のみなさんには、これらアクシデントも含めて報告をしたい。

行きは、成田からワシントン D.C. のダレス空港に入る行程だったが、ワシントンの雪のため飛行機の出発が 2 時間遅れた。最初の訪問地であるワシントン D.C. のダレス空港に着くと一面雪景色であった。ダレス空港ではモバイル・ラウンジと呼ばれるアメリカ的な乗り物に乗って、イミグレーションのあるターミナルに移動した（写真 1）。中はラウンジと言っても部屋の中に椅子があるだけで、部屋ごと移動するという乗り物である。

1．出張の目的

今回の出張の目的は以下の 2 点である。

① NIST 及び SEI とは最新の情報を交換することにより、今後のテーマ連携を探ること。

② WVU 及び SEI とは IPA/SEC で進めている先進設計・検証技術の海外事例収集を行うこと。

2．NIST

1 日目の NIST は、メインゲートのゲストセンターでの受け付けのため、順番待ちでかなり並んだ。当日は NIST でセキュリティのカンファレンスがあり、その関係で受け付けが非常に混んでいたが、順番になって、目的の研究所が雪のためまだオープンしていない事が分かり、いったんホテルに戻り、また出直すという事になった。後で分かったことだが、前日は雪のため NIST 自体が閉鎖になっており、当日も雪のために職員の出勤が 10 時からに変更になっていた。今回は初日会議開始が遅れるというトラブルからのスタートとなった。

写真 1　飛行機とターミナル間を結ぶモバイル・ラウンジから（ダレス空港）

1）Software Testing

SSD※4 傘下、Software Quality Group のバーバラマネージャ及び ICES 社のエスワラン氏から SAMATE※5 プロジェクトに関する活動として、セキュリティを含んだ 8 万ケースの不具合のあったプログラムをコードの単位で SRD※6 として集積・公開しているという報告があった。これに基づいて SATE※7 という 2600 万行になるテストケースが、実際の C/C++ や Java、PHP のコードより作成されている。利用者は自分のツールにこのテストケースを適用することにより、ツールの脆弱性と危険なコードが入っていないかをチェックすることができるという報告があった。

2）Robotic Systems

IPA/SEC から最近のホットな話題としてロボットと車の自動運転に関する議題案を上げたところ、EL（エンジニアリング研究所）※8 傘下の ISD※17 のウェイヴァリング部門長から製造用途におけるロボットシステムの取り組みの紹介を頂いた。

また災害対応ロボット※9 を市場から調達する際の、標準制定の取り組みも進めており、DARPA（米国国防高等研究計画局）※10 プロジェクトにおいても専用試験用設備を用いて試験手法の共同開発を推進するなどの活動を行っているとのことだった。

NIST 側は産業用ロボットが中心で機能、性能のメトリクスを作っていたが、セーフティ等の安全に関するメトリクスは使っていないことが分かった。IPA/SEC からは NEDO※11（独立行政法人新エネルギー・産業技術総合開発機構）の生活支援ロボット実用化プロジェクトを紹介し、興味を持ってもらえた。車の運転支援・自動運転システムについての活動はないようだが、AGV※12 の自動センシング等では共通項があるように感じた。

3．SEI

2 日目、3 日目の SEI では、ワシントン DC からピッツバーグまでの移動は雪の影響もほとんどなく、スムーズに行えた。SEI では、コンラッドマネージャの好意により、8 人のプレゼンターを集めて頂き、非常に密度の濃い内容と

なった。最初に IPA/SEC から SEC 名称と事業内容の変更の説明を行ったが、SEI からも昨年の組織変更で、CERT※13 Division, Software Solutions Division, Emerging Technology Center（40 人程度）の 3 組織になったとの説明があった。

1) Software supply chain

CERT のウッディー氏（コンピュータ緊急対応センターのセキュリティの専門家）から「Cyber Security Engineering and Security Risk」というタイトルで、サプライ・チェーン上でのセキュリティの分析手法に関する紹介が行われ、それに続き IPA/SEC もサプライ・チェーンのセーフティに関する紹介を実施した。ウッディー氏からも従来のソフトウェア・アシュアランスに加えシステム・アシュアランスを入れる必要があり、サプライ・チェーンに沿ったトレーサビリティはセーフティとセキュリティの両方で必要であるとのコメントがあった。

2) The real application example of advanced technology

IPA/SEC から、先進設計・検証技術の事例収集の取り組み紹介を行い、米国での事例収集のお願いを実施した。 またドゥランジュ氏からアーキテクチャを中心とした仕様とアーキテクチャを結ぶための AADL（言語）の紹介があり、ミーティングのラップアップ時にドゥランジュ氏の関係しているものが事例に近いという事で、後日、情報を頂いた。

3) Team Software Process

ニコルス氏と白井氏から「Benchmarking Software Development Performance Using TSP Data」というタイトルで TSP※14 に関する報告があった。CMMI※15 は組織が取り組むプロセスであり、TSP はチームで取り組むプロセスである。また PSP※16 は個人で取り組むプロセスについて記述されたものである。TSP では、ツールを使って個人の作業時間を計測して、それを蓄積してプロセスの分析、改善に利用している。TSP と CMMI のソフトウェア品質を比べると CMMI 導入プロジェクトよりも TSP を導入したプロジェクトのソフトの品質の方が高いという報告がなされた。またセキュア TSP というセキュリティを入れた報告もあった。TSP のデータ収集は、ソフトウェアのサポートなしには実現できないため、TSP ソフトウェアツール（Software Process Dashboard 等）が提供されている。またこのツールを使った TSP データベースも 2009 年より構築、運用されており、既に 109 のプロジェクトのデータが収集されているという報告があった。IPA/SEC の保有するデータ白書の情報と TSP のデータで日米比較をしてみると何か分かるかもしれないということで、後日情報交換を行うこととした。

4) Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE)

ファーガソン氏より、将来コストに何が跳ね返って来るかを予測するコスト分析手法についての紹介があった。これは、環境変化要因をどう抽出するかという点で、IPA/SEC の障害要因分析に適用できる可能性がある。

4．WVU

キューキック教授からは、Firefox や Eclipse プロジェクトの自然言語で書かれた 40 万件のバグ情報レポートを分析（基本的にテキストマイニング）して同じようなバグが多いという分析結果報告があった。また NASA IV&V※18 Facility のダウンズマネージャ、及び WVU のメンジーズ教授に IV&V の最新情報や、最近の取り組み内容、民間企業との共同研究、技術移転の枠組みなどについて紹介いただいた。

当初は、WVU の横に併設されている GSFC※19（ゴダード宇宙飛行センター）の IV&V Facility の見学を事前依頼していたが、NASA は 3 ヶ月前の事前登録が必要で見学できなかった。そのため WVU のカンファレンスルームに NASA のダウンズマネージャに来て頂き、最近の IV&V の動向として、NASA のスペースプログラムだけではなく、「ニューヨーク市の緊急通信変換プログラム」にも IV&V が使われていることを紹介頂いた。

5．おわりに

今回の出張目的である、今後のテーマ連携を探るという点に関しては、最新の情報を交換することにより、上記に記したような新たな知見を得ることができ、SSD のスリイラムチーフから頂いたヘルスケアなど産業領域を特定した活動連携の打診や、SEI のソフトウェア・サプライ・チェーンという IPA/SEC と同じテーマの調査・研究等、連携につながるテーマがあることが分かった。

また先進設計・検証技術の海外事例収集に関しては、SEI から関連する具体的な事例を送って頂くことができた。

今回は、ソフトウェアやシステムの信頼性・安全性という観点から意見交換を行ったが、SEI では CERT 部門の担当者が参加するなど、NIST、SEI ともセーフティとセキュリティは一体で考える必要がある点を強調していた。SEC も IPA のセキュリティセンターとの連携を更に密にして事業を進めていく予定である。

【脚注】

※ 1　National Institute of Standards and Technology
※ 2　Software Engineering Institute
※ 3　West Virginia University
※ 4　Software and Systems Division
※ 5　Software Assurance Metrics And Tool Evaluation
※ 6　SAMATE Reference Dataset
※ 7　Static Analysis Tool Exposition
※ 8　Engineering Laboratory
※ 9　Emergency Response Robots
※ 10　Defense Advanced Research Projects Agency,
※ 11　New Energy and Industrial Technology Development rganization
※ 12　Automated Guided Vehicle
※ 13　Computer Emergency Response Team
※ 14　Team Software Process
※ 15　Capability Matunity Model Integration
※ 16　Personal Software Process
※ 17　Intelligent Systems Division
※ 18　Independent Verification and Validation
※ 19　Goddard Space Flight Center

IPA/SEC における国際連携の推進
（オランダ TNO-ESI、英国 MISRA 編）

IPA/SEC 副所長
杉浦　秀明

SEC 調査役
十山　圭介

1．はじめに

IPA/SEC は、組込みシステムを含む情報処理システムの高信頼化の取り組みについて、国際連携強化を推進している。その一環として、2014 年 3 月に組込みシステムの研究開発で産学協働の活動を行っているオランダ TNO-ESI、及びこれまで C 言語のコーディングガイドラインで協力関係にある英国 MISRA を訪問し、双方における最近の取り組み状況、今後の連携強化に関する意見交換と議論を行った。オランダ TNO-ESI へは、今回が IPA/SEC として初めての訪問である。

2．オランダ 組込みシステムイノベーション（TNO-ESI）

1）組織概要

オランダでは、組込みシステムに重点を置いた革新的な研究開発やエンジニアリングへの取り組みが産学の協働の下に行われている。その活動の中核的な役割を担っている組織が、オランダ応用科学研究機構（TNO；Netherlands Organization for Applied Scientific Research）の組込みシステムイノベーション（ESI：Embedded Systems Innovation）（以下、TNO-ESI と記載する）である。

