2014年1月31日発行
第9巻第4号（通巻37号）
ISSN 1349-8622

SEC journal 35

巻頭言
新 誠一
技術研究組合制御システムセキュリティセンター（CSSC）理事長
国立大学法人電気通信大学情報理工学研究科　教授

所長対談
小川 紘一　東京大学　政策ビジョン研究センター シニア・リサーチャー
元東京大学知的資産経営・総括寄付講座 特任教授
ソフトウェア・リッチ時代の産業発展と標準化を考える

SECjournal 論文賞
SECjournal 論文賞 受賞論文発表

技術解説
アシュアランス技術を用いた鉄道信号の革新
松本 雅行　東日本旅客鉄道株式会社

ディペンダブルシステム構築と運用の技術
屋代 眞、髙村 博紀、松原 茂　独立行政法人科学技術振興機構　ディペンダブル組込み OS 研究開発センター

オープントレーサビリティツールプラットフォーム TERAS
宮本 貴之　キャッツ株式会社／一般社団法人 TERAS
渡辺 政彦　キャッツ株式会社／一般社団法人 TERAS
高田 広章　名古屋大学大学院／一般社団法人 TERAS
鶴保 征城　学校法人専門学校 HAL 東京　校長／一般社団法人 TERAS　理事長

制御システムセキュリティへの対応
入澤 康紀　IPA／情報セキュリティ技術ラボラトリー　研究員

事例紹介
パッケージソフトウェア品質（PSQ）認証制度創設1年目の展開
中野 正、鈴木 啓紹　一般社団法人コンピュータソフトウェア協会（CSAJ）PSQ 認証室

論文
ソフトウェアプロダクトラインの エンタプライズ・システムへの適用と評価
中村 伸裕　大阪大学　住友電気工業株式会社／谷本 收　住友電工情報システム株式会社
楠本 真二　大阪大学

組織の活動紹介
制御システムセキュリティセンター活動紹介
小林 偉昭　技術研究組合制御システムセキュリティセンター（CSSC）　専務理事　研究開発部長　CSSC 認証ラボラトリー長

報告
Embedded Technology 2013（ET2013）出展報告

Column
夢でなくなった物体瞬間移動！？

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

SEC journal No.35
Vol.9 No.4 Jan. 2014

ソフトウェア高信頼性とサイバーセキュリティ

技術研究組合制御システムセキュリティセンター（CSSC）理事長
国立大学法人電気通信大学情報理工学研究科 教授
新　誠一

現在、社会にサービスを提供している組込み系ソフトウェアのサイバーセキュリティ関連の活動を紹介する。合わせて、SEC が中心となって進めているソフトウェアの高信頼性手法をサイバーセキュリティ確保へ適用していくことを提案する。

2010 年夏に出現した stuxnet はイランの核燃料施設を攻撃した。このことが大きな転機となって、制御系のサイバーセキュリティ問題への取り組みが本格化した。制御システムセキュリティセンター（CSSC）は、2011 年 3 月に経済産業省の技術研究組合という形で民間企業を中心として発足した。IPA には組合員として当初から貢献頂いている。この場を借りて御礼申し上げる。

2012 年 7 月にお台場にある産業総合研究所内に東京研究センター（TRC）を設置し、2013 年 4 月には被災地である宮城県多賀城市に東北多賀城本部（TTHQ）を設置した。ここには国の補助を受け 7 台のテストベッドを研究、啓蒙、訓練、演習、標準化、認証などの業務を行うために作成した。TTHQ のお披露目を 2013 年 5 月に行って以来、国内外から多数の見学者に訪れて頂いている。詳しくは、本特集中の当組合の小林専務理事の解説を参照願いたい。

本稿では題目に従い、ソフトウェア高信頼性とサイバーセキュリティの問題を論じたい。まず、制御にソフトウェアをなぜ用いる必要があるかという根源的な問いについてであるが、それは「便利」だからである。容易にアルゴリズムや制御装置内のパラメータを変更できるので、時代、季節、個人などの状況に合った制御を行える。それ以上に、開発段階で何度も手直しが出来る。その意味では、「ソフト」である便利さとサイバーセキュリティの脆弱性は裏腹の関係である。

もちろん、ソフトの改変には認証などのセキュリティ機能が搭載されているものもある。しかしながら、国会や官庁、有力企業のサーバーから情報が引き出されている現状を鑑みると、現状のセキュリティ対策だけに頼るのは危険である。しかも、制御系は情報ネットワークだけでなく、制御ネットワーク、デバイスネットワークと攻撃される場所が多様である。制御システムセキュリティを確保するには情報セキュリティ技術と制御システム技術と両方に精通している必要がある。もっとも、片方の技術を習得するだけでも困難なのに、両方習得は無理である。そこで、連携が不可欠である。

この連携という視点で見ると、ソフトウェア高信頼性技術とサイバーセキュリティ技術は関係が深い。セキュリティが脆弱であれば、高信頼性とは言えない。また、逆に高信頼性のために磨いてきた手法がサイバーセキュリティにも有効である。例えば、相互レビューや相互依存性解析などの手法には注目している。

その中でも、HAZOP 解析 [1][2] は制御系セキュリティと関係が深い。この解析は、もともと化学工業における安全性確保で生まれたものである。これをソフトウェアに適用することで操作ミスや故障に強いソフトウェア、すなわち高信頼性を担保している。

制御系においても、すべてにセキュリティ対策をすることは難しい。システムの機能レベルを分析して、コストを懸けるべき所を洗い出す必要がある。その意味で、セキュリティ要件を HAZOP に入れていくことが高信頼性と高セキュリティを両立させる早道だと思う。

もちろん、相互レビューも FTA や FMEA、単体試験に統合試験などの各種高信頼性技術も有効である。サイバーセキュリティ対策も含めて、SEC でソフトウェアの高信頼性を研究開発して頂ければ幸いである。

【参考文献】
[1] http://www.ipa.go.jp/files/000005325.pdf
[2] http://www.ipa.go.jp/files/000004108.pdf

ソフトウェア・リッチ時代の産業発展と標準化を考える

東京大学 政策ビジョン研究センター
シニア・リサーチャー
元東京大学知的資産経営・総括寄付講座 特任教授
小川　紘一

×

SEC 所長
松本　隆明

これまで日本の製造業は、「すりあわせ型」の技術が強みだといわれてきた。しかし、世界を見れば、多くの分野が「組み合わせ型」の製造モデルにシフト。既成のモジュールを組み合わせることで完成品が出来上がるので、開発時の高度なすりあわせ作業が不要なのだ。同じような事業プロセスのパラダイムシフトは、ソフトウェア産業でも見て取れる。産業の国際競争力と国際標準化に詳しい小川紘一氏を招き、産業全体の観点から話を伺った。

松本：世界の産業は、組み合わせ型のプロセスに代表される「モジュール化」志向により、タイムリーに市場に製品を提供する方向に移行しています。さらには、モジュールを連携・制御するためにはソフトウェアが大きな役割を果たしますが、日本の産業界はこうしたパラダイムシフトに対して海外に遅れを取っているように思います。

小川 紘一（おがわ　こういち）

電子材料の研究で博士号取得後の 1973 年に富士通研究所入社。研究部長を経て 1992 年に富士通ビジネス部門の事業部長に就任。富士通の理事を経て、2004 年から東京大学で、基礎研究から事業部経営に至る経験をベースに、日本の国際競争力、国際標準化と事業戦略、イノベーション政策、知財マネージメントなどの研究に従事。現在、内閣官房にて「国際標準化戦略タスクフォース委員」や経済産業省の「産業構造審議会・情報経済分科会」の委員、「産業構造審議会・研究開発小委会」の委員を務める。

小川：液晶パネルや CD-ROM、DVD などは、日本企業が主導して技術を開発し、市場を開拓して普及させた製品です。しかし、日本企業はいずれも同じ弧を描いてシェアを落とし、市場撤退を繰り返しました（157 ページの図参照）。これだけ何度も同じことが繰り返されるのですから、その背後に共通の原因があるとしか考えられません。

これまで日本企業は、熟練のオペレーターや技術者によるすりあわせ型の技術で、外国と差別化してきました（ハードウェア・リッチ型）。複数の部品を互いに調整しながら、高品質の完成品を作るモデルです。しかし、モジュール化とソフトウェアが産業構造を変え、競争ルールまで変えてしまったのです。今ではデジタル家電の設計工数のうち、6 割以上がソフトウェアの開発になってモジュール化が急速に進み、昨日まで畑仕事していた人でも、今日から製造ラインに加わって最先端のテレビを作ることができます。この意味で工場中心のモノづくりから付加価値が消えようとしています。

私は、こうした産業領域をソフトウェア・リッチ型と呼んでいます。競争に敗れた日本企業の多くは、産業の主流がソフトウェア・リッチ型に移行したにも関わらず、ハードウェア・リッチ型の発想をそのまま引きずって、ビジネスに取り組んでいました。これは日本企業だけに起きたことではありません。1980 年代の米国や 1990 年代のヨーロッパでも同じように、ソフトウェア・リッチ型に変わった産業領域で伝統的な企業が競争力を失いました。

ソフトウェア・リッチ型になると技術があっという間に国境を越えて新興国に伝播するので、これまでとは全く違った競争ルールを考えた人がビジネスに参入します。ソフトウェア・リッチ型の産業であれば技術も知財も、そしてものづくりさえも瞬時に国境を超えるようになりますので、経営のオペレーションコストやその国の税制など、技術以外の要因で競争に勝てる仕組みを作ることも可能になり、例え技術的な蓄積が非常に少ない新興国の企業であっても市場参入ができます。

オープン＆クローズ戦略で勝ち抜く

松本：モジュールに適用されている差別化技術を特許として、知的財産権で守っていくことはできないのでしょうか？

小川：知的財産で守るには工夫が必要です。これまで日本企業はできるだけ多くの特許を出願・登録して差別化技術を守ろうとしてきましたが、この考えが本質的に機能しなくなっています。むしろ特許を出願して公開する

ことは、技術漏洩の主要ルートになっているからです。公開特許の情報を活用する２番手企業の方が、先行する企業よりＲ＆Ｄ費用が35％少なくて済むという分析結果が先進国にあります。さらにこれが新興国なら、先進国企業より60％以上も少ないＲ＆Ｄ費用で済むようです。

一般に日本の企業は、特許を国内に出願して外国出願しないことが多い。従って１年半後には公開情報になってしまいますので、新興国がその特許情報を使って製品を開発しても知財侵害にはなりません。特許を出願することが結果的に技術を漏洩させたのと同じことになってしまうのです。製品がソフトウェア・リッチ型になってモジュール化が進むということは、このような産業構造が大規模に出現することだったのです。

松本：ソフトウェアも特許申請できますが、確かに技術漏洩の防止策としては、あまり有効ではないかもしれません。著作権で守るという発想もありますが、ソフトウェアの場合、内容を少し書き換えるだけで、簡単に同じようなことができてしまいます。ソフトウェア・リッチ型の産業構造に変わり、そもそも知的財産権を守ることが難しくなってきました。

小川：日本はハードウェアを動かすためのソフトウェア、すなわち組み込みソフトウェアを知的財産にする努力をしなければならないと、私は考えています。組み込みソフトウェアを日本の得意なハードウェアと連携させた知財の取り方に知恵を絞り、トータルな技術体系として技術を守って行くべきです。ヨーロッパでは成功しているので、日本にできないはずはありません。

例えば、自動車サプライヤーのボッシュがその代表的な成功事例であり、新興国では日本のサプライヤーがエンジンECUでボッシュに勝てなくなりました。これまでの日本企業はそういう意識が弱かっただけです。ソフトウェアとセットにした知財マネージメントとセットにしたビジネスモデル駆使の重要性に気が付けば、日本企業もやれるはずです。

ヨーロッパはハードウェアをアジアから調達してこれをやりますが、日本には非常に強いハードウェアが身近にたくさんあるので、ヨーロッパより遥かに強力な技術体系として守れるはずです。

松本：カギになるのはオープン＆クローズ戦略でしょうか。他者が自由に使えるオープンの技術情報と、表に出さないクローズの技術情報、両方を持って競争しなければなりません。クローズ一辺倒で、技術情報を守ろうとしても無理があります。

アップルのように、iOSというプラットフォームを用意し、そのインターフェースをオープンにすることで、その上で様々な人がアプリを開発し、それに伴ってプラットフォームも売れる、というスキームが一つの成功例になると思います。

小川：iOSにはエレクトロニクス産業で最も尖鋭的なオープン＆クローズ戦略が隠されています。これを支えているのが知財権を武器にした目に見えない契約マネージメントです。それ以外に、航空機産業や自動車産業、半導体産業など、ほぼすべての巨大産業で目に見えないプラットフォームが、かなり複合的な技術体系とこれを支える知財や契約マネージメントによって構築されています。したがって一部の技術が他社に漏洩したからといって、あるいは国境を越えたといっても技術の全体系がコピーされることはありません。

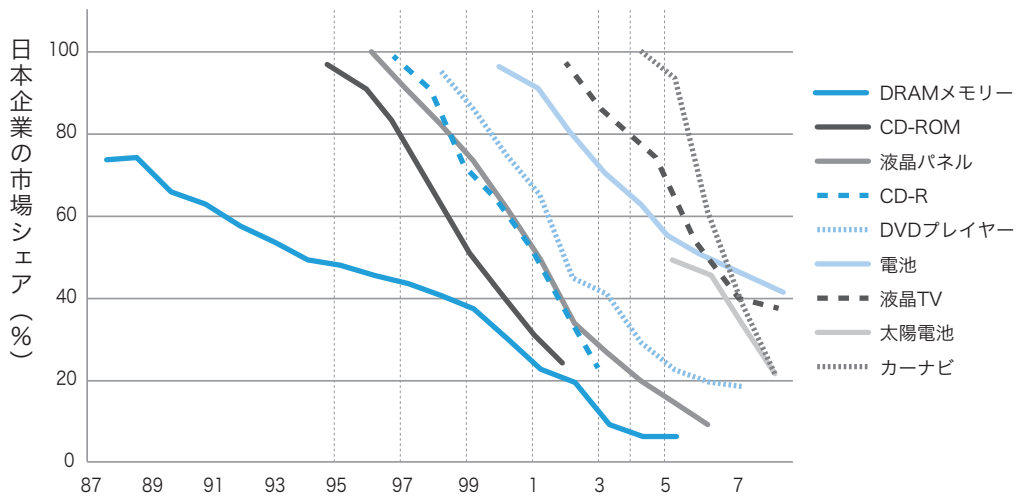
例えば、ボーイング787は完全にボーイング社が自社優位に構築したプラットフォームの上で開発されて

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）。

技術イノベーションで巨大需要を創出したはずの日本企業が

大量普及のステージになると市場撤退



います。日本にも個々の技術モジュールを理解し、作れる企業もありますが、全体像はプラットフォームを作ったボーイングしか分かりません。従ってパソコンのように、日本の部品メーカーに付加価値が集まることはありません。このようなプラットフォーム構造は、CAD/CAEなどを駆使するソフトウェア技術によってはじめて具体化されているのです。この意味でボーイングはソフトウェア・リッチな企業といってもいいでしょう。

自動車産業でも、目に見えるハードウェア技術ではなく、目に見えないソフトウェアの技術体系とハードウェア技術体系との緊密な連携によってビジネスの成否が決まるようになりました。確かに組み立て工場の段階で約３万点もの部品があるので、サプライチェーン・マネジメントと工場内の製造技術や生産技術、生産管理が自動車メーカーの強さの秘密だといわれてきました。しかしこれが競争力の源泉だったのは1990年代までです。

1990年代の後半になると、ソフトウェアが自動車の基幹部品のイノベーションで重要な役割を担うようになりました。例えばエンジン制御、モータ制御、電池制御、ブレーキ制御、ステアリング制御など、一つひとつはそれぞれのECUの中の組み込みソフトで最適制御されています。組み込みソフトがなければ、1980年代に自動車の進化が止まっていたでしょう。

さらに2000年代になると、自動車の最も重要な価値がソフトウェアによって決まるようになってきました。例えばハイブリッド車では、基幹技術を単純に組み合わせても燃費のよい車にはなりません。完成品の自動車としてあるべき全体最適の視点から、個々の技術モジュールを連動させながら最適制御することによってはじめて、従来の３倍以上に燃費が改善し、排気ガスを極端に減らす環境車になるのです。個々のテクノロジーでなく総合力で付加価値が生まれる、と言い換えてもいいでしょう。

再度強調したいのですが、この統合制御を支える技術体系がハードウェアではなくソフトウェアなのです。今後、絶対に衝突しない自動車、あるいはネットワークに繋がって新たな付加価値を生み出す自動車なども次々に現れますが、このようなプロダクトイノベーションや社会システムイノベーションをリードするのが、ハードウェアではなくソフトウェアなのです。この意味で自動車の価値を決めるのは、間違いなくソフトウェアになります。例え電気自動車になっても、あるいは燃料電池車になってもこの事情は変わりません。

松本：生産管理をきちんと行うことで、自動車を設計し、最適制御していくシナリオの全体像に強みがあるということですね。日本の企業は、そのあたりをもっと意識して良いのではないでしょうか？

小川：その通りです。自動車とスマホやパソコンが違うのは、モジュールの単純な組み合わせだけで完成品ができあがるか否かにあります。部分最適の積み重ねで済むか（パソコン）あるいは全体最適の追及か（自動車）が、が大きな違いです。例え電気自動車でも、調達した部品を全部集めて組み合わせるだけでは乗用車にはなりません。乗用車としての全体最適は統合型のソフトウェア技術体系によって実現されます。自動車メーカーの付加価値がここから生まれます。

日本企業はハードウェア・リッチな製品で競争力が非常に強かったので、ソフトウェアを駆使した統合化あるいは全体最適を実現させるノウハウがどんなに素晴らしいものかを、まだ理解できていない可能性があります。これを理解した上でモジュール化を徹底し、ソフトウェア技術で全体最適化するようなモデルを作れば、例え新興国企業の攻勢に遭っても日本の製造業はグローバル市場で間違いなく勝てるはずです。

ビジネス視点の標準化を

松本：オープン＆クローズ戦略ということでは、私が長年携わってきたエンタプライズ系のソフトウェアも、できるだけモジュール化して、既存のものをうまく再利用する発想で作ります。しかし、再利用する部分と差別化する部分を決めるのがとても難しく、モジュールの粒度や範囲の決め方が重要なポイントになります。

小川：オープン＆クローズ戦略を実行するために、公開する部分と秘密にする部分をどのように判断するか、が最も重要な経営判断となります。オープンソースやパッケージソフトを使って効率化したい。そうはいっても差別化するにはどうするかなどの判断を、エンジニアに任せると、テクノロジーの面から考えようとします。

今後の日本企業は、まずビジネスモデルや経営戦略の視点からオープンとクローズの境界を決定しなければなりません。しかし経営者やこれに準ずる幹部で、ソフトウェアの重要性を経営の視点から理解する人が少ない。したがって経営の視点ではなく技術者による技術的な視点が優先されてきました。

松本：ご指摘の通り、技術者の観点で考えることが、日本の企業では非常に多くあります。オープン戦略に欠かせない標準化団体の議論も、ほとんどが技術ベースです。経営やビジネスの視点で議論すべきなのでしょうが、実際には難しいと思います。

小川：国際標準化の目的は、これを主導する企業が自社の競争力をつけて大量普及と高収益を同時実現することにあります。標準化することは、それ自体が目的では決してありません。標準化とは事業戦略そのものなのです。ここでは、最初に標準化する領域（オープン）とブラックボックス化する領域（クローズ）を事前に設計しなければなりません。

多くの製品がソフトウェア・リッチになってきたので、

これも結局ソフトウェアの問題に帰着するのです。いずれにせよオープン領域とクローズ領域の境界を自社（自国）優位に決めるのは、自社（自国）の技術が圧倒的優位になっていないと、非常に難しい。

松本：経営戦略論より技術論のほうが、合意が取りやすいのでしょう。企業内においても、何十年も標準化に携わり、高度なスキルもありながら処遇が良くない、という問題があります。標準化活動は、経営戦略に直結する重要な企業活動であるという認識が少ないように思います。

小川：標準化に携わる人の処遇が日本企業の中で良くないというのは、標準化を経営の問題として捉えて来なかったからです。標準化を自己目的にする人が非常に多く、企業の国際力に貢献できなかったからです。標準化とは事業戦略そのものですので、オープン（標準化）とクローズ（利益の源泉）の境界を事前に設計できない人が、標準化に関与すべきではありません。標準化とは規格を作ることではありません。規格を作る前にオープン＆クローズの思想で自社の勝ちパターンを事前設計できないのであれば、オープン標準化に参加すべきではありません。ここにもソフトウェアと同じ経営上の課題が日本企業にあります。

松本：経営の視点から策定したオープン＆クローズ戦略に基づき、オープンにした自社の技術を標準にすることができれば、これほど強いことはありません。皆が使わざるをえないのですから、クローズにした部分で利益を稼ぐことができるはずです。

適地良品を求める

松本：オープン化、標準化が進む一方で、顧客に対してどのような価値を訴求するかが難しくなりつつあるように思います。自動車の場合も、インドのメーカーが30万円以下で車を売っています。誰でも、部品や工場さえ用意できれば、そこそこの自動車が完成する時代になると、付加価値を見いだすのが難しくなってくるのではないでしょうか？

かつての日本は、単なる組み合わせではなく、全体の品質が担保されている、使い勝手が良いなど、他者がなかなかできないところに強みがあったわけです。

小川：この問題を「品質」という言葉で表現すると、少し狭い意味に取られる恐れがあります。肝心なのは、自動車を買う人がトヨタや日産、ホンダ、スズキのブランドに期待するものを、そのクルマが備えているか、ということであり、価格も品質もその一部にすぎません。

高いブランド力を維持して期待通りの製品を作り続けられるのは、大規模企業の強みです。しかしながらパソコンのように部品の単純組み合わせで完成品ができあがり、全体最適で差別化し難い製品では、単純に狭い意味の品質やコストだけの競争になってしまいます。このような製品領域では、日本企業はアジアの企業に敵わないでしょう。しかし幸いなことに自動車では、例え途上国であっても部品の単純組み合わせで作る安いクルマは売れないのが現実です。

松本：以前米国では、日本車は貧弱で壊れやすいと考えられていたそうです。今では頑丈さが認知されていますが、かつての日本車のように、ブランドの訴求に問題があるケースが多いのでしょうか？

小川：日本の製品なら素晴らしいはず、というブランド力が1980年代から世界中に広がっているのではないでしょうか。しかし21世紀の日本企業が注意しなければならないのは、巨大市場が先進国から新興国へ移ったことにあり、新興国ではやみくもに高い過剰品質が通用しないことです。新興国市場で受け入れられるには、「適地良品」という設計思想がキーになります。品質それ自身ではなく、使う人たちのライフスタイルに適応した製品に価値があるのです。初めて自動車に乗る新興国の人に、ポルシェやランボルギーニの素晴らしさを語っても理解してもらえないでしょう。しかも値段が非常に高いのですから。

松本：エンタプライズ系のソフトウェアでも、日本のお客様は要求水準が非常に高い。サービスインしたらバグは基本的には出さないのが当たり前という大変な高品質を求められます。

ところが外国の企業は、「バグは当たり前で直せばいい」という考え方がほとんどです。日本のマーケットに慣れている企業が、自慢の品質を海外に持って行っても訴求できない、というジレンマはあるでしょうね。

小川：すべての日本企業がこのジレンマに直面しています。だからといって、日本製品の品質を下げて良いというわけではありません。例えばインドで売られるトヨタ車では、100万円以下の自動車よりも200万円の自動車がよく売れているそうです。タタ・モーターズの超低価格車である「ナノ」もあまり売れていません。その2倍も3倍もするクルマが売れている。

途上国の人にとっては、自動車を持つこと自体がステータスになります。そのためには、安物では意味をなさない。所得格差が大きいため、このような現象が起きるわけですが、途上国の市場で勝つには、値段を下げるのではなく、そこに住む人々の価値観やライフスタイルに適応した「適地良品」の車を開発し、同時にブランドを徹底して磨く、という手段こそが有効なのです。

繰り返しますが品質を下げてコストを下げるのではなく、適地良品という視点からムダを省いてコストを下げ、その国の人が良いと思う機能だけを取り込むべきです。そのために、開発する人が日本にいて設計してはだめで、現地で製品コンセプトと仕様を考えるべきです。本社からは「日本の基準にあわない」と反対されるかもしれませんが、それでもやらなければなりません。でなければ、

新興国の人に受け入れられないのですから。

松本：このようなモデルでは、様々なマーケット向けの製品ラインアップが必要になりますね。

小川：そこで、先ほどのプラットフォームが重要性を増します。基本的なプラットフォームは日本で作り、現地にテクニカルセンターか開発センターを置いて適地良品へカスタマイズする。そうでなければ、本当に現地のユーザに合った製品は作れません。

定石を知る軍師型人材を育てる

松本：SEC でも、今年度から利用者の視点を取り入れた新たな事業を始めました。例えば、これまで通り信頼性や安全性の向上を追求することは変わりませんが、更に「利用者にとっての安全性とは何？」という観点から、開発や運用を捉えていこうとしています。

小川：先ほどお話ししたように、アジアに行くと、日本に住む私たちの常識でマーケットを理解することはできません。日本ではなくアジア市場に住み、ここから適地良品の在り方を考えられる人材を数多く育成できれば、日本が持つ高度な製品文化がアジアで必ず受け入れられるようになります。アジアには約 30 億人もの人がいて、毎年 5%から 10%のペースで経済成長していますので、所得水準がドンドン上がって豊かになり、日本企業が提供するいい製品への需要が急速に高まってきます。

松本：自動車のユーザ視点も変わっています。かつてはドライバーの思うとおりに動くのが自動車でした。ところが、自動運転の場合にはドライバーが意図しない動きでも、その方がドライバーにとって安全ならそのように動く方が望ましいわけです。

小川:確かに今後は、自動車の開発のコンセプトが変わってきます。1990 年代から自動車にマイクロコンピューターが本格的に導入され、これが当たり前になると今度はエンジンとブレーキやステアリングとの連動制御が求められるようになりました。ここから自動車の価値が連動制御、すなわち統合型の全体最適に移行します。したがって自動車に期待する価値基準が変わり、競争の構造も変わります。

これを組み込みソフトの視点で語れば、ソフトウェアのソースコードをサプライヤーが持つかあるいは自動車メーカーが持つかで、自動車産業の競争力が大きく変わるでしょう。

例えばヨーロッパのオートザーで標準化された自動車のベーシックソフトウェアは、基本的にオープン化されているので、これまでのサプライヤーだけでなく新たに市場参入するサードパーティーでも作れます。

もし自動車メーカーがサードパーティーにソフトウェアだけ開発を委託して買い取り、それを使ってエンジン制御やブレーキ制御を作るようにサプライヤーへ発注したらどうなるでしょう？ サプライヤーにとって提供できる付加価値がハードウェアだけになるのではないでしょうか。明らかにサプライヤーから付加価値が失われ、サプライヤーの競争力が相対的に弱まって自動車の完成品メーカーに付加価値が集中します。こんな動きが既に見え隠れしはじめました。

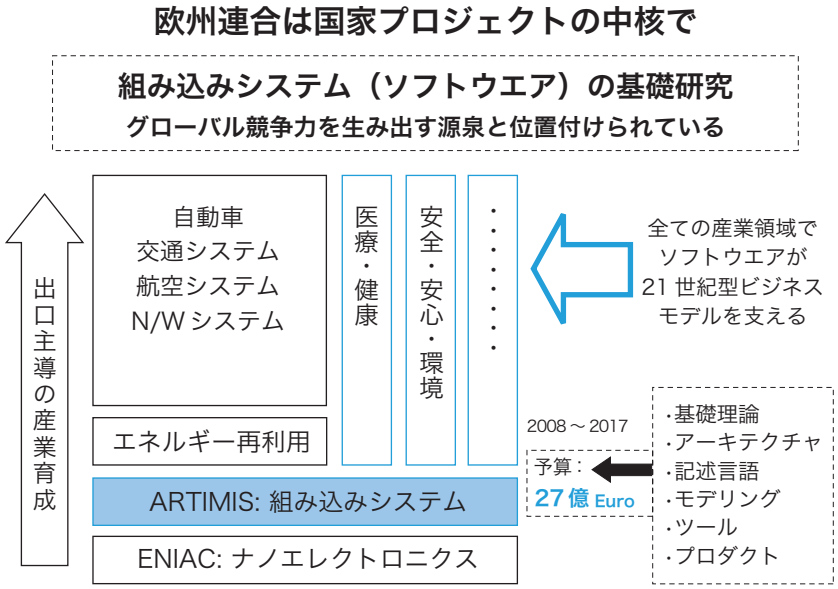
組み込みソフトが自動車設計に広く使われて不可欠になる 21 世紀の現在では、ソフトウェアの持つ機能を経営の問題として捉えなければならない背景が、ここからも理解されるのではないでしょうか。我々はソフトウェアが産業構造も競争ルールも変えてしまう事実をもっと深く理解する必要があります。