TNO は、欧州では最大規模を誇る中立・独立の総合研究機関であり、科学技術分野における応用科学研究を行うことを目的としてオランダ議会によって 1932 年に設立され、産業界の継続的な競争力強化とイノベーションの創造をミッションとしている。TNO の試験研究部門は、①クオリティオブライフ部門、②防衛公衆安全部門、③科学・産業部門、④環境・地球科学部門、⑤情報通信技術部門の 5 部門で構成されている。研究スタッフ総数は、約 4,000 名としている。

写真 1　アイントホーフェン工科大学内の TNO-ESI 本部棟

ESI は、2002 年にオランダ政府のファンドと民間等のファンドにより設立された。2013 年 1 月より、TNO 傘下の組織（TNO-ESI）となった。TNO-ESI は、組込みシステムの研究・開発に特化した活動を行っており、参加企業はヘルスケア・通信・エネルギー・交通・モバイルなど、幅広い産業分野にわたる。ESI は、分野横断的に横串を刺すソリューション提供を目指して、産業界とアカデミアのハブの役割を果たしている。こうした産学協働の研究開発推進方式を、「Industry-as-laboratory」、「Industry-as-classroom」と称している。

TNO-ESI の本部は、アイントホーフェンのアイントホーフェン工科大学内に置かれている。

2）「組込みシステムロードマップ 2014」

TNO-ESI の活動は、毎年リバイズを行っている組込みシステムロードマップに従って行われる。このロードマップは、優先して取り組む分野を方向づけるものとなっている。現時点での最新版は、2013 年 12 月に公表された「HTSM Roadmap Embedded Systems 2014」である。このロードマップは、オランダのアカデミア、産業界、TNO-ESI 等から約 40 名の関係者が関与し策定された。本ロードマップでは、産業界のニーズから導かれた重点課題として、例えば、System of Systems のモデルベース・アーキテクチャ及びエンジニアリング手法、分野横断的なシステムの互換性や相互作用、ディペンダビリティとシステム品質、強力な設計モデリングツールなどを挙げている。また、欧州委員会の研究投資計画である HORIZON 2020 や EUREKA における連携も進めていくとしている。

3）システム高信頼化部門の取り組み（モデルベースによるテスト・検証等）

TNO-ESI では、優先して取り組むべき重点分野として、

①システム・パフォーマンス、②システム品質と信頼性、③次世代持続型システム（Future-proof systems）、④変化適用型システム（Systems in context）の 4 分野を位置付けている。

このうち②のシステム品質と信頼性を担当するシステム高信頼化部門では、組込みシステムの品質・信頼性の向上にフォーカスし、モデルベース・テスティングやモデルベース障害診断など、多くのプロジェクトを実施している。とくにユニークな取り組みとして、「テストベース・モデリング」と呼ばれるモデルベース・テスティングの逆の流れでテストによってモデルの自動生成を行う活動が挙げられる。また、「スペクトラムベース・フォールト・ローカリゼーション」といった新たな手法の産業適用についての取り組みも実施している。

4）IPA/SEC と TNO-ESI との連携

今回が IPA/SEC として初訪問となった TNO-ESI では、上記の通り組込みシステムの中でもとくに産業界とアカデミアの参画を得たモデルベースに関連した取り組みが重点的に進められている。こうした取り組みから得られた知見は、弊機構における重要インフラにおける高信頼化対策などに関連した活動を進める上でも有用であるため、今後、TNO-ESI との間で継続的に連携を図ることとしたい。

3．英国 MISRA

1）組織概要

MISRA（The Motor Industry Software Reliability Association）は、自動車用システムにおける安全で信頼性の高いソフトウェア開発手法の普及を目指す、自動車メーカ、部品メーカや研究者から成る欧州の自動車業界団体である。

MISRA の活動は、1990 年代初頭に英国政府の「SafeIT プログラム」と呼ばれるプロジェクトとして始まった。1994 年には、「MISRA ガイドライン」として知られる自動車向けソフトウェア開発ガイドラインが策定された。英国政府の資金提供が終了した後、今日に至るまで、MISRA メンバーの活動は自己資金により継続している。こうした活動を通じて得られた成果として、世界を席巻する国際的なソフトウェア設計標準規格として広く利用されている MISRA C がある。MISRA の主な活動は、企業に所属する委員によってワーキンググループを構成して進められており、自動車にとどまらずクリティカル・システム全般の高信頼化に向け、C 言語や C++ 言語のコーディングガイドラインをはじめとする設計規準を発行している。ワーキンググループ活動も、航空、医療、エネルギー、通信、鉄道、自動車等からメンバーが参加しており、こうした活動を通じて得られる成果は全産業に適用可能なものとなっているとしている。

MISRA のオフィスは、そのステアリングコミッティー企業である MIRA Ltd. に置かれ、ロンドンの北西約 160Km のバーミンガムの東に位置するヌニートンに置かれている。

2）IPA/SEC における ESCR C 言語版の改訂

MISRA と IPA/SEC は、これまで IPA/SEC による ESCR 策定やその JIS 化におけるルールの引用などで協力関係にある。今回、IPA/SEC からは、2014 年 3 月に発行した ESCR C 言語版の改訂（IPA/SEC が 2007 年に発行した ESCR Ver.1.1 の改訂版で、新たに近年開発現場で使用されることの多くなった C 言語の最新 JIS 規格「C99」に対応したもの）について詳細な説明及び討議を行った。MISRA からはプロジェクトマネージャ、C や C++ ワーキンググループの議長など 6 名が参加した。ESCR C 言語版の改訂に関する MISRA の関心は高く、MISRA 側より英語版の提供について要請があった。

3）MISRA と IPA/SEC の間での双方向のレビュー活動における連携強化

2013 年 3 月、MISRA は MISRA C 改訂版（MISRA C:2012）を発行した。MISRA C:2012 では、これまで MISRA C:2004 でサポートしていた C 言語「C90」へのサポートは維持しつつ、「C99」へのサポートを拡張した。この MISRA C:2012 の策定プロセスでは、MISRA からの要請を受けて IPA/SEC の ESCR チームが改訂版ドラフトに対するレビューを実施した。この点について、IPA/SEC による MISRA C の改訂レビューコメントは、不明確な点の改善や分かりやすい記述とするために有用であったとの評価を MISRA から得ている。また、MISRA C:2012 では、IPA/SEC が策定した ESCR の中に記載されている事例も新たに引用されている。

今後、MISRA メンバーとの議論を IPA/SEC におけるガイドライン策定においても反映していくことは有意義であることから、MISRA と IPA/SEC のこうした連携関係を更に深め、レビューの段階で双方の専門的知見を活かして行くことで合意した。

4）C++ 版ガイドラインの改訂に向けた協力など

MISRA 及び IPA/SEC ESCR の C++ 版の改訂に向けた検討状況に関しては、双方とも現在、WG 立ち上げの議論を始めている段階で、同様のフェーズにあることから、今後、双方の WG での議論の進捗を踏まえ、適当な時点で進捗や取り組みの方向性について意見交換を行うこととした。

更に、MISRA の活動について、モデルベース開発向けのガイドラインやルール逸脱の許容に対するガイドラインについて意見交換を行った。このうち、ルール逸脱の許容に対するガイドラインは、「Approved deviation compliance」と呼ばれるものであり、ルール逸脱をどのような場合に許容するのかという基本的な考え方を示すものである。2013 年 2 月には、MISRA C:2004 に対するルール逸脱の許容の考え方を初めて示した MISRA C ADC（Approved deviation compliance for MISRA C:2004）が MISRA から公開された。

宇宙システムにおける上流工程仕様の妥当性確認技術

独立行政法人　宇宙航空研究開発機構　情報・計算工学センター
安全・信頼性推進部 上席開発員 技術領域総括

片平　真史

1 概要

宇宙システムに代表されるセーフティクリティカルシステムでは、要求された機能が諸条件の元で動作するという信頼性の向上のみではなく、安全性の視点でもその健全性の検証が求められる。特に、ソフトウェア開発において、仕様どおり動作することの保証は、従来の品質工学、信頼性工学に基づく手法により強化が可能であるが、要求仕様等の上流工程における仕様定義時の正確性の欠如、不完全性の残留、想定不足がシステムの安全性に脅威をもたらすことが多い。

宇宙分野でも、NASA 等の過去の事故事例からの教訓として、上流工程の仕様に関する問題に起因した事象が注目されている。JAXA（独立行政法人 宇宙航空研究開発機構）では、この解決のために、ソフトウェア IV&V（独立検証および妥当性確認）活動として、「モデル検査」や過去経験知に基づく「チェックリストベースレビュー」を導入している。

宇宙業界標準の開発プロセスとして、JAXA では、ソフトウェア開発標準（JERG-0-049）を定めており、その中で、妥当性確認を表 1 のとおり定義している。

図 1 に示すとおり、検証はあるリファレンスに基づいて評価されるものであるが、妥当性確認は、顧客や想定環境から抽出された上流要求に対し、開発全工程を通じてその成立性を確認するものである。ここでは、妥当性確認の視点で導入してきた「モデル検査」および「チェックリストベースレビュー」の取組みをその目的や期待効果とともに解説する。

表 1　検証および妥当性確認 (JERG-0-049 から引用)

用語	説明
検証	客観的証拠を提示することによって、規定要求事項が満たされていることを確認すること。
妥当性確認	客観的証拠を提示することによって、特定の意図された用途または適用に関する要求事項が満たされていることを確認すること。