松本：エンタプライズ系のソフトウェアでも、請負開発の場合には成果物の権利を誰が持つかはケースバイケースです。大体は発注側が持つケースが多いですが、駆け引きはかなり重要になります。権利を発注側が持つと、受注側ではそれをモジュール化して再利用することができなくなります。

小川：こうした問題を経営視点から捉えて、事業戦略にくみ上げる「軍師型人材」を育てなければ、日本のソフトウェア産業はもとより、日本の製造業そのものが衰退していきます。軍師型人材がいて初めて、オープン＆クローズの戦略を駆使しながらグローバル競争力を維持発展させ、国の雇用と経済成長に貢献するのです。これも製造業がソフトウェア・リッチ型になって初めて顕在化したことでした。

松本：軍師型人材はどのように育てますか？ 経験を積ませるしかないでしょうか？

小川：一歩一歩経験を蓄積して行くには時間がかかり過ぎて先を行く欧米に追いつけ



ません。まずやるべきことは、欧米企業がこれまで 20 年以上にわたって蓄積した成功モデルを分析することです。この分野で数多くの成功モデルと定石を知る人を外国からどんどん連れてきて、日本で人材を教育しなければなりません。定石を学び、これをそれぞれの企業の実体に合わせて応用しながら人材を育成するのです。明治政府が、当時もっとも進んだドイツの参謀本部から非常にレベルの高い人材を招へいして参謀を育成した意味を、もう一度思い起こさなければなりません。

松本：最後に、IPA や SEC に対するご要望をお聞かせください。

小川:何より、軍師型人材を育成してほしいと思います。それが現在 IPA や SEC のミッションでないなら、ミッションを自らの手で拡大していただきたいのです。もしそれもできないのであれば、今日お話ししたようなことを多くの人が議論できる場を作れないでしょうか？ 現在の日本で、少なくともソフトウェアの分野については、公的機関としてこれを実行できるのが IPA だけなのですから。

私が少しだけ関係を持つ NEDO では、世界各国の産業の動きや競争政策、知財政策などを知る人材を招き、何度も意見交換をしました。そうすると NEDO の内部でもマインドが変わってくるのです。学ぶことにお金を惜しんではいけません。こんなことは､わずかなお金でやれますので､これをやろうとする意志決定だけが問題のはずです。

IPA も海外のソフトウェアの専門家を呼び、特に欧米企業が事業戦略の中でソフトウェアをどのように位置付けているか、ソフトウェアが競争政策の中でどんな位置付けになっているのか、またソフトウェアの専門家がどう育成され、どう処遇されているのか、などを我々が理解することから始めるのも一案です。

あるいは、インドのソフトウェア関連企業が日本にオフィスを構えていますので、彼らを IPA や SEC に呼んで意見交換するだけでも非常に効果的なはずです。私自身、インドへ行くたびにソフトウェアで驚くことが多く、開発プロセスにしろ品質管理にしろ、日本はもうインドに敵わないのではないかとさえ思っています。

日本企業も、コア領域のソフトウェア開発だけを国内でやって、あとはインドにオフショアリングしている企業が多いはずです。ここでもオープン＆クローズの経営思想を持たないとオフショアリングがうまくいきません。それ以前に我々がオフショアリングのための仕様書が書けないことも大きな問題ですので、仕様書の標準化が必要です。

松本：SEC でも仕様書を含む開発プロセスの共通フレームを作る、という形で取り組んでいます。しかし、それで仕様書が書けるレベルにはなりません。

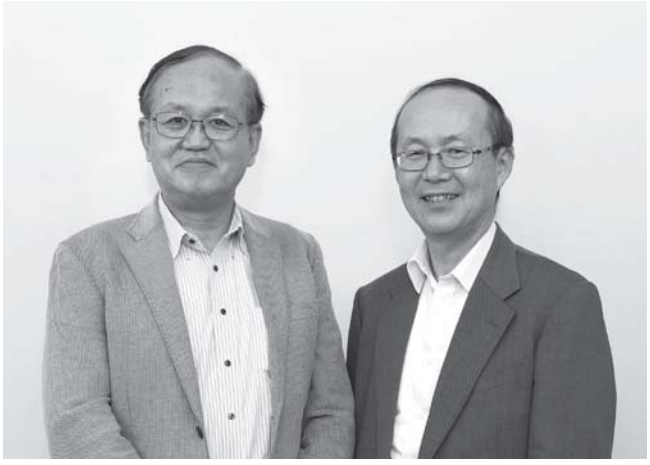
日本では、オフショアの場合でも、開発する会社側に仕様書の内容を理解できる人がいて、「あ・うん」の呼吸で開発するようなケースが依然として多いのではないでしょうか。

小川：適切な仕様書が書けるのは、ソフトウェア・リッチ型へ移行した 21 世紀の産業に必須の組織能力です。1980 年代の後半からオフショアリングに取り組んだ IBM も、実は大変な苦労をしながらこのノウハウを身につけ、1990 年代に躍進しました。我々は同じ苦労をしないために、この点だけでも先に苦労した欧米企業から学ぶ必要があります。

その点､IPA が標準化を後押しした「Ruby」は､オフショアリングにまつわる問題を解決する大きな手段ではないでしょうか。プログラミングの工数は人件費そのものですので、もし Ruby を使うとプログラミング効率が 5 ～ 10 倍も高くなるのであれば、ソフトウェアの開発コストが 1/5 あるいは 1/10 になることを意味します。組み込み用の Ruby もドンドン進化してきました。今後さらに進むマイクロプロセッサーの技術革新が、Ruby の応用範囲をドンドン広げてくれるはずです。スーパコンピューター「京」で使える Ruby すら開発されています。

また Ruby は仕様書を書けない人でも気軽に使えるような言語ですので本当にすごい。Ruby 言語を高校はもとより中学校でも必須科目として教えれば、国民総プログラマー時代が到来し、グローバル市場で日本企業の競争力が復活するはずです。この中から必ず軍師も生まれてきます。

松本：SEC もグローバルな視点を忘れずに取り組んでいきたいと思います。本日は貴重なお話をありがとうございました。



小川紘一先生の主な著書

- 『オープン･アンド･クローズ戦略―日本企業再生の条件』（翔泳社）
- 『国際標準化と事業戦略』（白桃書房）
- 「The Effect of Technological Innovation on the International Division of Labor」（共著）『A.Gower ed. Platform,Market and Innovation』、Cheltenham UK and Northampton, MA,US, Edward Elgar)

SECjournal 論文賞 受賞論文発表

SEC は、我が国ソフトウェア産業発展のための様々な取り組みを実施しておりますが、その取り組みの一つとして、ソフトウェア工学に関する論文に賞を設け表彰を行っております。

今年の SECjournal 論文賞は、2011 年 7 月から 2013 年 6 月までに投稿された合計 18 編のうち、査読者により採録された 8 編の論文を候補とし、選考委員会と表彰委員会による厳正な審査の結果、3 編を選出いたしました。

各賞の発表と表彰式は 2014 年 1 月 17 日に第 11 回 WOCS[2] と併催で実施されました。本年は最優秀賞 1 編、優秀賞 1 編、所長賞 1 編が表彰され、片山表彰委員長による審査報告は本号 165 ページに掲載されています。なお、3 件の受賞論文は 34 号に掲載されています。

SECjournal 論文賞表彰委員会 委員

委員長	片山　卓也	北陸先端科学技術大学院大学 学長
委員（50 音順）	有賀　貞一	AIT コンサルティング株式会社 代表取締役社長
	岩野　和生	独立行政法人科学技術振興機構 研究開発戦略センター 上席フェロー
	大原　茂之	スキルマネージメント協会 理事長
	國井　秀子	一般社団法人情報サービス産業協会 副会長
	林　　弘	独立行政法人情報通信研究機構 監事
	松田　晃一	独立行政法人情報処理推進機構 顧問
	松本　隆明	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

SECjournal 論文賞選考委員会 委員

委員（50 音順）	飯泉　紀子	株式会社日立ハイテクノロジーズ 研究開発本部 第四部 主管技師
	大島　啓二	元株式会社日立情報制御ソリューションズ 技師長
	神庭　弘年	一般社団法人 PMI 日本支部 会長
	楠本　真二	大阪大学 大学院情報科学研究科 教授
	紫合　治	東京電機大学 情報環境学部 情報環境学科 ソフトウェア工学研究室 教授
	新谷　勝利	新谷 IT コンサルティング 代表
	寺中　勝美	NTT ソフトウェア株式会社 監査役
	平山　雅之	日本大学 理工学部 教授
	古山　恒夫	東海大学 理学部 情報数理学科 教授
	松本　健一	奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア工学研究室 教授
	水野　修	京都工芸繊維大学 大学院工芸科学研究科 准教授
	神谷　芳樹	みたに先端研合同会社 代表
	峯　　恒憲	九州大学 大学院システム情報科学研究院 情報知能工学部門 准教授
	森崎　修司	名古屋大学 大学院情報科学研究科 准教授
	山城　明宏	東芝ソリューション株式会社 生産統括部 品質保証担当 主幹
	山本　雅基	名古屋大学 大学院情報科学研究科 特任教授
	山本　里枝子	株式会社富士通研究所 ものづくり技術研究所 シニアディレクター
	鷲崎　弘宜	早稲田大学 理工学術院 基幹理工学部 情報理工学科 准教授

選考委員会では、全委員の査読結果を含め、審査を行った。
ただし、委員が著者の論文や委員の関係者の論文については、該当委員は審査を行っていない。

最優秀賞

アプリケーション保守サービスの定量化手法

酒井　大

優秀賞

システム価値向上を目的としたScrumの試行・評価

中村　伸裕、服部　悦子、永田　菜生、楠本　真二

所長賞

若年技術者向け ソフトウェア開発研修プログラムの開発と評価

大森　久美子

SECjournal 論文賞 2013

上段左より、松田　晃一、松本　隆明、岩野　和生
片山　卓也、大森　久美子、酒井　大、中村　伸裕、藤江　一正

（敬称略）

SECjournal 論文賞　表彰委員会審査報告

SECjournal 論文賞
表彰委員会委員長
北陸先端科学技術大学院大学 学長
片山　卓也

SECjournal は、ソフトウェア開発現場におけるソフトウェアエンジニアリングの実践や実践可能な手法などについての知見を報告し、新しい方法論の開発現場への導入の役割を果たしてきた。表彰委員会は、厳正な審査を行い本年対象論文の中から以下の３編の論文を表彰論文とすることを決定した。いずれもソフトウェア開発現場や研究コミュニティにとって貴重な内容のものである。今後ともこのような優れた論文を積極的に掲載し、わが国のソフトウェア開発技術の進展に貢献してゆきたい。

最優秀賞：「アプリケーション保守サービスの定量化手法」酒井　大

アプリケーション開発のコスト見積もりに関しては、ファンクションポイントやソースコード行数などのメトリックの利用が定着し、それなりに合理的な見積もりを行うことが定着している。その一方、アプリケーションの保守に関しては、その内容が多岐にわたることなどから、保守サービスの定量化的見積もりは困難とされてきた。

本論文は、日本アイ・ビー・エム（株）の社内アプリケーション開発部門における経験を基に、保守サービスの定量的見積もり手法の開発を行った報告である。多様な保守サービスの分類、作業量の見積もりを対象としての設計・作成部分の特定、標準的作業者により行われた標準的サービスとの対比によるサービス保守量の決定、納期や品質要求の厳しさに比例したコストの修正、などにより保守量を定義している。

本手法は極めて実際的・現実的な方法であるが、この手法により精度の高い保守量の算定が行えるためには、よくマネージされた保守業務が行われ、それに関するデータが蓄積されていることが必要である。特に、本手法に含まれる多くのパラメータ（比例定数）を現実データに照らして適切に設定する必要があろう。保守コスト定量化を合理的に行い、それを基に保守業務の改善に取り組んだ貴重な報告である。

優秀賞：「システム価値向上を目的とした Scrum の試行・評価」中村　伸裕

ビジネス変革の要求に応えるため企業情報システムの開発プロセスとして、アジャイルプロセスの評価が高い。その一方で、アジャイル開発については、短寿命のプログラムを公式ドキュメントを残さずにソースコードのみを作るようなイメージが定着していることもあり、企業情報システムの開発のためのプロセスとしての認識が薄い。特にわが国においては、開発実績に関する詳細な文献が少ないこともその利用を妨げている。

本論文は、住友電気工業（株）の情報システム部において、企業内事務処理システムの開発に Scrum と呼ばれるアジャイルプロセスの一つを試行し、その評価を行ったものである。同組織はこれまでも QCD 改善を継続的に行い CMMI レベル 5 を達成しているが、このような実績を踏まえて Scrum 導入による設計品質向上やプロセス改善効果、開発者のモチベーション向上効果などを評価している。4.5 月に亘る試行の内容や従来の Waterfall 型の開発との定量的な比較などが行われ、Scrum 導入の効果が示されている。特に、若い開発者が開発の全行程に参加することの意義やモチベーションの向上など、教育の効果も高いことが示されていることが興味深い。

所長賞：「若年技術者向けソフトウェア開発研修プログラムの開発と評価」大森　久美子

若年技術者の設計力と開発力の向上を目指して、NTT 関連企業社員を対象に実施、改良を続けてきた PBL 研修についての報告である。要求分析工程を重要視し、問題発見から、要求分析、受け入れテストまでのソフトウェア開発の全工程を含んだ研修を設計・実施し、その評価を受講者のレベル向上、研修内容、指導方法について行った報告である。

ソフトウェア開発の経験が不十分な受講生を対象にして 6 日間の短時間の基礎研修とフォロー研修により、開発プロセス、非機能要件、品質管理、プロジェクト管理なども含めた研修を行い、受講者のレベルアップが図られていること、また、知識詰め込み型ではなく気づきや受講者の自主性を重んじる研修が行われたと報告されている。これは、一流企業の修士、博士課程修了新入社員や入社数年の社員を対象としており、受講者のレベルがもともと高いことはあるにしても、研修がきちんとした計画に基づいて実行され、評価に基づいた改良がされていることによるものである。研修の内容が著書として出版され、大学や他機関でも利用されていることは、研修の質の高さを示すものであり評価に値する。

― 受賞者 プロフィール ―

アプリケーション保守サービスの定量化手法

酒井　大（日本アイ・ビー・エム株式会社 グローバルビジネスサービス IGA アプリケーション・サービス）


酒井　大

論文の題材とさせていただいた「アプリケーション保守サービスの定量化手法」は、IBM の社内システム開発部門である IGA で試用しています。IGA は日本 IBM グループの社内基幹業務を支える約 350 のアプリケーションの開発・保守を行い、日本 IBM グループと海外 IBM の社員、ビジネス・パートナーの業務を支えています。そんな中で、常に先駆者として、新しいプロセスやメソドロジーを開発・保守の現場に適用・実証してきました。この定量化手法もその中の 1 つです。

アプリケーションの保守は、開発とは違い成果が見えにくく、サービス量を測ることが困難でした。このため、対価に見合うサービスを提供していることが説明できないなど、様々な弊害をもたらしてきました。この状況に風穴を開け、サービス量を測れるようにするのがこの定量化手法です。IGA で実証実験を繰り返しながら、論文に記載した「量を増やしている作業に着目する」ことを中心とした手法が形成されてきました。完全な測定方法は存在しませんが、より現実に近い測定を可能とする方法論であると評価しています。

今後も試行を繰り返しながら、保守サービス工数のシミュレーションなど多様な用途に適用範囲を拡大していきたいと考えています。

システム価値向上を目的とした Scrum の試行・評価

中村　伸裕（住友電気工業株式会社／
住友電工情報システム株式会社／大阪大学）
服部　悦子（住友電工情報システム株式会社）
永田　菜生（住友電気工業株式会社）
楠本　真二（大阪大学 大学院情報科学研究科）


中村　伸裕


服部　悦子


永田　菜生


楠本　真二

エンタプライズ・システム開発では利用部門の潜在的ニーズを引き出すことが 1 つの課題となっている。アジャイル手法は解決策の 1 つとして期待されているがその仕組みは明確になっていない。

今回 Scrum の試行を行った結果、複数の開発者が 1 つの機能を設計することで複数の設計案が提案され、設計案の選択の際、利用者の潜在的なニーズを引き出し評価していることがわかった。また CMMI のプロセス領域ごとに評価すると要件管理、技術解、妥当性確認、決定分析と分析、プロジェクト管理の領域で改善効果があることがわかった。さらにインタビューの結果、開発者のモチベーションが高いことがわかり従来の開発手法より付加価値の高いシステムが開発できる状態であることを確認した。

若年技術者向けソフトウェア開発研修プログラムの開発と評価

大森　久美子（NTT サービスイノベーション総合研究所 ソフトウェアイノベーションセンタ）


大森　久美子

若年技術者の設計力及び開発力の底上げを目的としたソフトウェア開発の PBL（Project Based Learning）研修を、5 年間にわたって実施しながら改善し、ソフトウェア開発研修プログラムとして集大成しました。

この研修プログラムは、受講者の気づきに対するフィードバック手法や新しい要求分析の技法、受講者が受発注両方の立場を経験することを特徴としており、評価の結果、受講者自らが問題を発見し、その要因と対策を自発的に考えることができるようになったことが確認できました。

この研修プログラムの内容は 5 冊の書籍として公開しており、複数の大学や研究機関でも活用頂いております。

今後も研修の継続を通して、研修プログラムの定量的評価手法の検討などに取り組んでいきたいと思います。

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 電気ネットワーク部 部長
松本　雅行

社会基盤の一つである鉄道システムは、様々なニーズへの対応と変化する状況への適応が求められている。これら異種性と適応性に対応できる技術をアシュアランス技術と呼び、欠くことのできないものとなっている。列車輸送管理及び列車制御における、段階的なシステム構築オンラインテストなどの適用例について述べる。

1 はじめに

日本の社会は、少子高齢化、グローバル化、景気の低迷などの大きな環境の変化を受けており、社会の構造・組織などにも影響を与えている。これらの影響はシステムにも及んでおり、これまでのシステムには信頼性・安全性だけが求められていたものが、今や社会や利用者からの要求の変化に即座に対応し、その責任を果たし続けることが求められてきている。また、ネットワークの発達に伴い多くのシステムが有機的に接続されるようになっている。このようなシステムでは異種のニーズを持つだけでなく頻繁にニーズが変化しているなど、システムの外的変化及び内的変化の両方に対して対応が求められており、システムの安定稼動が今まで以上に求められていくことになる。システムの社会における責任はより一層重くなっていると言える。

社会基盤の一つである鉄道システムにおいても、異種のニーズへの対応と状況変化への適応が求められている。これら異種性と適応性に対応できる技術をアシュアランス技術と呼び、欠くことのできないものとなっている。

2 アシュアランス技術の歴史

アシュアランス技術研究については、社会情勢の変化する環境の中で、米国において研究が始められた。米国政府が中心となり、大統領令によって CIAO（Critical Infrastructure Assurance Office）といったアシュアランス研究機関が設立されたほか、DIAP（Defense-Wide Information Assurance Pro- gram）計画などのプロジェクトも進行している。こうした背景から、米国、日本及び欧州の研究者が共同で 1996 年に、アシュアランス技術に関する議論の場として、IEEE 主催の国際会議 HASE（High Assurance Systems Engineering Symposium）が設立された。HASE は 1996 年以降毎年開催されており、毎回 30 件以上の発表が行われている。

もともと、HASE は国防総省などが主体となっており、軍事技術関連が中心であることから、北米以外での開催は許可されていなかったが、電子情報通信学会でのアシュアランス研究会などにおける活発な活動が認められ、2002 年の大会 HASE2002 は日本の東京において開催されている。

3 アシュアランス技術とは

3.1 定義

アシュアランス技術とは、システムの安定稼動を保証する技術で、異種性と適応性の２つの要素を考慮したシステムに適用される技術の総称であり、ここでは、この異種性と適応性の２つを合せ持つシステムをアシュアランスシステムと定義する [1][2]。

3.2 異種性

システムは信頼性、安全性をはじめ多くのニーズを満

足させるものとして構成されている。このニーズの要求度合いはシステムによって異なっており、これをニーズレベルと呼ぶ。システムの稼働の保証に求められるものとして、例えば、以下のようなニーズが挙げられる。

- 信頼性
- アベーラビリティ
- フェイルセイフ
- 安全性
- リアルタイム性

システムを設計する際には、これらのニーズごとにニーズレベルを満たす必要があるが、そのレベルはシステムごとにまちまちである。例えば、制御システムにおいては安全性とリアルタイム性が求められているのに対し、情報システムではアベーラビリティや信頼性がより求められる。もちろん、これらのレベルを、すべてのシステムのニーズよりも高く設定すれば、すべてのシステムのニーズを満たすことができるが、それは現実的ではない。異種のニーズレベルを持ったシステムの稼働を保証するため、異種のニーズレベルの共存を許容する能力のことを異種性と呼ぶ。

3.3 適応性

システムの置かれている環境は、そのシステム自身も含めて、常に変化し続けている。この変化には、システムの成長、システムの初期設定、システムの移行、システム変更と改修、システムの異常などがある。システムが置かれている環境は、そのシステム自身も含めて、常に変化し続けている。従って、そのシステムが置かれる環境を事前に仮定し、システム構築時に設計に反映させるといったことは不可能である。

また、たとえば通勤電車での日中の時間帯、朝夕ラッシュ時と終電の時間帯では時間に対するお客様の制約や要望が異なるため、システムに求められる信頼性やリアルタイム性のレベルは時間ごとに変化する。

こういった、システムの状況によるニーズの変化や時間的なニーズの変化に対応するためには、常に柔軟に対応しうるシステムの構築が必要となる。このような状況変化への対応能力を適応性と呼ぶ。

4 鉄道における適用例

4.1 列車輸送管理における適用

列車輸送管理においては、運行制御や設備の管理などの制御システムと、運行情報の伝達や旅客への案内といった情報システムとが共存している。信号機や転てつ機などのデバイスをコントロールする制御系のシステムは、扱う情報量は少ないがそのリアルタイム性、安全性に対する要求は非常に厳しい。一方で、乗客への情報サービスや運行計画の管理などを行う情報系では、制御系に比べて処理の速度は秒単位を求められるものではなく、フォールトトレランス性に対する要求も高くはないものの、データベースなどの大量のデータを処理する必要性があるため、高速かつ広域の情報処理能力が求められている。このように、輸送管理においては情報系と制御系という異種のニーズを持ったシステムの共存、異種性が求められている。

これに対応したシステムが列車の運転を管理するための運行管理システムと呼ばれるシステムが運転線区ごとに作られている。このシステムは、地理的にも広範囲で大規模であり、システム化も長期間を要すため、必然的に段階的な構築が必要となる。また、稼動したシステムは列車運転の性格から 24 時間連続運転システムとなり、システム拡帳時にも稼動システムの安定稼動を保証した無停止拡張が前提条件となり、かつ使用者の新しいニーズ（使用実績を踏まえての機能向上等）を吸収した変化・成長を前提とした構築が必要となっている。つまり、適応性が求められている。

これらの課題を克服するため、自律分散システム技術をベースとしたアシュアランス技術を活用して、首都圏に ATOS（Autonomous Decentralized Transport Operation Control System）と呼ぶシステムを拡大・成長させつつ 20 線区、182 駅に展開してきた [3]。

①システム線区展開・拡大の課題

ATOS をＪＲ東日本の東京圏の鉄道に導入し、線区展開・拡帳する場合、次のような技術的な課題を克服する必要がある。

a. 大駅では、複数線区の列車が運行されているが、線

区のシステム化を一斉に実施することは困難なため、段階的な構築が必要となり、システム化は長期間となる。

b. 線区展開・拡大したシステムは 24 時間連続運転システムとなり、システム拡大において安定稼動を保証して無停止拡張が必要となる。

c. 各線区は相互乗入れのため独立システムにならず、システム構築中には、ネットワーク、コンピュータ上に稼動に必要なデータと未稼動設備を試験するためのテストデータを共存させる必要がある。

d. システム構築には長期間を要するため、使用者の新しいニーズ（折返し運転支援機能の充実等の使用実績を踏まえての機能向上等）を吸収、システムを成長させる必要がある。

②段階的構築の概要

アシュアランス技術は、高いレベルでの信頼度を保証しつつシステムを変化・成長させる技術であり、運行管理システムを大規模線区に展開・拡帳する手段として有効である。ソフトの適切な管理を実施するとともに、稼動中設備を制御するデータと試験のためのデータの混在を可能とすることにより、システム評価・品質の保証とシステムを段階的に無停止拡張することができた。

このデータの混在は、第一段階は駅装置コンピュータで、第二段階は中央ネットワークで、第三段階は中央装置コンピュータで、第四段階は運行管理ネットワークで実現した。

③段階的構築の手法

本システムは、自律分散システムであり、分散配置した駅装置が自律的に動作しながら、中央装置も 1 構成要素として全体統括を分担し、トータルで 1 つのシステムとして稼動できる。中央と駅、駅と駅相互間の通信では、個別診断などを除いてシステム間の直接やりとりをしないようにしている。各装置は、データの内容を表わす『機能コード』を付加したメッセージを宛先指定せずにネットワークへ送出するブロードキャスト伝送を行っている。そして受信側では各装置自身が機能コードによって必要な情報を取捨選択する方式としている。さらに、既に稼動している装置が送出するデータと試験中の装置が送出するデータとを区別する『フラグ』を設け、受信側の装置がアプリケーションレベルでフィルタリングすることが可能となるデータ構造としている [4]。

本システムでは、既に稼動している装置と試験中の装置が存在し、装置及びネットワークに稼動中設備が必要となるデータと試験中の装置が送出するデータとが混在するため、それぞれに支障しない基本的な仕組みが必要となる。このため各装置にはモードを定義し、このモード状態に応じた動作を行うようにアプリケーションレベルで規定することとしている。図 1 に示すように、各装置はネットワークに、前述したモードに対応したフラグを付加したデータを送出し、受信側の装置では機能コード及びフラグにより、自装置のモードと対比して、その装置自身が動作を決定する仕組みである。

4.2 列車制御システムにおける適用

高密度運転線区の列車制御システムにおいては、高輸送力、高安全性、高信頼性が求められる。当初の自動列車制御システム（以下旧 ATC と呼ぶ）は、固定した区間ごとに列車の存在を検知し、そこに列車を進入させるかどうかの判断により、各列車の速度を地上の集中制御装置で求め、それを各列車の車上制御装置に指示していた。これは、ブレーキ性能の一番悪い車両に合わせた区間長とするため、列車運転間隔が長く高密度輸送は難しかった。そこで、各列車が、自らの位置を認識し、地上からは停止する位置のみを列車に伝送する。この停止位置に対応したブレーキパターンに基づいて列車速度の制御を自律的に行う自律分散型列車制御システム（以下 D-ATC と呼ぶ）の開発が必要となった。さらに、新しい列車制御システムに取り替える際には、新旧両システムが共存する中、D-ATC の構築及び試験のためにその旧 ATC を止めることは出来ないため、システム全体の運行を妨げず、D-ATC の段階的投入と、旧 ATC を持つ列車とを共存させながらのオンライン稼動中のテストを保証するアシュアランスシステム構築技術を確立する必要がある。

一般にシステムは単体で動くことはまれで、他の異種システムとの情報の授受や、システムを取り巻く外部環境の変化の影響を受けながら稼働している。アシュアランス技術とは、このように複数のシステムがネットワーク等を介して互いに接続されたとき、それぞれの異なった目的や機能が互いに妨害されることなく、状況に応じて連携し共存できることを保証する技術であり、列車制

御システムにもアシュアランス技術の適用が求められている。D-ATC における個々の課題について、どのようなアシュアランス技術を用いて解決したか、その方策を以下に述べる[5]。

①システムの拡張

ATC の更新においては、全線区を一度に更新するのではなく、幾つかの段階に分けた形で更新を行う。D-ATC の導入を予定している線区は、旧 ATC が設置されている区間であるため、本システムの導入に当たっては、システムの境界箇所において旧 ATC と D-ATC の新旧切換制御が必須となる。旧 ATC 区間から D-ATC 区間に列車が進む時は、有効な電文を受信することによって D-ATC 制御となり、逆の場合は切換区間の電文の「切換」情報を読み取り旧 ATC 制御に移行する（図 2）。