2 取組みの目的

ソフトウェア開発の上流工程における要求仕様に関する問題に起因して、宇宙分野では以下の事故が発生している。

- NASA（アメリカ航空宇宙局）が 1998 年 12 月に打上げた火星探査機 MCO（マーズ･クライメイト･オービター）は、1999 年 9 月 23 日に火星軌道進入の際に、通信が途絶えた。事故調査の結果、予定されていた 140-150 km よりも低い 57 km の軌道で侵入してしまったことが主原因とされたが、これは、軌道モデルで使用されるソフトウェアの単位系の取り違え（メートル法のはずが誤ってヤードポンド法となっていた）により発生したとされている。**＜正確性の欠如＞**
- NASA が 1999 年 1 月に打上げた火星探査機 MPL（マーズ・ポーラー・ランダー）は、1999 年 12 月 3 日に、火星大気圏へ突入する際に、探査機からの信号が途絶え、行方不明となった。信号途絶の理由は現在でも不明であるが、最も可能性の高い原因として、着陸用の脚の展開時の衝撃を着陸と誤って判断したソフトウェアの問題とされている。また、この根本原因として、センサの振舞いに関する情報がソフトウェア要求仕様に欠落したと言われている。**＜不完全性の残留＞**
- ESA（欧州宇宙機関）が 1996 年 6 月 4 日に打上げたアリアン 5（Ariane 5）初号機は、打上げ後 37 秒後に予定していた飛行経路を大きく逸脱し爆発した。

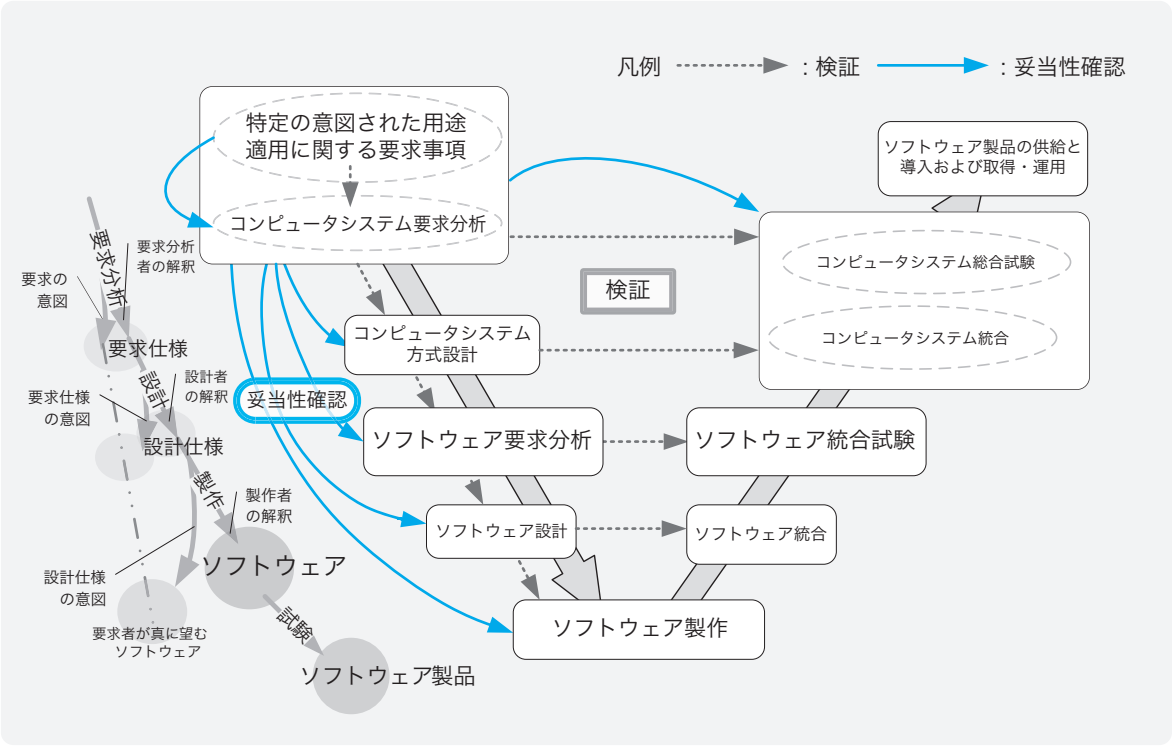


図１　検証と妥当性確認の概念図　(JERG-0-049 から引用)

事故調査の結果、一つの主要要因はアリアン４から再利用されている実行されないはずのソフトウェアが動作したためである。破壊に至った一つの副要因として、想定外のセンサの出力に対する処理が欠落していたことが挙げられている。**＜想定不足＞**

本事例で紹介する取組みは、これらの過去の事故・重大不具合の経験に基づき、ソフトウェア開発の上流工程における要求仕様等の仕様定義時に、正確性の欠如、不完全性の残留、使用方法・環境の想定不足を極力排除し、セーフティクリティカルシステムの安全性を向上させ、事故を未然回避することを目的としている。

3 取組みの対象、適用技術・手法、評価・計測

3.1. 取組みの対象製品と工程

ここでの取組みは、宇宙システムの中でも、変更が困難であり、また、短時間でセーフティクリティカルな判定が必要となる「人工衛星システム」や「ロケットシステム」に用いられる搭載ソフトウェアを対象として紹介する。

また、ここで紹介する取組みは、上流工程の仕様定義時の妥当性確認に限定しているが、実際の開発作業においてはそれ以外の工程にも適用されている。

3.2. 適用技術・手法

ソフトウェア開発の上流工程における要求仕様等の正確性の欠如、不完全性の残留、使用方法・環境の想定不足を排除するために、適用した技術・手法である「モデル検査」および「チェックリストベースレビュー」をここでは紹介する。この「モデル検査」および「チェックリストベースレビュー」は、技術・手法として、それぞれのデメリットを補完しながら使用している。通常 IV&V では、目的（観点）に対して、図２に示すように

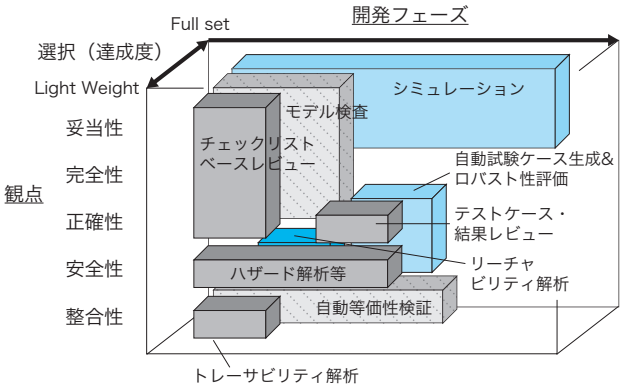


図２　IV&V 手法の組合せ・選択

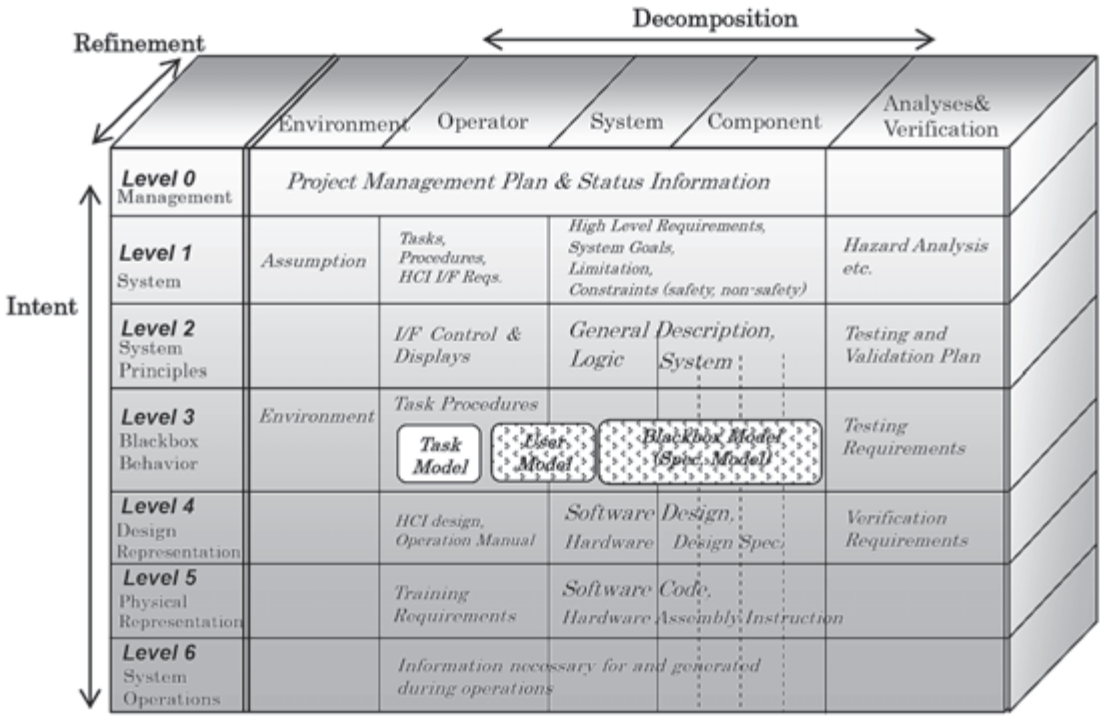


図３　Intent　Specification の構造

複数の手法を組合せ・選択し使用している。それぞれの技術・手法のメリット、デメリットを意識した適用が課題抽出を最大化するためには重要なポイントとなる。

モデル検査技術は、適切なモデル作成および検査目的を設定することにより、モデルやツールにより、複雑な事象を検査し、思いつきにくい想定シナリオを抽出、検証できる。

しかしながら、モデル検査技術のデメリットとして、比較的小規模な宇宙分野の搭載ソフトウェアであっても、ソフトウェア全体に対し、詳細な動作まで機能をモデル化し検証することは困難である。この解決のためには、モデル検査では困難な大局的な確認を行う必要がある。過去の事故・不具合や開発時の留意事項に基づいた経験則から作成されたチェックリストを用いて、人がレビューを行い、検査を比較的網羅的に実施している。このチェックリストベースレビューのデメリットは、検査対象の挙動が人により可読で、検討できる程度の単純な内容であることが点検内容の限界となることである。