②異種システムとの共存

D-ATC における異種システムとは、同じ列車制御システムでもニーズレベルが異なる旧 ATC と自動列車停止装置（ATS）がある。これらのシステムが工事期間、試験期間また使用開始後も共存できる必要がある。

そこで、D-ATC では次のようなソフト、ハードの対策を施した。

a. D-ATC、旧 ATC 両方の信号が軌道回路に重畳できるように、旧 ATC で使用されていた 2.8 ～ 3.8kHz の周波数を避けて、D-ATC の周波数帯を 11.9 ～ 13.1kHz とした。また、軌道回路割の変更を前提にして、無絶縁軌道回路を採用し、現行の信号波は阻止して D-ATC 波のみ通過できるバイパスボンドを開発し挿入した。

b. D-ATC 電文には「有効 / テスト」情報を持たせて、この情報が「テスト」の場合には車上装置は旧 ATC で制御できるようにした。

c. 車上装置は、D-ATC、旧 ATC 両方の信

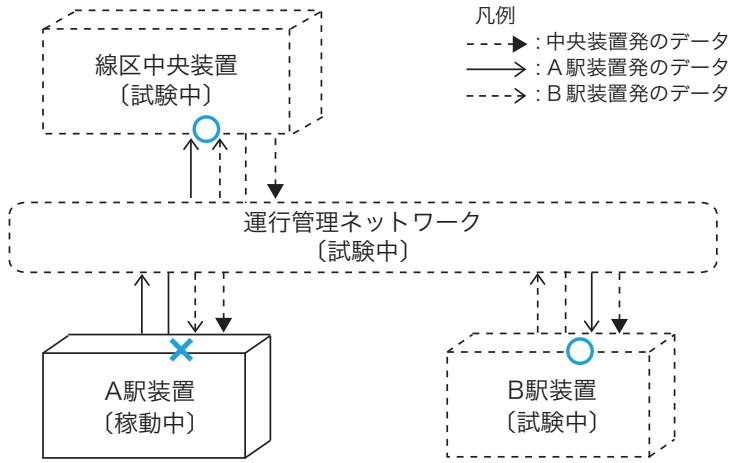


図 1　段階的構築の基本的な処理の概要

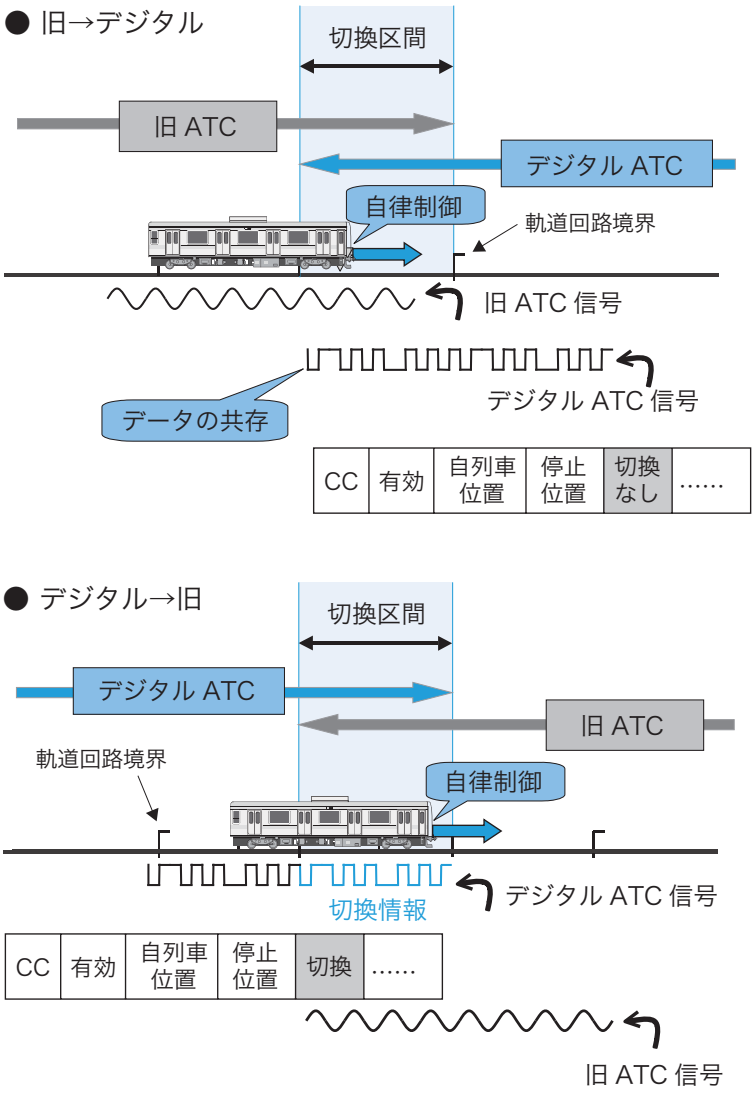


図 2　システム境界箇所における切換制御

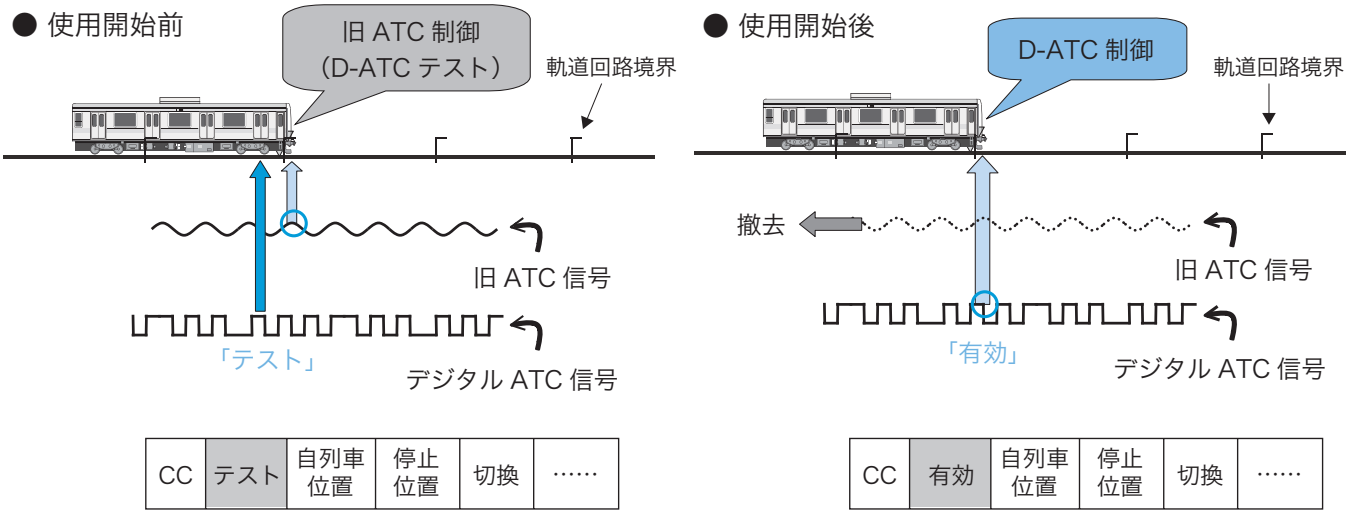


図３　オンラインテストと使用開始切換

号波を受信し、その状況に応じてどちらのモードで制御するか自律性を持たせた（図３）。

③オンラインテスト

新旧システムを共存できるようにしたおかげで、旧 ATC を運転しながらテストを行うことも可能となった。昼間、旧 ATC の信号に重畳させて D-ATC のテストデータを流す。車上システムはテストか有効かのフラグを見て処理するので、テストデータの場合には旧 ATC の信号に従って運転する。つまり、地上のシステムのテストは昼間列車の運転時間に十分なテストを行うことが出来る。また、車上システムのテストも、列車運転は旧 ATC によって行って、D-ATC の電文による確認試験を行うことも可能としている（図 3）。

④短時間でのシステム切換

限られた時間で D-ATC の使用開始切換を行うには、最小限の操作で D-ATC の切換を行うシステムとしなければならない。

具体的には、D-ATC 電文の「有効 / テスト」情報を利用して、この情報を変更するだけで車上装置がいずれのシステムで動作するかが選択できる。これにより、極めて短時間でシステムの切換ができるようにした（図 3）。

⑤列車運転の継続性

D-ATC が設置される区間と、旧 ATC または ATS 区間が設置されている区間をまたがって運行する列車は、その境界でシステムの切換が必要となる。通常は、停車後、地上システムの切換と車上システムの乗務員による手動切換を行うという方法をとっている。そこで、切換区間の D-ATC 電文に「切換情報」を付加し、D-ATC から旧 ATC へはこの情報により自律的に切換を行なう。また、逆の場合は D-ATC 電文を受けることによって D-ATC に自動的に切り換わる。このような構成とすることにより運転の継続性を図ることができた。

5　あとがき

アシュアランス技術の応用例を、鉄道システムについて紹介したが、これ以外にも宇宙応用、医療応用など情報サービスシステムなどの多くの社会インフラシステムに実用化が進んでいる。異種システム共存技術やオンラインテスト技術を活用することにより、システムの拡張やシステムを稼動させながらの保守管理ができるようになる。このアシュアランス技術を用いて鉄道システムの革新を今後ますます進めていきたいと考えている。

【参考文献】

[1] I-Ling Yen、“Toward Integrated Methods for High-Assurance Systems”、IEEE Computer、1998

[2] 森 欣司、「アシュアランスシステムのニーズと技術動向」、電子情報通信学会、アシュアランス研究会、2000

[3] 上条 恵司他、「アシュアランス技術（運行管理システムの大規模展開）」、日本鉄道電気技術協会、2001

[4] 森 欣司、「自律分散システム入門」、森北出版株式会社、2006

[5] 松本 雅行、「新しい列車制御システムの開発とアシュアランス技術」、日本信頼性学会、2000

ディペンダブルシステム構築と運用の技術

独立行政法人科学技術振興機構
ディペンダブル組込み OS 研究
開発センター センター長
屋代　眞

独立行政法人科学技術振興機構
ディペンダブル組込み OS 研究
開発センター 研究員
髙村　博紀

独立行政法人科学技術振興機構
ディペンダブル組込み OS 研究
開発センター 研究員
松原　茂

現在の社会インフラや生活環境を支える情報技術は目覚ましく進歩しているが、一方でシステム障害は無くならず、時に人命や社会に大きな影響を与えている。安全、安心、快適な生活を支えるディペンダブルな情報システムを構築し運用することを目指して科学技術振興機構 CREST※1 研究領域で取り組んできた DEOS プロジェクトの成果を紹介する。

1 はじめに

現在の社会インフラや生活環境を支えているコンピュータシステムは膨大なソフトウェアにより動作している。ソフトウェアはプログラミング言語、開発環境、開発・運用ツール、開発プロセスなどの研究により進歩を遂げてきたが、システムの障害は無くならない。科学技術振興機構はディペンダブルなシステムを構築するためのソフトウェアの基盤技術を研究するために CREST 研究領域「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」（以下、DEOS プロジェクトと略す）を2006年に立ち上げた。DEOS プロジェクトは本年度で終了するが、本稿ではその成果の中から開発・運用プロセスに関する成果を解説する。DEOS プロジェクト成果の詳細は「DEOS プロジェクト研究成果集」[1] を参照されたい。

2 なぜ情報システムの障害は無くならないか？

今日、情報システムなしに生活を送ることは不可能になっている。携帯電話やネット家電によるサービスは言うに及ばず、行政、金融、流通、医療、交通、防衛、エネルギー、通信、放送システムなどいたるところでその恩恵を受けている。現代のシステムはサービス内容が多岐にわたるため、システムの規模は巨大にならざるを得ず、またその構造は複雑なものになっている。以下、今日のディペンダブルシステム構築と運用における課題を要約する。

①ブラックボックスの存在

複雑なシステムではコスト・納期などの制約からすべてをゼロから開発することは稀であり、既存システムを局所的に改造して使用し続けることが多い。その際に COTS※2 やレガシーソフトウェアがブラックボックスとして使われることが多く、そのことがシステムの把握や障害の分析を難しくしている。

②複数の開発組織の連携

システム開発に係る開発者・運用者は複数の企業や国

【脚注】
※1　戦略的創造研究推進事業（Core Research for Evolutional Science and Technology）
※2　Commercial Off-The-Shelf : ここでは既成品として販売やリースされるソフトウェア

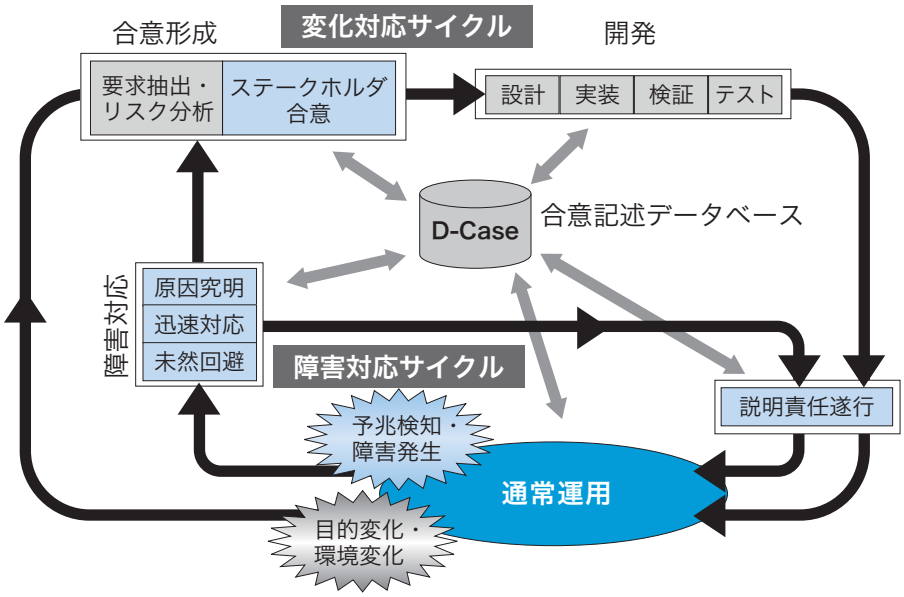


図 1　DEOS プロセス

まとめると、今日の情報システムは変化し続けるオープンシステムであり、①、②、③のような不完全性、④、⑤のような不確実性という排除できない問題がある。オープンシステムをマネージするために必要な能力、すなわち実環境の中で長期的に運用されるシステムがその目的や環境の変化に対応し、システムに関する説明責任遂行を継続的に支援しつつ、利用者が期待するサービスを継続的に提供し続ける能力を、「オープンシステムディペンダビリティ（OSD：Open Systems Dependability）」と名付けた。OSD に関する議論や内容は文献 [2] に詳しく解説されている。