モデル検査技術としてここで紹介するものは、適用実績のある手法のうち、以下の検査技術を代表事例として導入した時系列順に紹介する。

（1）SpecTRM（Specification Tools and Requirements Methodology）

（2）SPIN

また、チェックリストベースレビュー手法として、チェックリストの概要とレビュー方法について紹介する。

4 取組みの実際、および実施上の問題、対策・工夫

4.1. モデル検査技術

モデル検査技術は、システムのある性質に対しての「振る舞い」「状態遷移」等をモデル化することにより、ツール上で仕様内容をモデル検証することができる。すなわち、適切なモデル作成および検査目的を設定することにより、複雑な事象を検査し、思いつきにくい想定シナリオを抽出できる。

4.1.1. SpecTRM (Specification Tools and Requirements Methodology)

SpecTRM は、米 MIT（マサチューセッツ工科大学）Nancy G. Leveson が開発した方法論である。また、Safeware Engineering 社によりシステム／ソフトウェア安全設計支援ツールとして市販されている。SepcTRM は、形式的仕様記述 SpecTRM-RL（Requirement Language）により構成され、図３に示す Intent Specification で体系化された安全設計をガイドする統合化手法である。

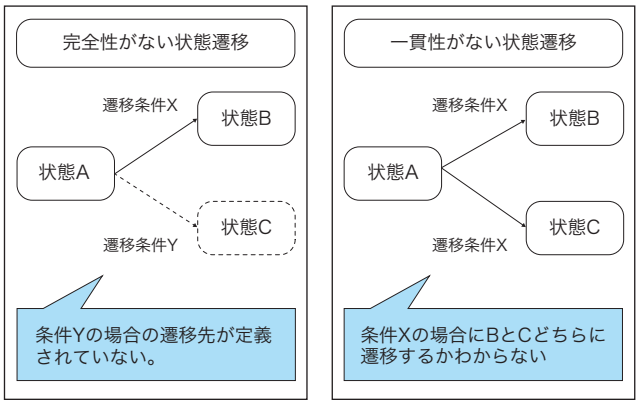


図４　SpecTRM のモデル解析機能

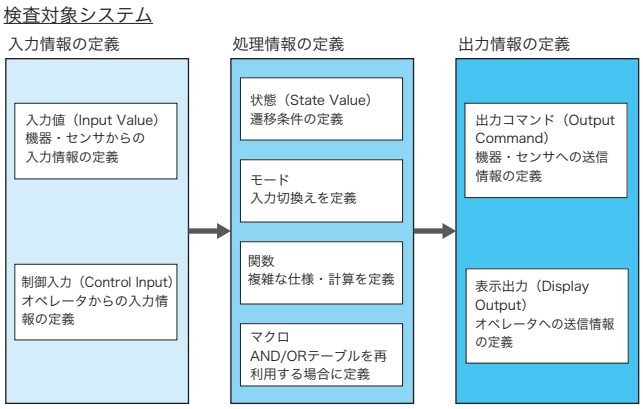


図５　SpecTRM-RL の６要素の関係

これらの手法は、過去の不具合事例の解析結果から必要となる安全上の考慮事項を網羅できるように構築されている。SpecTRM を利用することにより、Intent Specification に沿った対象システム／ソフトウェアのモデリング、モデル検査、シミュレーションを行なうことが可能となり、ガイドに従って安全設計を進めることができる。

モデリングおよびモデル検査を行うことにより、通常の自然言語による要求仕様のレビューでは判断することが難しい、仕様の一貫性（状態遷移先が複数発生することはないか）や、完全性（予想されない状態に陥ることはないか）の問題を抽出することができる。このモデリングおよびモデル検査は図２の Intent Specification では Level3 活動と対応付けられる。SpecTRM の特徴として、検査対象として選定した限定的な範囲で、システムが問題のある状態へ陥らないことを網羅的に自動検査し、上流工程からシステムの完全性・一貫性をモデル検査で検証しながら設計を進めることができる。

- 要求段階の抽象的な仕様から検査可能で、上流工程で問題を見つけることが可能
- ツールにより自動的に網羅的な検証が可能（他の方法で必要な検査式の設定がいらない）
- SpecTRM ツールは、Intent Specification Level3 にモデル検査として、以下の２つの自動分析用の「モデル解析機能」を持っている。（図４参照）
- 完全性分析 Completeness Analysis：状態遷移に関する遷移条件の抜けがないこと
- 一貫性分析 Deterministic Analysis：遷移先が一意でない遷移条件がないこと

SpecTRM モデルは、モデル記述言語 SpecTRM-RL で作成することになる。モデル作成は、システム開発におけるプログラミングに相応する作業と言え、モデル記述言語の文法等を学習する必要がある。検査モデルには、まずシステムの振る舞い（状態遷移仕様）を定義する必要があり、また状態遷移を表現するために、環境変化の表現、入力情報・出力情報等を文法に沿って定義する。

―SpecTRM-RL の６つのモデル要素―

（1）　入力（Inputs；機器・センサからの入力とオペレータ（主にディスプレイ）からの入力を区別して定義可能）
（2）　出力（Outputs；機器・センサへの出力とオペレータ（主にディスプレイ）への出力を区別して定義可能）
（3）　状態（States）
（4）　モード（Modes）
（5）　関数（Functions）
（6）　マクロ（Macros）

この６種類の各モデル要素の関係を図５に示す。対象システムの入出力情報や処理情報を識別し、各モデル要素でシステム仕様を表現し SpecTRM に登録していく。ただしモデル要素は全てを使う必要はなく、検証内容に応じてモデルを取捨選択することが可能である。

導入効果：

必要なシステムや人の動作をモデル化することにより、その完全性や一貫性を自動的に検査でき問題点を抽出、検証できる。完全性解析結果は、遷移先定義が無いことを意味するが、定義が不要なパターンも出力される可能性があるため、出力内容を分析し、各パターンの意味付けを行う必要がある。一方、一貫性解析結果は、遷移条件が一意に決まらない状態遷移があることを意味

するが、遷移不要なパターンも出力される可能性があるため、出力内容を分析し、各パターンの意味付けを行う必要がある。この結果、意味のあるとされたものは、「想定外・障害発生時のケース」に相当する。すなわち、この状態になった場合にシステムがどうすべきか決まっていないことを示しており、安全上の問題をもつものであるため、対応策を検討する必要がある。ここでは紹介しなかったが、安全性、一貫性の自動検査機能の他、入力値を与えて挙動の確認を行うシミュレーション機能もある。

SpecTRM を用いた検査では、システムの想定外・障害発生時のケース抜けを通常 1 システム辺り数十点の問題を抽出できる。また、対象システムの挙動と利用者の運用手順との不成立箇所の抽出にも利用され、手順・マニュアル開発に反映できる。

特徴：

- 様々な仕様を分析する際、モデルは比較的簡単にコピーができるため、コピーにより仕様を変えて自由度の高い分析を実施することができる
- 状態遷移や遷移条件等の情報が統一フォーマットで集約されるので、仕様の確認をする場合に設計書を見なくても必要な情報が得られやすい
- ツールがモデルの管理機能およびインポート・エクスポートを持っているため、過去に使用したモデルを蓄積し再利用する仕組みを用意すると、検査作業の効率化が図れる

課題：

以下のケースには向いていない。

- 複数の状態遷移が時間の経過とともに連携しながら発生するケース

（SpecTRM は状態遷移変化を伝搬し探索することが困難なため）

- 状態遷移の実行条件がシンプルなケース

（モデリング工数がかかる割には、複雑な条件に対する検査結果とならない）

4.1.2. SPIN

SPIN は、現 JPL Gerard J.Holzmann らが開発したモデル検査ツールである。この手法は、検査モデル作成に必要な情報（状態、実行条件、実行内容、外部イベント、制約条件等）が明確になっている工程が対象となる。特に、対象ソフトウェアが仕様どおりに稼働しない状態の検出、対象ソフトウェアの実行条件・状態遷移等に矛盾および抜けがないことの検証に向いている。SpecTRM に比較して、より仕様が具体的に定義された段階に導入しやすい。

この SPIN による検査には以下の入力情報を用いて作業を行う。すなわち、これらの情報が明確に定義できる段階から適用することが適切である。

■ システム機能仕様情報

- システム機能の処理内容
- システム機能の起動条件
- システム機能の処理フロー等

■ 検証条件情報

- システム異常状態の定義情報（エラー出力する状態）
- システム正常終了の情報等（正常終了でない状態を異常状態として使用する）

検査結果として、以下の結果を得ることができる。

■ 検証条件の成立可否

■ 反例情報（初期状態から検証条件を満たす状態までの経緯）

■ 状態遷移の環境情報（検証条件を満たした状態の、状態管理変数の値等）

検査結果として得られた反例情報を一つずつ実際に問題であるかどうかを評価しながら判定していくことになる。

導入効果：

比較的仕様が明確になってきた場合に導入すると、SpecTRM では検査が困難な、複数の状態遷移が時間の経過とともに連携しながら発生する問題を抽出することができる。

宇宙システムに適用を行い、設計の初期段階の仕様に対して危険な状態になるシナリオ（反例情報）を識別することができた。このシナリオは、安全上問題に繋がる可能性の高いものであり、対策検討につながった。

課題：

効果を上げるためには、反例情報の分析および検証条件の設定をある程度定型化しながら行う等の工夫が必要である。

4.1.3. モデル検査技術の導入上の課題と解決策

モデル検査技術の JAXA における導入事例から以下の経験則がある。

表2　ボイジャー・ガリレオチェックリスト

Interfaces
(1) Is the software's response to out-of-range values specified for every input?
(2) Is the software's response to not receiving an expected input specified? (That is, are timeouts provided?) Does the software specify the length of the timeout, when to start counting the timeout, and the latency of the timeout (the point past which the receipt of new inputs cannot change the output result, even if they arrive before the actual output)?
(3) If input arrives when it shouldn't, is a response specified?
(4) On a given input, will the software always follow the same path through the code (that is, is the software's behavior deterministic)?
(5) Is each input bounded in time? That is, does the specification include the earliest time at which the input will be accepted and the latest time at which the data will be considered valid (to avoid making control decisions based on obsolete data)?
(6) Is a minimum and maximum arrival rate specified for each input (for example, a capacity limit on interrupts signaling an input)? For each communication path? Are checks performed in the software to avoid signal saturation?
(7) If interrupts are masked or disabled, can events be lost?
(8) Can any output be produced faster than it can be used (absorbed) by the interfacing module? Is overloaded behavior specified?
(9) Is all data output to the buses form the sensors used by the software? If not, it is likely that some required function has been omitted from the specification.
(10) Can input that is received before startup, while offline, or after shutdown influence the software's startup behavior? For example, are the values of any counters, timers, or signals retained in software or hardware during shutdown? If so, is the earliest or most-recent value retained?