をまたいで存在していることも少なくなく、システムの全容を完全に把握することが難しい。また、サプライチェーンマネジメントの問題も関わってくる。

③開発者と運用者の関係

開発者と運用者の間でお互いの現場や用語の違いの認識が不十分なことが障害に繋がることも多い。最近ではできる限り言葉を定義し共有することが普通になって来てはいるものの、完全にあいまいさを排除することはできない。

④環境の変化

インターネットを通じたサービスの提供、情報の交換、新規プログラムや更新のダウンロードなどの普及により、情報システムが使われる環境はそのライフサイクルの中で常に変化し続けることが多くなり、環境の変化により新たな問題や障害が起こることがある。

⑤要求の変化

長期にわたり継続的にサービスを提供するシステムでは、開発された時のサービスの目的や利用者の要求が、ライフサイクルを通じて変化することも多くなってきた。技術の進展や法規制・国際標準の変更などによる要求の変化もある。要求の変化により新たな問題や障害が起こることがある。

ソフトウェア開発はプログラミング言語、開発・保守プロセス、プロジェクトマネジメント手法、ツールなどの進化により、その開発のスピードや品質など著しい進歩を遂げている。既に一般的になっている CMMI、SysML などのプロセスやツールを正しく活用しても上記の問題点は必ずしも解決していない。IPA/SEC が定期的に纏めている情報システムの障害情報 [3] 等にも見られるように、単なるソフトウェアのバグと言うよりも前述の原因による障害が数多くみられる。巨大で複雑になったシステムをマネージするためには、サービス継続及び説明責任という観点で問題を捉えライフサイクルを通じて一貫した手法や技術が必要である。以下の章では、DEOS プロセスとそれを実現する核となる技術である D-Case を紹介する。

3　DEOS プロセス

社会インフラ、企業や公共の基幹システム、組込み機器のようなシステムは膨大なソフトウェアを含み、長期間使われ、その運用中にシステムは環境変化・目的変化に対応すべく修正されていく。そのためシステムの開発、運用・保守、EOL※3 という一連の流れとしてプロセスを

【脚注】
※3　End Of Life : 製品やサービスの終了

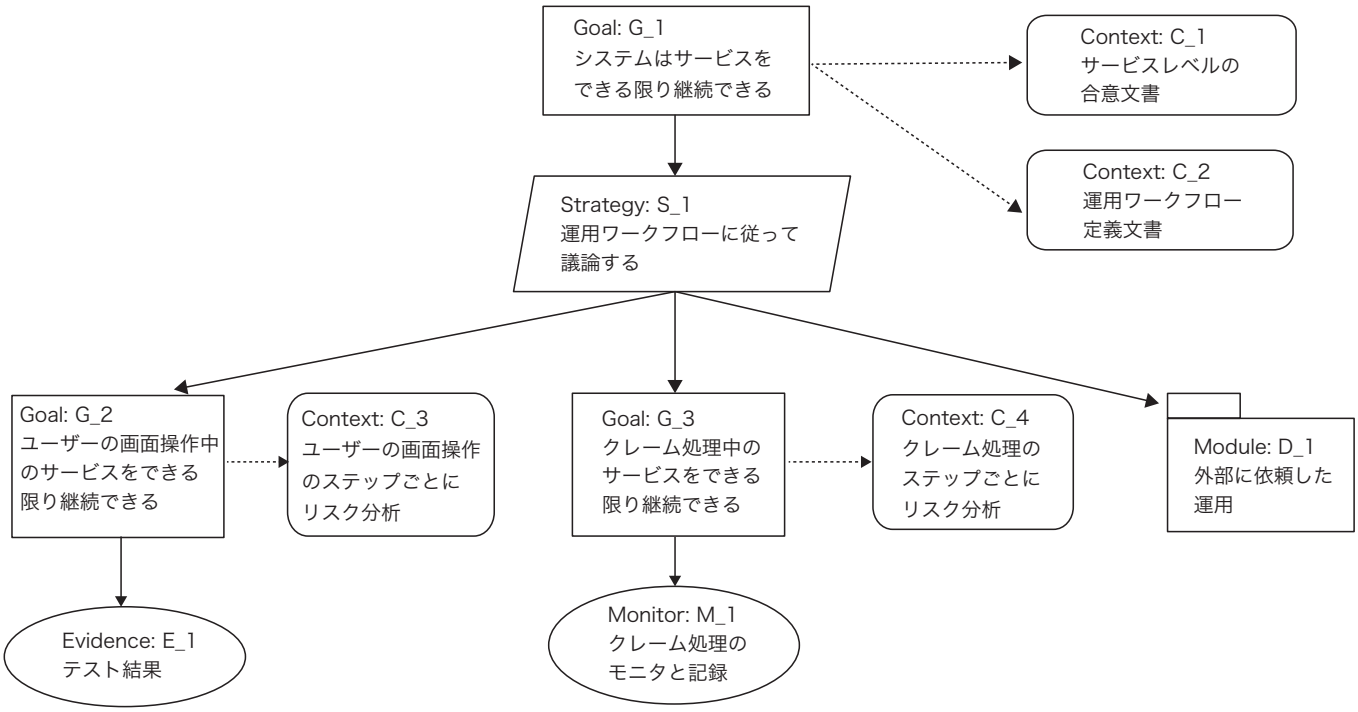


図 2　D-Case の記述例

分けて考えることは難しい。開発と運用・保守をシームレスに連携させ、システムのライフサイクルを通じて一貫したマネジメントの仕組みをもった新たなマネジメントプロセスが必要である。システムを安全・安心・快適に使い続ける、すなわちディペンダブルに開発・運用して行くためには上記で述べたように、システムによるサービスを継続し、開発・運用において実行したことに対して説明責任を果たすことが重要になる。

このような観点から開発・運用のプロセスを見直すと、システムをディペンダブルにするためには；1. 通常運用で何をすべきか、2. 障害発生時に何をすべきか、3. 変化対応で何をすべきか、を定義して実行する必要があることが分かる。多くのシステムや組織ではこれらのプロセスが存在していることが多いが、有機的にシームレスに融合・結合するための仕組みが不十分なことと、これらのプロセスが明示的になっていないことが多いことから、前述したようにシステムに障害が発生し、それによって社会的に甚大な影響を及ぼすような事態に発展することが起こっている。

我々は上記の問題を解決するために DEOS プロセスを提唱した（図 1）。DEOS プロセスは障害に対して「未然防止機能」や「障害対応機能」を含み、システムの変更を起動するための「再発防止機能」も統合した反復的プロセスである。システムは「変化対応サイクル」の中で合意形成に基づいて開発・運用され目的や環境の変化に対応する。通常運用では合意に基づいてシステムの振る舞いや環境の変化を監視する。障害や予兆が検知された場合は合意に基づきシステムの自己回復機能あるいはオペレータの動作により「障害対応サイクル」に入る。この過程で新たな合意形成が必要と判断されると再び「変化対応サイクル」に入って合意を見直し、システムを変更する。これら一連の作業は後述する D-Case に基いて指示され記録として残されると共に、説明責任を果たすためのベースとなる。DEOS プロセスは利害関係者間の合意形成、説明責任の遂行を含むシームレスなプロセスである。DEOS プロジェクトではそのプロセスを支える技術・ツールを整備した。

4 DEOS プロセスを実現する技術 ― D-Case と事例 ―

D-Case は利害関係者の合意形成のための手法やツールの総称である。アシュアランスケースを基に、システムをマネージする仕組みとしてモニターノード、システ

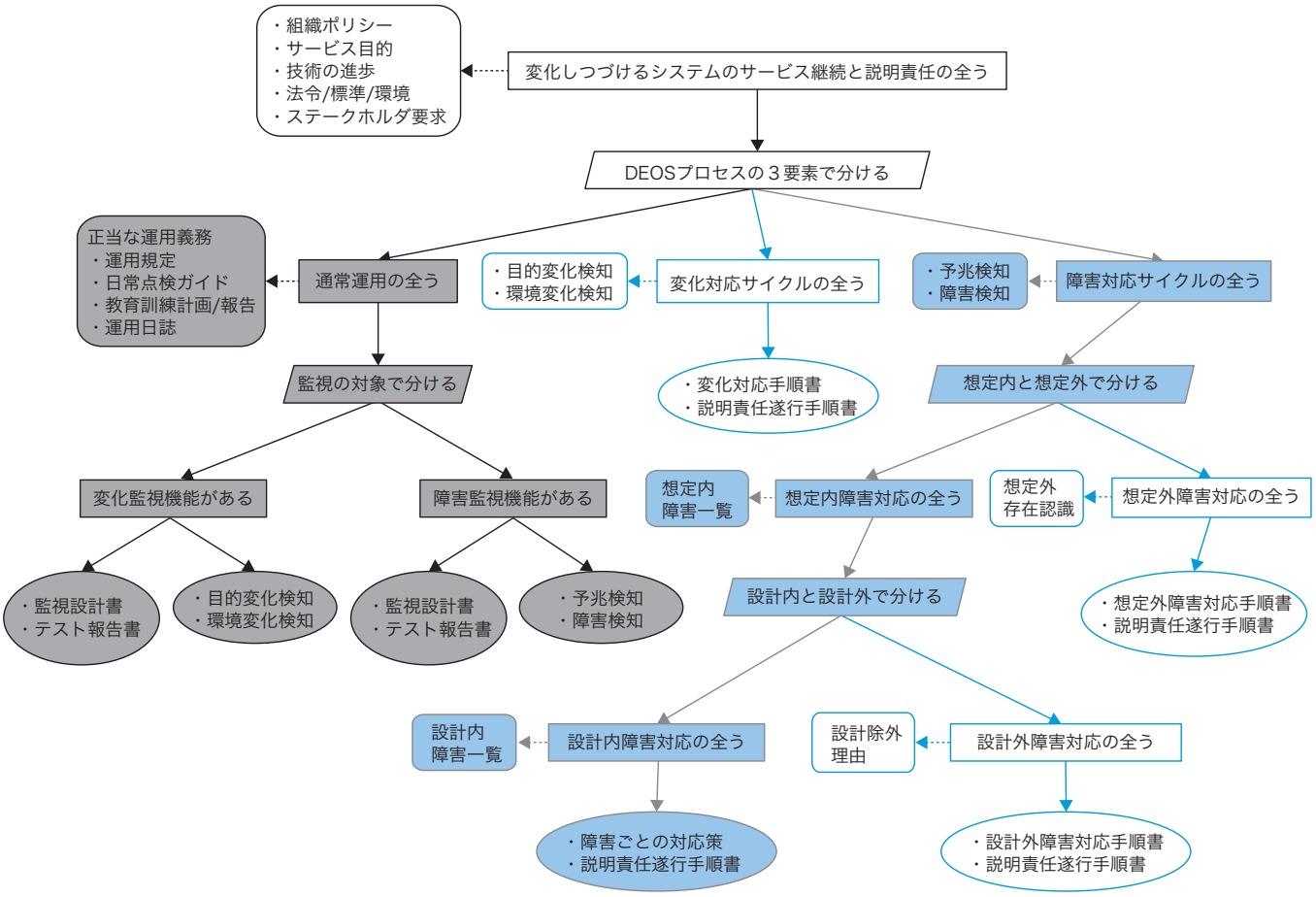


図３　DEOS プロセス基本部分の D-Case による記述

ムの動作を変更させるアクションノード、他のシステムとの連携に関する事項を記述するための外部接続ノードを追加して拡張したものである [4]。アシュアランスケースは車載システムの機能安全の国際規格 ISO 26262 でも言及されている安全ケースのもとになるもので、確信を得るための証拠が提示され構造化された議論をまとめたドキュメント群のことである。アシュアランスケースは国際規格 ISO/IEC 15026 シリーズとして制定されている。

D-Case（図 2）ではその表現に GSN※4 と呼ばれるグラフィカルな手法を採用し、まず主張したい事柄をトップゴールに掲げ、何が前提となっているのかをコンテクスト（前提）ノードに書き、どのような戦略により主張を分解していくのかを戦略ノードに記述しながら議論を構造的に詳細化していく。これによりトップゴールはサブゴールに分解され、サブゴールを保証する証拠に議論が達するまでこの分解は続けられる。D-Case では、システムの運用に関しても証拠を残さなければならないので、運用中のシステムの挙動にかかわるデータ（モニターによる測定値やログなど）も含まれる。これら運用中のシステムにかかわるデータが設定範囲を越えた場合は、通常運用から逸脱が起きていると判断し、設定範囲に収まるように障害対応を行なわなければならない。当プロジェクトではシステムにかかわるデータの収集とその障害対応は D-Script とよばれるスクリプトにより実現している。また、障害の原因究明の結果再発防止策やシステム改善が必要となり変化対応サイクルに進む場合は、再合意をおこない D-Case を更新して、システムをマネージする。このようなプロセスを確実に遂行し説明責任を果たせるようにするためには、D-Case の履歴を各種開発・運用ドキュメント群とリンクさせる仕組みが必要で

【脚注】

※４　アシュアランスケースを記述するための表記法の一つ：Goal Structuring Notation

ある。我々はこのために合意記述データベース（D-ADD）を開発した。前章で述べたDEOSプロセスはD-Caseで表現して、基本的なD-Caseとして活用できる（図3）。

DEOSプロセスを実現するためには、各種ドキュメント群をマネージ出来るようにD-Caseを記述すること、システムを柔軟に制御してディペンダビリティを達成するための記述がD-Caseに明示されていることが重要となる。この中には責任者、担当者、有効期限などについても記述される。さらに、我々はシステムの監視、異常発生時のシステムの柔軟な対応を可能とする実行環境として、D-RE※5を開発し、D-Caseとの連携に関する研究を進めてきた。D-Caseはサービス継続のための必要事項の合意文書であるとともに、説明責任を果たす際に重要な役割を果たすことからもDEOSプロセスの核であるといえる。D-Caseを記述することにより、利害関係者間の合意事項やリスクコミュニケーションが明示化され、サービス継続と説明責任の方針が明確になる。