Robustness
(11) In cases where performance degradation is the chosen error response, is the degradation predictable (for example, lower accuracy, longer response time)?
(12) Are there sufficient delays incorporated into the error-recovery responses, e.g., to avoid returning to the normal state too quickly?
(13) Are feedback loops (including echoes) specified, where appropriate, to compare the actual effects of outputs on the system with the predicted effects?
(14) Are all modes and modules of the specified software reachable (used in some path through the code)? If not, the specification may include superfluous items.
(15) If a hazards analysis has been done, does every path from a hazardous state (a failure-mode) lead to a low-risk state?
(16) Are the inputs identified which, if not received (for example, due to sensor failure), can lead to a hazardous state or can prevent recovery (single-point failures)?

・強力なモデル検査の他、モデル検査の前のモデル作成段階で多くの仕様上の問題を抽出できる

また、モデル検査技術の導入上の課題と JAXA における解決策は以下のとおりである。

・モデル検査に依存し過ぎると大観的な重要な問題を見逃すことがあるが、チェックリストベースレビューなど他の手法との組合せによってこれを防ぐことができる。
・システムおよび環境をすべてモデル化することが困難なため、誤った絞り込みや抽象化を行うことにより、重大な問題を見逃すことがある。作業実施前に確認したいシナリオを分析し、関係する範囲を事前に仕様設定者と確認を行うことで回避できる。
・モデル化の段階および検査結果から正しく問題点を抽出するためには、対象システムのドメイン知識がないと誤った問題の抽出となるため、ドメイン知識を習得してから、またはドメイン知識を有する者と作業を行うことが望ましい。
・モデル検査はモデル化した範囲である程度網羅的に特徴抽出を行うことができる反面、連続系の制御則の様なシステムを検査することが困難である。特定の検査したい対象シナリオが明確になっている場合は、モデルを動作（シミュレーション）させてランダム入力に対する検証を行うことが有効である。

4.2. チェックリストベースレビュー手法

4.2.1. ボイジャー・ガリレオチェックリスト

JAXA が現在利用しているチェックリストの一つとして、1996 年に JPL（NASA ジェット推進研究所）Lutz が発表した Safety Checklist※1 がある。通称、ボイジャー・ガリレオチェックリスト（表 2）である。

Lutz は、1991 年に Jaffe, Leveson らが過去の事故データの分析結果に基づき提唱したセーフティクリティカルのクライテリアを参考にチェックリストを作成した。また同時に NASA の探査機ボイジャー（Voyager）およ

【脚注】
※ 1　Robyn R. Lutz "Targeting Safety-Related Errors During Software Requirements Analysis," The Journal of Systems and Software, Vol. 34, Sept, 1996, pp. 223-230.

びガリレオ（Galileo）で実際に発生した 192 件の不具合を詳細に分析し、チェックリストを検証した。このチェックリストを用いることで、192 件の不具合のうち 77% である 149 件は、要求分析段階で発見できたことが分かった。

4.2.2. JAXA 独自チェックリスト

4.2.1 項で紹介したものは、JPL の不具合分析の結果を活用したボイジャー・ガリレオチェックリストの導入例であるが、日本の宇宙業界でも同種の不具合が発生しており、これらの不具合情報や開発・検証の経験者の知見を体系化した JAXA 独自のチェックリストを整備、利用している。表３に姿勢制御系チェックリストの例を示す。

4.2.3. チェックリストレビューの導入効果

ボイジャー・ガリレオチェックリストや姿勢制御系チェックリスト等を利用してレビューを行うことにより、通常のレビューよりも過去の不具合事例・経験知に基づく確認ポイントを確実にチェックでき、問題ない場合も含み点検結果の記録を残すことができる。ボイジャー・ガリレオチェックリストでは、宇宙システムの適用を通じて、要求仕様上の問題点を通常 1 システムにつき数点の問題点が要求定義段階で抽出できる。タイムアウト時間の設定がなく、待ち続けてしまうケースや、信頼性の観点でバックアップ計算機に切替える場合にシステムの状態によっては、切替信号を送信されずにシステムが停止するケースが発見され、重大な事故の原因を要求段階で未然に抽出・解決できることが実証できた。また、姿勢制御系チェックリスト等の場合、単位系等の初歩的な間違えから想定外の故障時の処理機能の抜け等複雑な問題まで幅広いレベルの要求仕様上の問題点を抽出でき通常 1 システムにつき数十点の問題点が要求定義段階で抽出できる。２章で示した MCO の例の単位系の誤りもこのチェックリストを利用してチェックしていれば要求定義段階で発見できたはずである。JAXA では同様なチェックリストを５つのシステム分類に対し、準備しており、重大な事故の原因を要求段階で未然に抽出・解決できることが実証できた。

4.2.4. チェックリストベースレビューの導入上の課題と解決策

チェックリストを用いたレビュー活動は、過去の不具合事例・経験知に基づく確認活動が可能となるが、課題としては、問題点を正しく抽出できるか否かが、レビュー活動を行う作業担当者のスキルに依存してしまうことである。また、限られた時間の中ですべての確認項目を同様の深さで確認することは期待する結果が得られないことになる。

このチェックリストを用いたレビューの質・効率を作業担当者のスキルに極力依存しないように、JAXA では、それぞれのチェック項目に、詳細なチェック方法を補足説明している。以下に人工衛星姿勢制御系サブシステムのチェックリストをサンプルに解説する（図６）。

また、複数の作業担当者で共同して実施する場合は、

表３　姿勢制御系チェックリスト　（抜粋）

項目	内容	詳細	事例	対象文書
入出力データ	物理量データ毎に単位が明確に記述されているか？	搭載ソフトウェアの中では、角度は、ラジアン単位、度単位が混在して用いられているので、単位の扱いについては十分な注意が必要である。	設計基準に従って入出力データインタフェースは統一されているが、既開発品の利用により、機器毎に異なる場合がある。	データ仕様
ON/OFF 手順	ON/OFF 手順、および ON 時データ処理手順は明記されているか？	使用するセンサによっては、電源 ON 後、センサ信号を用いるまでにマスキング等の初期設定操作を必要とするものがある。	マスキングをしていない場合、誤った動作をしたことがあった。	要求仕様書
異常信号処理	異常信号（雑音）に対して処置策が施されているか？	パルス信号に雑音が重畳し、その結果、物理的に可能な回転速度範囲を逸脱していないかをソフトウェア側で絶えずチェックする必要があり、要求仕様に規定されているか？		要求仕様書
フォールトトレラント設計	モード別の FDIR の設計が妥当か？	故障耐性の設計として、モードごとの故障解析（FMEA、FTA）の有無、クリティカル故障の抽出、それに対する設計上の対応策（FDIR 機能）がとられているか？冗長構成をとるのが難しい、あるいは不可能な部分を除いて単一故障点が回避された設計となっているか？	故障を特定できるか？センサを使わないモードで FDIR が機能していないか？	要求仕様書

チェック項目、詳細確認ポイントをスキルに合わせて配分し、完了後に作業担当者間で確認を行うことで作業の質・効率を向上できる。

優先度：　高、中、低

　重要な機能に関わるチェック項目、あるいは問題が多く見られる項目の優先度

　難易度：

　易……ひとつの項目のチェックを、一人の担当者が15 分以内で行える

　中……30 分以内

　難…… 1 時間以上

　姿勢制御系チェックリストのチェック項目の構成と所要時間を表 4 に示す。

　実際の利用時には、対象となる宇宙システムの特徴・リスクおよび配分できる作業予定時間から、それぞれの装置に対して、優先度のレベルを選択し、点検項目を決定する。あらかじめ定義された難易度に合わせて、ボイジャー・ガリレオチェックリストと同様に作業担当者のスキルと照らしながら、作業分担を行っている。

5 達成度の評価、取組みの結果

　今回、適用した技術・手法である「モデル検査」および「チェックリストベースレビュー」の個別の導入の効果は、 4 章で述べたとおりである。ここでは、当初目的として設定した上流工程における仕様定義段階の 3 つの課題の改善状況について表 5 にまとめる。

例えば、チェック項目（2）

Is the software's response to not receiving an expected input specified? (That is, are timeouts provided?) Does the software specify the length of the timeout, when to start counting the timeout, and the latency of the timeout (the point past which the receipt of new inputs cannot change the output result, even if they arrive before the actual output)?