DEOSプロセスを円滑に運用するためには、D-Caseを記述し、さらに対象システムをマネージするためにD-Script、D-RE、D-ADDとの連携が必要となる。そのためのツールとしてD-Case記述ツールを開発し、さらにCMIS※6をインターフェスとして開発プロセスで作成される文書をD-Caseと連携させる機能、OSLC※7を通じてSysMLツールとの連携を可能にする機能を開発し、既存のプロセスや手法との統合も図っている。

D-Caseを使った実証実験の一つとして、日本科学未来館のフロア案内ロボットの開発運用に適用した。このロボットはART-Linux※8上に各種センサや機能を実装し、30 × 130mの展示会場をロボットが人や障害物を避けながら巡回し、来訪者との対話、デモの時刻と内容の宣伝を行うことなどを目的としている。この実証実験はロボットの機能、運用、安全、説明責任、改善の議論をD-Caseとして記述し、その内容をステークホルダ（サービス提供者）の日本科学未来館と合意し、一日当たり1万人強の来訪者が来る環境においてロボットを運用し、求められたサービスを安全に提供する事ができた[5]。

D-Case事例や実証評価活動はD-Case実証評価研究会ホームページに紹介されている[6]。また、DEOSセンターホームページでもD-Caseツールや事例を公開している[7]。

5 技術の実用化と今後の課題

DEOSプロジェクトにおいて研究開発されたソフトウェアはDEOSセンターホームページからダウンロード可能となっている。また、我々は開発してきた技術や概念が広く使われるために標準化活動をおこなっている。デジュール規格としてIEC TC56においてプロジェクトが開始され国際標準IEC 62853（OSD規格）の策定を進めている。また、デファクト規格としては2013年7月にThe Open Groupで当プロジェクトの考え方を反映した技術標準が策定された[8]。

DEOSプロジェクトにおける研究開発成果は成果集や書籍、論文、ソフトウェア等により広く利用可能な形で発表されている。これらの技術を実際に産業界で使っていくためにコンソーシアム「DEOS協会」※9が2013年10月に設立され、今後はコンソーシアムを中心にプロジェクトの成果が展開されていく[9]。

6 謝辞

DEOSプロジェクトは研究総括の所眞理雄氏を始め、多くの研究機関や企業から多数の方々に参加頂き成果を上げることができた。ここに謝意を表す。

【脚注】

※5 DEOS Runtime Environment: DEOS実行環境
※6 標準化団体OASISの定義するコンテンツ管理システムのインターフェス標準
※7 Open Services for Lifecycle Collaboration：開発ツールなどの相互連携のためのデータ交換の標準仕様
※8 （独）産業技術総合研究所で開発されたReal-time Linux
※9 正式名称は「一般社団法人 ディペンダビリティ技術推進協会」

【参考文献】

[1] 所 眞理雄、他："DEOSプロジェクト研究成果集"、科学技術振興機構 DEOS-FY2013-SS-01J、2013/11/15.
[2] Mario Tokoro (eds): Open Systems Dependability, CRC press, 2012.
[3] 松田 晃一、他：連載 情報システムの事故データ、SEC journal No.26, 27, 28, 30, 32, 34
[4] 松野 裕、山本 修一郎：実践D-Case, オンデマンド出版、2012年
[5] デジタルヒューマン工学研究センター：D-Caseのロボット応用～日本科学未来館フロア移動ロボットを題材にして～ DEOS-FY2013-RA-01J
[6] D-Case実証評価研究会 HP：http://www.dcase.jp/index.html
[7] DEOSセンター HP：http://www.dependable-os.net
[8] The Open Group, Dependability through Assuredness™ Standard, 2013
[9] 一般社団法人 ディペンダビリティ技術推進協会 HP：http://deos.or.jp/index-j.html

オープントレーサビリティツールプラットフォーム TERAS

キャッツ株式会社 グループマネジャ
一般社団法人 TERAS プロジェクトマネジャ
宮本　貴之

キャッツ株式会社 取締役 副社長
一般社団法人 TERAS 開発委員長
渡辺　政彦

名古屋大学大学院情報科学研究科情報システム学 教授
一般社団法人 TERAS 技術委員長
高田　広章

学校法人専門学校 HAL 東京 校長
一般社団法人 TERAS 理事長
鶴保　征城

オープントレーサビリティツールプラットフォーム TERAS※1 は、2013 年 6 月にバージョン 2 がリリースされた。2014 年にバージョン 3 のリリースに向けて開発中である。トレーサビリティとは何かをコモンクライテリアを例に具体的に示し、その後、TERAS のアーキテクチャ、機能、そして課題について紹介する。

1 はじめに

近年、安全・安心な社会に向けて機能安全やセキュリティが重要なテーマである。機能安全規格やセキュリティ評価基準ではソフトウェアの「トレーサビリティ」を要求する。ソフトウェアの「トレーサビリティ」とは、『ソフトウェア開発の成果物である文書間において追跡が可能である』ことである。良好な安全性や高いセキュリティの指標として「トレーサビリティ」を示すことが説明責任を果たすことになる。

多種多様なツールから生成される多種多様なソフトウェアの成果物間のトレーサビリティにオープンなプラットフォームを提供するのが TERAS である。

2 トレーサビリティとは

前述したようにソフトウェア開発におけるトレーサビリティとは、『ソフトウェア開発の成果物である文書間において追跡が可能である』ことである。具体的にソフトウェア開発におけるトレーサビリティを示すために、セキュリティ評価基準や機能安全規格にあるトレーサビリティを解説する。

情報技術セキュリティ評価基準 ISO/IEC15408（コモンクライテリア）[1] には、保証クラス、ファミリ、コンポーネント、EAL（Evaluation Assurance Level）がある。

保証要件の最も抽象的なセットはクラスと呼ばれる。各クラスには、保証ファミリが含まれ、保証ファミリには、保証コンポーネントが含まれ、保証コンポーネントには保証エレメントが含まれる。クラスとファミリは、保証要件を分類するために使われ、コンポーネントは PP（Protection Profile）/ST（Security Target）に保証要件を特定するために使われる。コモンクライテリアでトレーサビリティは ADV（開発）クラスの RCR（Representation CoRrespondence）ファミリに属する概念となる。コンポーネントと EAL との関係を表 1 に示す。

表 1　ISO/IEC15408 開発者向けセキュリティ評価に関する要件

保証クラス	保証ファミリ	評価保証レベルに基づく保証コンポーネント						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
開発	ADV_FSP	1	1	1	2	3	3	4
	ADV_HLD		1	2	2	3	4	5
	ADV_IMP				1	2	3	3
	ADV_INT					1	2	3
	ADV_LLD				1	1	2	2
	ADV_RCR	1	1	1	1	2	2	3
	ADV_SPM				1	3	3	3

【脚注】

※1　Tool Environment for Reliable and Accountable Software の略で、2011 年 4 月に設立された一般社団法人 TERAS が提供するオープントレーサビリティツールプラットフォームである

表２　トレーサビリティに関するアクションエレメント

レベル	開発者アクションエレメント	証拠の内容・提示エレメント	評価者アクションエレメント
ADV_RCR.1 非形式的対応の実証	開発者は、提供する TSF 表現の隣接するすべての組の間の対応の分析を提供しなければならない。	提供された TSF 表現の隣接する各々の組に対し、分析は、より抽象度の高い TSF 表現のすべての関連するセキュリティ機能性が、抽象度の低い TSF 表現に、正確かつ完全に詳細化されていることを実証しなければならない。	評価者は、提供された情報が、証拠の内容・提示に対するすべての要件を満たしていることを確認しなければならない。
ADV_RCR.2 準形式的対応の実証	同上	＋ 提供された TSF 表現の隣接する各々の組に対し、どちらの表現も最低限、準形式的である部分に対しは、表現のそれらの部分の間の対応の実証は、準形式的でなければならない。	同上
ADV_RCR.3 形式的対応の実証	＋ 対応する表現がともに形式的である部分については、開発者は、対応を証明しなければならない。	＋ 提供された TSF 表現の隣接する各々の組に対し、どちらの表現も形式的である部分で、それらの部分の間での対応の証明は、形式的でなければならない。	評価者は、形式的な分析を選択的に検証することによって、対応の証明の正確さを決定しなければならない。

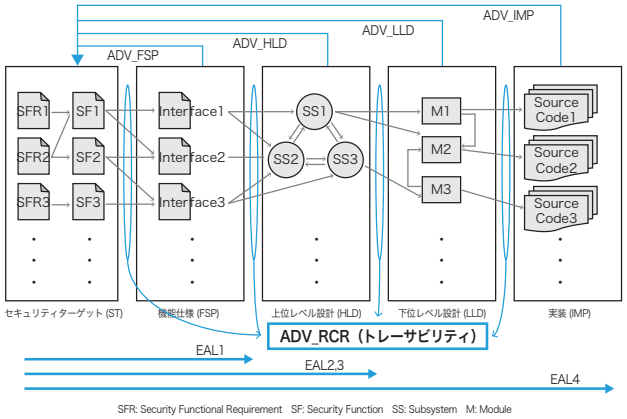


図１　セキュリティとトレーサビリティ

保証ファミリ※2は、レベル付けが行われ、コンポーネントにレベルを付ける方法の根拠が示される。この根拠は、適用範囲、深さ、及び／または厳格性の観点からである。トレーサビリティに関係する ADV_RCR は３レベルである（表 2）。

開発者のアクションは、レベル１では非形式的で、レベル２では準形式的で、レベル３では形式的な TSF※3 に関する成果物間の対応関係を提供しなければならない。ADV_RCR と保証ファミリおよび EAL の関係を図１に示す [2]。

機能安全規格である ISO20262、車載ソフトウェア開発プロセスモデルである AutomotiveSPICE、IEEE Standard for Software Verification and Validation、そして IPA/SEC の ESPR（Embedded Software Process Reference）におけるトレーサビリティに関する記述を表３に示す。

表３　規格・プロセスとトレーサビリティ

文書	記述
ISO 26262：2011(E) Functional safety Part2 Management of functional safety Annex B:Examples for evaluating a safety culture	- 貧弱な安全文化の指標例：説明責任（アカウンタビリティ）がトレーサブルではない。 - 良好な安全文化の指標例：機能安全に関わる意思決定の責任がトレーサブルであることを保証するプロセスである。
AutomotiveSPICE ENG.4 ソフトウェア要件分析　Level 1	参照元の要件とソフトウエア要件との間でトレーサビリティを作成しているか。
IEEE Standard for Software Verification and Validation（IEEE Std 1012-2004）5.4.1 アクティビティ：コンセプト V&V（プロセス：開発）	・獲得要求とシステム要件とのトレーサビリティを検証する。 ・システム要件とソフトウエア要求とのトレーサビリティを開始する。
ESPR SYP2 システム・アーキテクチャ設計 SYP2.2 システム・アーキテクチャ設計の確認 2.2.1 システム・アーキテクチャ設計書の内部確認	・システムを構成する機能ブロックの分割が適切であり、システム要求で求められる事項が現実可能かどうか（トレーサビリティの確認）。 ・システム要求やテスト仕様との対応（トレーサビリティ）が取れているか。

【脚注】

※２　ADV クラスの保証ファミリ構成：
ADV_FSP：機能仕様に関する要件
ADV_HLD：上位レベル設計に関する要件
ADV_LLD：下位レベル設計に関する要件
ADV_IMP：実装表現（ソースコード）に関する要件
ADV_RCR：追跡性（トレーサビリティ）に関する要件
ADV_SPM：セキュリティ方針モデリングに関する要件
ADV_INT：TSF 内部構造に関する要件

※３　TSF：TOE（Target Of Evaluation）Security Functionality の略で、セキュリティ機能要件（SFR）が正しき動作するために必要なすべてのサブシステムやモジュールである

3 TERAS とは

トレーサビリティ対象となるソフトウェアの成果物は多種多様な文書から構成される。「要求定義書」、「機能仕様書」、「基本設計書」、「詳細設計書」、「ソースコード」、「テスト仕様書」、「テストケース」、そして「テスト成績書」などがある。さらに、「要求定義書」、「機能仕様書」、「基本設計書」、「詳細設計書」などでは、「ユースケース図」、「クラス図」、「シーケンス図」、「状態遷移図表」、そして「ブロック図」などのモデルが成果物の一部になる。

多種多様なオーサリングツールから生成される多種多様なソフトウェアの成果物間のトレーサビリティにオープンなプラットフォームを提供するのが TERAS である。

3.1. TERAS アーキテクチャ

TERAS はオーサリングツールとトレースリポジトリを分離した構造をもつ（図 2）。

分離することで、ソフトウェア開発における成果物の表現を損なうことなく、トレーサビリティを管理することができる。

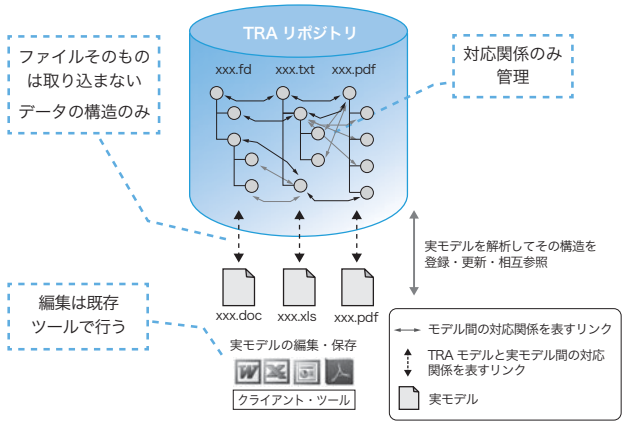


図 2　TERAS トレースリポジトリ

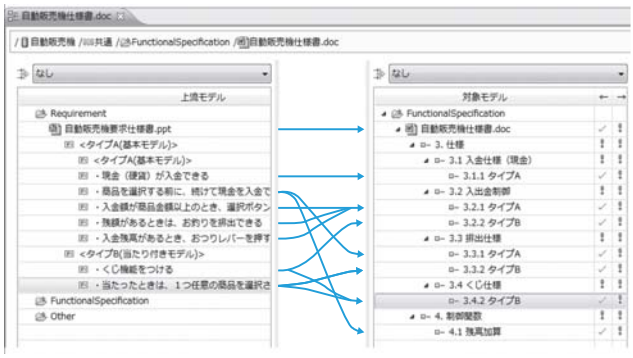


図 3　リンクエディタ

ソフトウェア開発の成果物は膨大な量であり、多種多様な文書から構成される。さらにソフトウェア開発は差分／派生開発が多く、複数のバージョンが存在する。こうしたソースコードを含めた成果物の構成管理に構成管理ツール Subversion が広く使われている。こうした従来の既存環境をそのまま活用できるように TERAS では OSLC（Open Services for Lifecycle Collaboration）に対応する（図 4）。