詳細確認ポイント：

- 安全上重要なソフトウェアに対し期待される入力値にタイムアウト時間が設定されているか？
- タイムアウト時間が設定されていない場合、対象のソフトウェアの挙動が安全であり、期待どおりに動作するか？
- 必要なハードウエア、インタフェースは存在するか？また、他の事象によりインタフェースが利用できないことはないか？
- タイムアウト時間はインプット、アウトプット側で定義・了解されているか？

・・・など。

図 6　人工衛星姿勢制御系サブシステムの補足説明例

表 4　チェック項目の内訳

優先度	高			中			低			計
難易度	易	中	難	易	中	難	易	中	難	
センサ A	6		1	5			1			13
センサ B	3	1	1	4			1			10
センサ C				4	2					6
センサ D	2						4		1	7
センサ E				1	1	1				3
アクチュエータ A	6	1					1	1	1	10
アクチュエータ B	3			4			3	1		11
全体システム	1	1	2			1	4	2	3	14
計	21	3	4	18	3	2	14	4	5	74
作業時間 (h)	5.25	1.5	4.0	4.5	1.5	2.0	3.5	2.0	5.0	29.25

表 5　上流工程における仕様定義の改善状況

上流工程の仕様の課題	改善状況
正確性の欠如	仕様中の単位誤記・正負／極性反転等の初歩的であるが重大な事故を引き起こす過誤等、記述の曖昧さに起因する複雑な挙動のチェックリストベースレビューにて多くの問題を早期に抽出できる。また、仕様書間の一貫性をモデル検査技術により確認することで、人がチェックしにくい複雑な利用シナリオ上の過誤を検出可能となる。
不完全性の残留	仕様書の記載が不足・欠落したことによるアルゴリズムや機能の欠落・欠陥については、完全性をモデル検査技術により抽出可能となる。しかしながら、本来のシステムに期待される正しい動きを理解・把握する人とともに、意図する挙動かどうか最終確認することが、より精度を高めることにつながる。また、仕様記述の際に欠落しやすい情報についてはチェックリストなどに含め、レビューにて点検することが更に効果を上げることになる。
使用方法・環境の想定不足	システムが動作する環境が全て網羅され仕様が設定されていることは少ないのが一般的であるが、特に安全確保のためには、使用方法・環境の想定を超えた場合でも安全な動作が期待される。環境や使用方法も含めてモデル検査を行うことで、想定しない危険シナリオ等をある値度抽出することができるが、今後の改良が必要である。

実践的情報教育協働ネットワーク：enPiT



大阪大学大学院情報科学研究科
春名　修介　　楠本　真二　　井上　克郎

情報技術を駆使し、社会が抱える問題を解決できる人材の育成のため、全国 15 大学が連携した「分野・地域を超えた実践的情報教育協働ネットワーク（通称：enPiT）」が平成 24 年度より始まっている。産業界・外部団体との密接な連携のもと、実践的な教育プログラムが展開されており、平成 25 年度は、修了者 309 名、参加教員 241 名、参加大学 47 校、連携企業数 87 社と順調な滑り出しを見せている。

1 enPiT とは

情報技術を駆使し社会が抱える問題を解決できる実践力を持つ人材を育成するため、文部科学省により「情報技術人材育成のための実践教育ネットワーク形成事業」が、平成 24 年度から開始された。本事業は、複数の大学と産業界が協力して全国的なネットワークを形成し、実際の課題に基づく課題解決型学習等の実践的な教育を実施・普及させることを目的とした公募型事業である。公募の結果、大阪大学を代表校とする「分野・地域を越えた実践的情報教育協働ネットワーク（Education Network for Practical Information Technologies：enPiT）」が採択され、活動を開始した。

1.1. 教育分野

enPiT では、図 1 に示すように、全国 15 の大学（連携大学）が中心となり、図 2 に示す運営体制のもと、enPiT に参加する大学（参加大学）や連携企業と密に連携し、人材が必要とされている以下の 4 つの分野において、実践的な情報教育を展開している。

○ クラウドコンピューティング分野
- 大阪大学、東京大学、東京工業大学、神戸大学、九州工業大学

○ セキュリティ分野
- 東北大学、北陸先端科学技術大学院大学、奈良先端科学技術大学院大学、慶應義塾大学、情報セキュリティ大学院大学

○ 組込みシステム分野
- 九州大学、名古屋大学

○ ビジネスアプリケーション分野
- 筑波大学、産業技術大学院大学、公立はこだて未来大学

enPiT では、修士課程の学生を対象とし、平成 25 年度から 28 年度の 4 年間で、合計 1200 名程度の学生を修了させる予定である。

1.2. 教育プログラムのフレームワーク

enPiT では、各大学が以下に示すような共通のフレームワークに基づき、教育プログラムを設計し実施している。主な教育対象は修士 1 年生であるが、修士 2 年や社会人に対しても教育を行っているプログラムもある。

・基礎知識学習 実施時期：４月～６月

各大学の講義や遠隔講義等の教育コンテンツを利用して、その分野で必要とされる基礎知識の学習を行う。

・短期集中合宿 実施時期：７月～９月

複数の大学の学生が一同に会して課題解決型学習

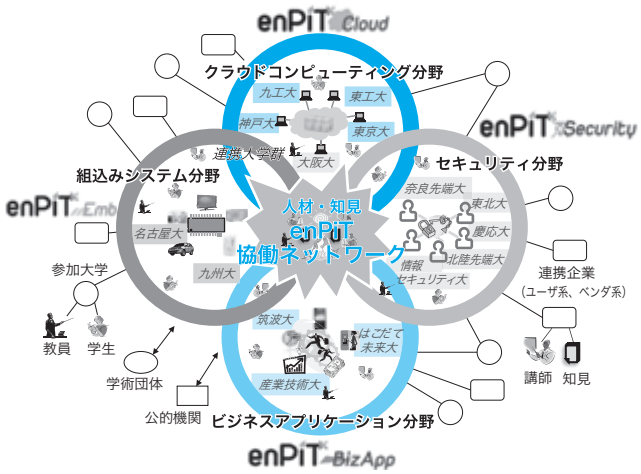


図１　enPiT の４つの教育分野と 15 連携大学

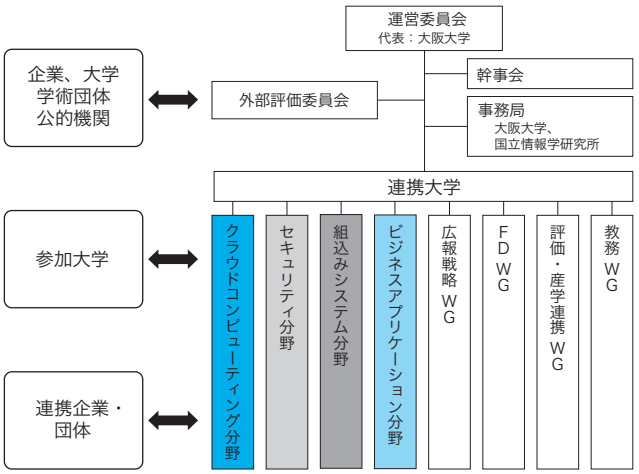


図２　enPiT 運営体制

(PBL:Project Based Learning) を短期間集中的に行い（5 日間× 2 回程度)、交流を深め、分野の具体的な知識を修得する。

・分散 PBL 実施時期：10 月～３月

各学生は自大学の環境において、種々の連絡手段を利用しながら、他大学の学生と協力し PBL を実施する。最後に成果発表会を行い、プログラムの修了となる。

2 活動状況

本章では、連携大学の活動を中心に enPiT の活動内容について説明する。

2.1. クラウドコンピューティング分野

クラウドコンピューティング分野（enPiT-Cloud）は、所謂ビッグデータの分析手法、新しいビジネス分野の創出といった社会の具体的な課題を、クラウド技術を活用し解決できる人材の育成を目標としている。５つの大学で、Cloud Spiral（大阪大学・神戸大学)、クラウド実践道場（東京大学)、Cloud Bauhaus（東京工業大学)、クラウド Q9（九州工業大学）の４つのプログラムを実施している。

< 基礎知識学習 >

クラウドコンピューティングを支える仮想化・分散処理・大規模データ処理などの要素技術、ソフトウェア工学、プロジェクトマネジメント、ファシリテーション、アジャイル開発手法といった実際のソフトウェア開発で必要となる技術の習得を行う。また、クラウドの利活用による新サービス創出の観点から、ロジカルシンキング、クリエイティブシンキングなどの共通的な講義・演習も

図３　西日本での短期集中合宿の様子

重視し、合わせて実施している。

< 短期集中合宿 >

地理的な関係から東日本（クラウド実践道場と Cloud Bauhaus の合同）と西日本（Cloud Spiral とクラウド Q9 の合同）の２個所で実施している。図３は西日本で実施された短期集中合宿の様子である。分散 PBL の準備として、クラウドシステムの構築や具体的な Web アプリケーション開発を体験させると共に、クラウド利活用に関する最新動向に関するセミナーも開催している。

< 分散 PBL>

数名から構成されるグループ単位で、大規模 POS データ分析によるコンビニ販売戦略の立案、クラウドを用いたビジネスモデル提案、学生生活を豊かにするアプリケーション開発など、クラウドならではの特徴を持つテーマを自ら企画し、開発の進捗管理を行いながら完成させる実践的なソフトウェア開発を体験する。

2.2. セキュリティ分野

セキュリティ分野（enPiT-Security）は、社会・経済

活動の根幹にかかわる情報資産及び情報流通のセキュリティ対策を、技術面・管理面で牽引できる実践セキュリティ人材の育成を目標としている。５つの大学（情報セキュリティ大学院大学、東北大学、北陸先端科学技術大学院大学、奈良先端科学技術大学院大学、慶應義塾大学）が連携して、実践セキュリティ人材の育成コース（SecCap）を実施している。

< 基礎知識学習 >

情報セキュリティ運用リテラシーに関する共通科目と連携大学が提供する基礎科目から構成されている。暗号、情報理論、情報ネットワーク、社会システム、システム監査、セキュリティマネジメントなど幅広い科目で構成されている。

< 短期集中合宿 >

ハードウェアを対象としたもの、システムやソフトウェアを対象としたもの、企業組織のリスク管理を対象としたものなど、技術系・理論系の演習に留まらず社会系の演習も選択することができる。仮想環境を準備し、実際にセキュリティ攻撃を受けたときの一連のプロセス（分析・対策・被害レポートのまとめ・報告）を体験するデジタルフォレンジック演習のように、実際の企業の中で行われている状況を想定した演習が用意されている（図４）。

図４　ネットワークセキュリティ検査演習の様子

← 掃除機型ロボットの走行中の地図作成

↓ ロボットチャレンジ競技フィールド

図５　ESS ロボットチャレンジの様子

< 分散 PBL>

上記の実践演習を更に強化するため、実社会で活躍する講師陣による最新のセキュリティ事情や実用的な技術に重点を置いた講義・演習を提供している。また、実践演習の一環として、平成 25 年 10 月に開催されたセキュリティ技術を競うコンテストである MWS Cup 2013 の CTF イベントに学生チームとして参加し、優秀な成績を収めている。

2.3. 組込みシステム分野

組込みシステム分野（enPiT-Emb）は、組込みシステム開発技術を活用して産業界の具体的な課題を解決し、付加価値の高いサイバーフィジカルシステム（CPS）を構築できる人材の育成を目標としている。九州大学が実施する連合型 PBL と名古屋大学が実施する発展型 OJL（On the Job Learning）の２つのプログラムを実施している。両プログラムとも、問題発見・解決能力を身につける「基本コース」と、管理技術とその運用方法まで踏込んだ高度な問題解決能力を身につける「発展コース」から構成されている。

< 基礎知識学習 >

機械系・電気電子系・ソフトウェア系と多岐にわたる技術基盤を持つ組込みシステムの特性を理解するための組込みシステム基礎、開発の基本となるソフトウェア工学やプロジェクトマネジメント、先進的な取り組みであるモデル駆動開発などを広く学習する。

< 短期集中合宿 >

開発プロセス、プロジェクト計画、プロジェクトの予実管理、開発文書の書き方、チームビルディング、コミュニケーションといった分散 PBL に備えるための演習を集中的に実施する。また、学会活動とも連携し、この期間内に連合型 PBL では掃除機型ロボットの走行を題材にした ESS（Embedded Systems Symposium）ロボットチャレンジへ（図５）、発展型 OJL では、SWEST（Summer Workshop on Embedded System Technologies）にそれぞれ、参加する。

< 分散 PBL>

連合型 PBL では、ESS ロボットチャレンジを目指すグループと、自らテーマを決め開発を行うグループに分かれる。発展型 OJL では、産業界から発案された開発課題に対して、学生、教員、企業の管理者や担当者が開発チームを作り、専任のプロジェクト・マネージャの管理のもと、実業務さながらの開発を行う。

2.4. ビジネスアプリケーション分野

ビジネスアプリケーション分野（enPiT-BizApp）は、先端情報技術や情報インフラを有機的に活用し、ビジネスニーズや社会ニーズに対する実践的問題解決ができる人材の育成を目標としている。筑波大学、産業技術大学院大学、公立はこだて未来大学の３校を連携大学とし、先端情報技術の習得に加え、問題解決のためのソリューション提供を重視した教育を行っている。

< 基礎知識学習 >

ソフトウェア工学、サービス指向システム開発、アジャイル開発手法、e ビジネス、情報システム構築プロジェクトマネジメントなどビジネスアプリケーション開発の基本的な技術についての講義を行っている。

< 短期集中合宿 >

サービス提供企業との連携のもと、モバイルサービスソフトウェアやビジネスアプリケーションの短期集中演習を通じて、サーバ、クラウドといった情報インフラの利用技術を習得する。開発に際しては、アジャイル開発手法の一つである Scrum を取り入れている。また、受講者の幅広いニーズに対応するため、スタンダードコースと期間が短いライトウェイトコースを設けている大学もある。

< 分散 PBL>

社会の実問題を捉えるような課題設定を行うために、連携企業が顧客となり実際のシステム開発を受講者グループに発注する形で PBL が進められる。受講者は顧客を交えた要求獲得、顧客との折衝、システム開発、納入、保守など、実際のシステム開発を体験することができる。このように、学生が自主的に問題を発見し、取り組むことができる機会を提供し、創造的なソリューションを発案する能力や、潜在的な顧客に対してソリューションを提案する能力の強化に努めている。図６は、成果発表会の様子である。

図６　成果発表会の様子

2.5. 作業部会（WG）活動など

上記４分野の教育活動を側面で支援する活動について簡単に触れる。

・広報戦略 WG

ホームページの運営、広報物の配布、大学・教員・学生のニーズ及び認知度の調査などの広報戦略を担当する。

・教務 WG

分野横断講義の実施や教材の共有方法等の検討を担当する。

・評価・産学連携 WG

受講生が最先端の情報技術を実践的に活用することができる人材として育成されたか、産業界の意見が反映されているかなどの評価を担当する。

・FD WG

若手教員の交流や教育手法の改善など教員の FD（Faculty Development）のための施策を担当する。

・その他（発表会など）

年度末に開催される各大学の成果発表会や年１回開催される enPiT シンポジウム（本年年度は 2015/1/27 名古屋大学にて開催予定）などを通じて、enPiT の成果の公開に努めている。

3 現状と今後

平成 24 年度の準備期間を経て、本格活動を開始した平成 25 年度は、修了者数 309 名、参加教員数 241 名、参加大学数 47 校、連携企業数 87 と協働ネットワークとして大きな成果を収めることができた。また、受講者アンケートの結果も好評で、順調な滑り出しを見せている。

平成 26 年度より、非情報系学部・学科の学生に対しても門戸を広げ、複数領域の知識・スキルを併せ持つ人材（ハイブリッド人材）の育成も進めていく予定である。

今後、より多くの大学でこのような実践的な情報教育が普及し、多くの学生が参加できるよう、今までの実績を広く広報して行く所存である。また、新たな大学や企業、組織などとの連携を深め、enPiT ブランドの更なる普及・確立を目指すと共に、enPiT 修了生が日本の情報産業界のリーダとなって活躍することを期待している。

enPiT　http://www.enpit.jp/

Column

組込みソフトウェアの時代と日本の「ものづくり」

IPA 顧問
松田　晃一

トヨタ自動車は、今年２月にプリウスの大規模なリコールを発表しました。原因はハイブリッドシステムのモーター制御用ソフトウェアの設計ミスとのこと。国内外で 190 万台というかつてない規模のリコールとなったようです。なぜかこの同じ時期に、日本を代表する自動車メーカが相次いでリコールを発表しています。今年２月には、ホンダフィット、３月には日産アルティマ（米国中心）、４月にはマツダ アテンザ　続いて三菱自動車が軽自動車 eK をリコールといった具合です。そして、注目すべきはそのすべてが組込みソフトウェアの不具合によるリコールである点です。

土俵が変わった「ものづくり」

日本の製品はその高い品質によって強い競争力を保っています。その源泉は、日本の「ものづくり」の力であることは間違いありません。製造現場において高い意識と技能を持った従業員が、日々のカイゼン活動などを通して、高品質な製品を均質に量産する力を維持してきたことにあります。専用の部品群や専用の回路を複雑に組み合わせて作られる製品であればあるほど、それを高品質に量産できる「ものづくり」の力は強い競争力を発揮できるわけです。しかし、そのような複雑な製品も、汎用的な電子制御装置とその上で動く組込みソフトウェアにどんどん置き換えられていっています。このことは、日本が得意とする「ものづくり」の力を発揮する場が、どんどん狭くなっていることを意味します。なぜなら、日本の「ものづくり」力が大きな力を発揮するのは、多くの部品群を複雑に組み合わせ、擦り合わせて作るハードウェア製品を量産する部分だからです。そのような部分が少なくなってソフトウェアの比重が高くなればなるほど、日本のこれまでの強みが通用し難くなります。「ものづくり」の競争の場が移っているのです。今までの「ものづくり」の力に寄り掛かったままでは、日本製品の競争力は削がれる一方ではないでしょうか。

組込みソフトウェア時代の「人づくり」

では、このような組込みソフトウェアの時代を迎えた現在、新しい「ものづくり」はどう考えれば良いのでしょうか？

第１に「ものづくり」の競争の場がソフトウェアに移っていることを正しく認識し、組込みソフトウェアの重要性を理解し、そのための人材の育成に取組むことです。ハードウェアを中心とする製品については、「ものづくり」の重要性は十分に理解され、結果として強い日本の「ものづくり」力が実現されました。しかし、ソフトウェアは目に見えず実体が掴みにくいためか、なかなか理解が進まず、経営陣の関心も低いようです。ハードウェアを中心とした製品の時と同様に、人材育成や品質管理に十分な経営資源を投入し、組込みソフトウェアを中心とする製品の時代に適した新しい「ものづくり」の形に造り替えることが何より重要ではないでしょうか。

組込みソフトウェア時代の品質管理

第２に、品質の高い組込みソフトウェアの設計・製造力です。ハードウェア中心の製品の時代には、高い品質のものづくりを世界に誇っていました。それを維持するには、可能な限り欠陥の少ないソフトウェアの開発技術を追求することがまず必要です。しかしどれだけテストを重ねたとしても、欠陥を０にすることは不可能です。冒頭のリコールの例のように、残った欠陥が市場で顕在化する可能性はますます高くなることを覚悟せざるを得ません。それをどうやって最少の被害で抑えるかが重要な技術になるでしょう。組込み製品の場合は、エンタープライズシステムのような一品生産のものと違って、何万、何十万という製品が市場で実際に使われるわけですから、その運転状況をリアルタイムでモニタリングし集積したビッグデータを分析することによって、ソフトウェアの欠陥の予兆を見出し、修正する技術の可能性を追及したいものです。

組込みソフトウェア時代の製品企画

そして第３の点は、ユーザが「オッ」と驚くような新しい体験を提供できる価値・機能を生む製品企画力です。これまでの製品でもこれは非常に重要でしたが、組込みソフトウェアの場合は、ハードウェアに比べて実現できる機能の自由度が高く、新しいアイデアを製品に組み込むことが比較的容易になります。だからこそ、製品の価値を高める機能のアイデア、企画が一層重要で、勝負の分かれ目になると思います。

組込みソフトウェアの時代に相応しい、新しい「ものづくり」の文化を創り出し、これまでの日本の製品の強みに更に磨きをかけていきたいものです。

みんなでソフトウェアの垣根を乗り越えよう

ソフト・エッジ
ソフトウェア開発の科学を求めて

中島 震、みわ よしこ　著

ISBN: 978-4621053836
丸善出版刊
新書判・272 頁
定価 760 円（税抜）
2013 年 3 月刊

現代社会における IT の重要性は誰しもが認識していることだが、その IT のほとんどがソフトウェアによって制御されていることを認識している人はそれほど多くないのではないだろうか。銀行の ATM システムや電子行政システムなどの所謂 IT システムに限らず、自動車や家電機器などの製品も、その開発費の 60%以上をソフトウェアの開発費が占めるほどになっている。現代社会はソフトウェアが支えていると言っても過言ではない。

しかしながら、ハードウェアに比べてソフトウェアは無形で目に見えない。このため、その本質が何なのか極めて理解し難い。理解し難いということは制御しづらいということであり、それは社会生活に対して大きなリスクとなる。

本書では、人工衛星打ち上げロケットの爆発事故や銀行システム統合時の障害など、ソフトウェアが原因で実際に起こった事故を例として、ソフトウェアの制御がいかに難しいかについて述べている。更に、ソフトウェアの本質を社会科学的に考察することで、その複雑さが持つリスクを体系的に整理するとともに、こうしたソフトウェアをリスク少なく開発するための方法論について紹介している。

本書の題名のエッジという言葉には境界という意味があるが、多くの人々にソフトウェアが持つ本質的な複雑さを理解してもらい、その境界を乗り越えてもらいたいという筆者の意図が込められている。専門家以外の一般の読者にもわかり易く解説されており、エッジを乗り越える人々が増えていくことが期待できる。

（松本　隆明）

人間社会にとってのエンジニアリングとは？

エンジニアリング システムズ
複雑な技術社会において人間のニーズを満たす

オリヴィエ・L・デ・ヴェック、他　著
春山真一郎　監訳

ISBN: 978-4-7664-2110-1
慶應義塾大学出版会刊
A5 判・248 頁
定価 3,600 円 (税抜)
2014 年 2 月刊

この本をある方に薦められ、システムズエンジニアリングの新たな技術書かと想像しつつ購読したところ、全く違った内容であり、興味深く読ませて頂いた。

システムズエンジニアリングは、大規模・複雑なシステムのプロジェクトを様々な利害関係者等の要求も踏まえつつ成功に導くためのものであるが、本書「エンジニアリングシステムズ」では更に、システムの外側にある環境や法規制等を含む社会システムとの相互作用を考慮し、全体最適を目指すことの必要性を説いている。

書中では、過去から現在に至る様々なシステムやその産業、更には社会システムの変遷や相互作用による変化等に関して多くの具体例を交えて説明されており、読み進むにつれ、従来の、かつ自分自身のエンジニアリングとしての視野の狭さを感じさせるものであった。

例えば、電気自動車はガソリンを消費せず CO_2 の排出がないため環境に良い、と言っても、その電気はどこでどのようにしてつくられ、バッテリーはどのような材料で作られているのか等等、社会全体・地球全体として本当に環境に良いのか十分に議論されることは少ない。また、ガソリン車から電気自動車に代わることにより、ショッピングセンタの駐車上から都市交通全体まで、様々な影響が出てくると思われる。

エンジニアは、開発している製品・システム自体、更にはそれに関与する利害関係者や接続されるシステムの範囲だけでなく、それが及ぼす社会システムへの影響までも考慮した上で、より最適なものをデザインする必要がある事を強く感じさせる一書であった。なお、これらを説明する重要な要素として「イリティ”ility”」という用語が紹介されており、これからは頻繁に耳にするようになると思われる。本書の前半は抽象的な部分が多いが、後半にかけて、より具体的で分かりやすくなっている。

（中村　雄三）

編集後記

今号の SEC journal では、所長対談として長年ソフトウェア産業界で幅広くご活躍の芝浦工業大学院教授の國井秀子先生にご登場を願い、日本のソフトウェア産業の多岐にわたる課題に対する先生の業界人、教育者としてのそれぞれの側面からイノベーティブなお考えをお聞きすることができ、示唆に満ちた対談内容となっております。

また今号は成果特集号として、2013 年度の SEC 成果報告を特集しています。ことに、安全・安心な社会を目指す活動として、先ごろ記者発表を行いました重要インフラ障害から学んだ教訓集は、多くの企業の方々からのご協力を頂いた SEC の立場ならではの成果となっており、広くお役立ていただけるものになっています。

さて、去る 5 月 14 日から 16 日の 3 日間、東京ビッグサイトにて「第 11 回情報セキュリティ EXPO」が開催され、IPA 出展ブースでの一連のセキュリティ・セミナーでは 3 日間で約 10,000 人のご来場を賜りました。 これも最近立て続けに起こっている情報セキュリティへの脅威を、身近なこととしての関心の高まりがうかがわれます。同時に、デファクト・スタンダードという世界の狙われやすさ、怖さ、もろさを考えさせられました。

（編集長）

編集部より

次世代のソフトウェア・エンジニアリング等に関して、忌憚のない意見をお待ちしております。下記の FAX またはメールにてご連絡ください。

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

SEC journal　編集委員会

「第 11 回情報セキュリティ EXPO」　　（撮影：M. Sugisaki）

SEC journal®　第 10 巻 第 2 号（通算 39 号）　2014 年 7 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。

ITパスポート試験 のご案内

ー ビジネスにＩＴを活用する すべての社会人のための「国家試験」ー

- ビジネスにＩＴを利活用するためには、情報システム部門に限らず、利用する側の社員一人ひとりにも“ＩＴ力”が求められています。
- ｉパス（ＩＴパスポート試験）は、セキュリティ、ネットワーク等のＩＴに関する基礎知識をはじめ、企業活動、経営戦略、会計や法務、プロジェクトマネジメントなど、幅広い総合的知識を測る国家試験です。
- ｉパスを通じて、社員一人ひとりに“ＩＴ力”が備わることにより、組織全体の“ＩＴ力”が向上し、様々なメリットが期待されます。

ｉパスのメリット

ＩＴを活用した業務効率化とビジネス拡大に！

ｉパスを通じて習得したＩＴの基礎知識を活かすことで、業務にＩＴを積極的に活用し、業務効率化につながります。また、ＩＴに関する基礎知識は、社内の情報システム部門等との円滑なコミュニケーションにも役立ちます。営業職であれば、顧客に対して製品やサービスを具体的にわかりやすく説明できるようになり、顧客のニーズをより深く把握できるようになり、ビジネスチャンスの拡大にもつながります。

情報セキュリティ対策・コンプライアンス強化に！

社員一人ひとりが、情報セキュリティやモラルに関する正しい知識を身につけ、意識することで、情報セキュリティに関する被害を未然に防ぐことができ、「情報漏えい」などのリスク軽減、企業内のコンプライアンス向上・法令順守に貢献します。

経営全般に関する知識など幅広い知識がバランスよく習得できる！

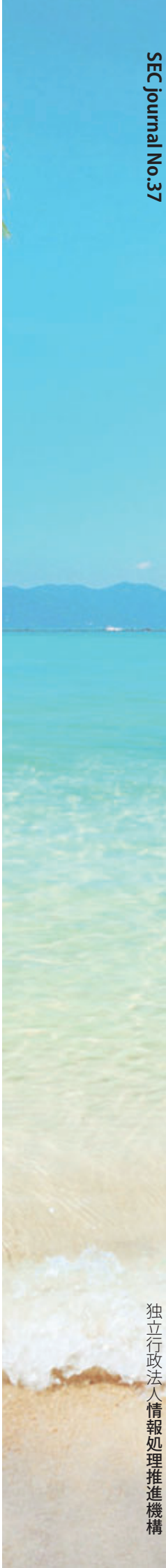
ｉパスは、ＩＴに関する知識にとどまらず、企業活動、経営戦略、会計や法令など、ＩＴを活用する上で前提となる幅広い知識がバランスよく習得できます。そうした知識が身につくことにより、業務の課題把握と、ＩＴを活用した課題解決力が備わり、組織全体の業務改善につながります。

詳しくは、ｉパス Web サイトをご覧ください。https://www3.jitec.ipa.go.jp/JitesCbt/index.html
※企業の活用事例、企業の声、合格者の声など魅力的なコンテンツがご覧になれます。

IPA Better Life with IT

SEC jounal No.37
第 10 巻第 2 号（通巻 39 号）
2014 年 7 月 1 日発行

ISSN 1349-8622



SEC journal No.37

独立行政法人情報処理推進機構