OSLC は、開発ツールの相互運用性を向上するという共通の目標を持つ企業・組織・個人から成るオープンコミュニティである。OSLC では、共通の REST（Representational State Transfer）プロトコルと共通の開発ライフサイクルデータの表現を定義し、異なるツールを相互運用するための仕様を定義している。ソフトウェア開発における開発ライフサイクルを支えるツールの例としては、要件管理ツール、構成管理ツール、そして課題管理ツールなどが含まれる。

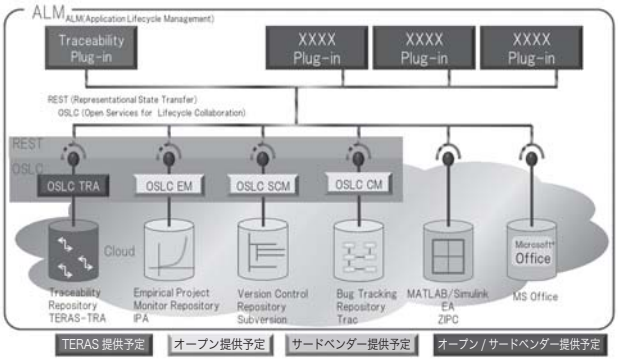


図 4　TERAS プラットフォームアーキテクチャ

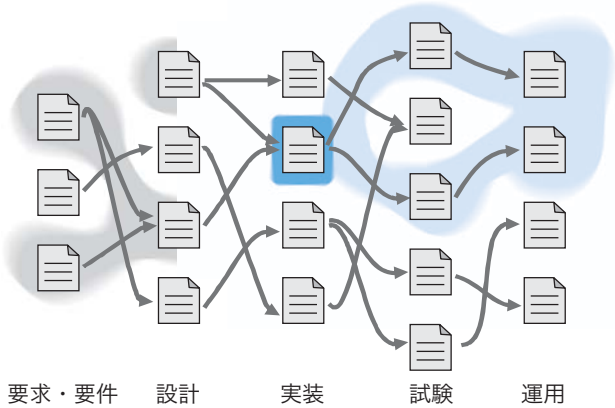


図 5　影響範囲検索

3.2. TERAS 機能

ここからは、TERAS の機能を紹介する。

(1) リンクエディタ

TERAS は、Word や Excel 等のファイルそのものを取り込まず、データ構造のみを取り込み、トレーサビリティを管理する（図 2）。リンクエディタは、取り込んだデータ構造のリンクを編集する（図 3）。

バージョン 3 では、Word、Excel、PowerPoint、PDF、MATLAB/Simulink、Enterprise Architect 、ZIPC、 テ キストファイルであれば、ファイルの内部構造を抽出可能である。その他の成果物は、ファイル単位で管理可能である（図 3）。

(2) 影響範囲検索

派生製品の開発や不具合対策時の大きな関心は修正の影響範囲である。あるモジュールを修正した際に影響を受ける範囲はどこまでで、影響範囲に対してどのような修正を行い、どのような試験を行えば良いか。ある不具合によって見直しが必要な製品はどれなのか。TERAS を活用することで、影響範囲をしっかり管理でき、影響範囲検索機能でわかりやすく確認できる（図 5）。

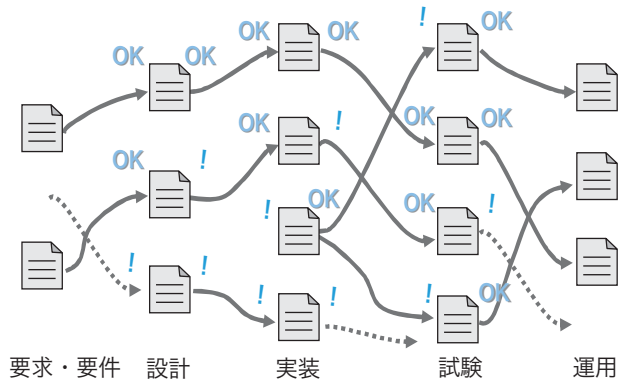


図 6　カバレッジ確認

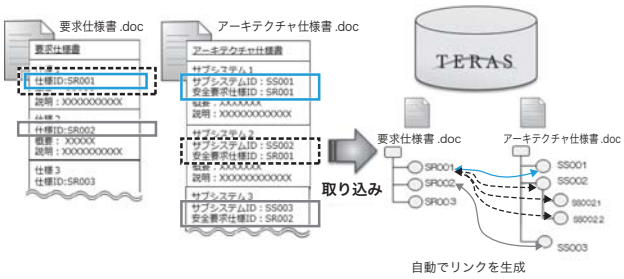


図 7　タグベースリンク

(3) カバレッジ確認

開発プロセスで作成される複数の成果物間の対応関係にモレ・ヌケがないことを確認するのがカバレッジ確認である。

例えば、要件から仕様、設計、実装にトレーサビリティを作成してあれば、要件に対してモレ・ヌケなく検討・実現されているかどうかの確認が容易にできる（図 6）。

(4) タグベースリンク（自動リンク）

トレーサビリティ対象文書にあらかじめ “ トレースタグ（ID）” が記載されていれば、そのトレースタグを解析し、対応する項目間を自動でリンクする機能である（図 7）。

(5) バージョン管理ツール連携（Subversion）

TERAS はトレーサビリティ管理機能とバージョン管理ツール（Subversion）を連携させ、バージョン管理されたファイルでトレーサビリティ管理ができる。

日々の細かな試行錯誤中はバージョン管理ツールだけで成果物を管理し、ある程度成果物の内容が固まってきた段階（例えば、レビュー時やリリース毎）で、TERAS を活用してトレーサビリティ情報を管理する（図 8）。

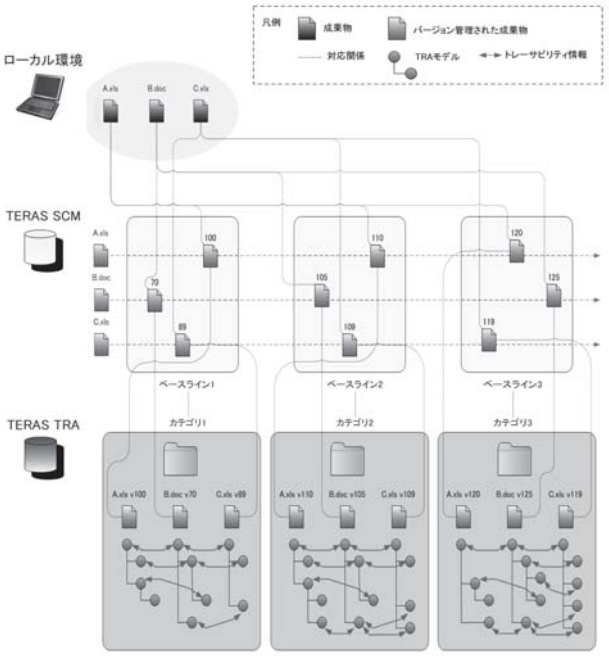


図 8 バージョン管理ツール連携

(6) Trac、Redmine 連携

バージョン 3 では、新たに Trac、Redmine と連携する。これらのツールと連携することで、Trac、Redmine が管理しているチケットの情報（タスク、要求、バグ等）をトレーサビリティ対象として取り込むことができるようになる。

TERAS のカバレッジ確認結果や影響範囲検索の結果は、実施すべきタスクや変更要求となるため、チケットとして起票して管理しておくとモレ・ヌケなく作業が実施できるようになる。この支援機能として、TERAS からチケットを起票できる機能も用意している（図 9）。

4 TERAS 課題

バリエーション管理とトレーサビリティリンクに関する課題がある。トレーサビリティリンクとは成果物間の対応を示すものである。しかしながら、現在のトレーサビリティリンクではリンクの AND 条件と OR 条件がないため、バリエーションを考慮した場合にトレーサビリティリンクが十分条件か必要条件かを明らかにすることができない課題がある。

具体的な例を図 10 に示す。図 10 の左側のシステム 1 における要件と設計の関係において、要件 A を満足するために、設計 P と Q を採用している。また、設計 Q は、要件 B に依存している。図 10 の右側のシステム 2 では、システム 1 から、要件 B が要件 C に代わったため、設計 Q の代わりに設計 R を採用している。自動車を例にすると、要件 A は自動変速（AT）で共通であるが、要件 B は前輪駆動（FF）で、要件 C は後輪駆動（FR）となる。

システム 1 とシステム 2 の要件と設計の関係を単純に重ね合わせると、要件 A を満足するために、設計 P、Q、R をどの組み合わせで採用すればよいかがわからない（図 11）。

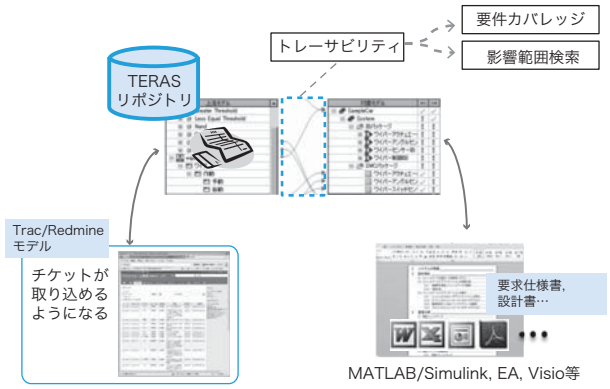


図 9　Trac、Redmine 連携

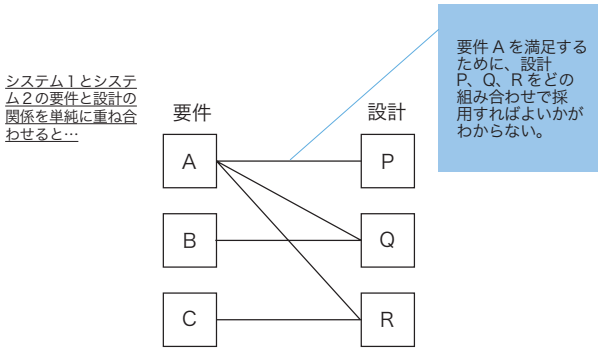


図 11　バリエーションとトレーサビリティリンク課題（2）

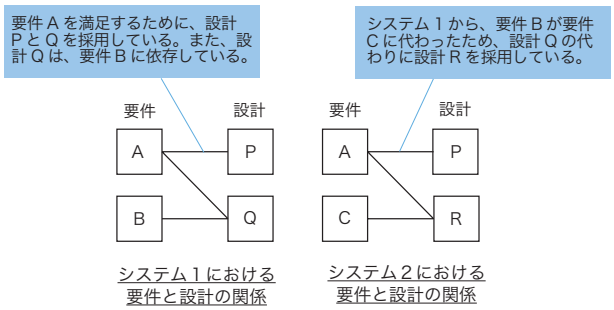


図 10　バリエーションとトレーサビリティリンク課題（1）

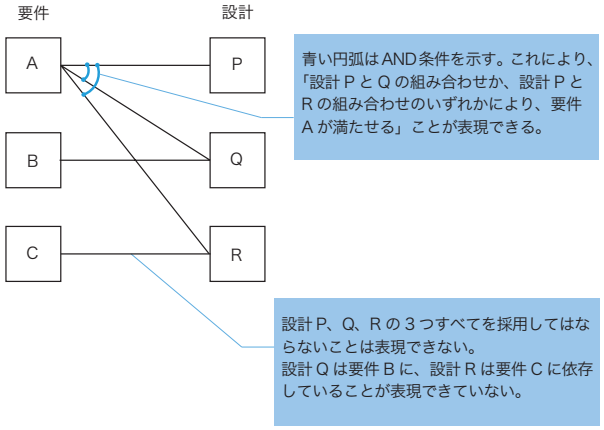


図 12　バリエーションとトレーサビリティリンク課題（3）

そこで、図 12 に示す円弧の AND 条件記号をトレーサビリティリンクに付与する。これにより、「設計 P と Q の組み合わせか、設計 P と R の組み合わせのいずれかにより、要件 A が満たせる」ことが表現できる。しかしながら、設計 P、Q、R の 3 つすべてを採用してはならないことは表現できていない。設計 Q は要件 B に、設計 R は要件 C に依存していることが表現できていない。

このようにシステム 1 とシステム 2 の要件と設計の関係を同時に管理する場合にどうするかといった点が課題である。

バリエーション管理をトレーサビリティリンク上で表現するのではなく、フィーチャモデル上で表現する、または、成果物側に C 言語の #if のように表現し、トレーサビリティエディタがこのようなバリエーション情報を読み取り、トレーサビリティリンクを表示する方法も考えられる。

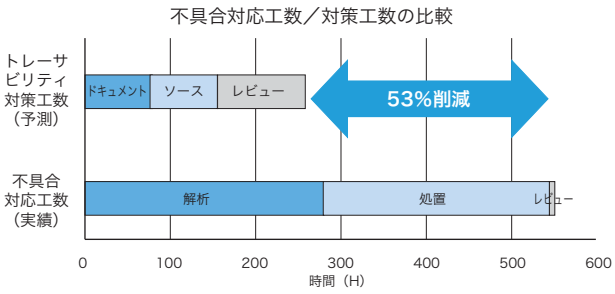


図 13　トレーサビリティによる工数削減効果

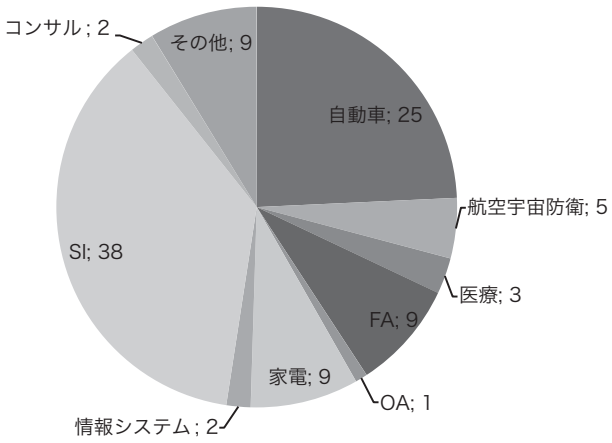


図 14　TERAS 実証評価企業分類

5 ロードマップ・活動報告

また、ソフトウェア開発におけるコスト問題対策でも「トレーサビリティ」が期待されている。エンタプライズ系ソフトウェアでは、新規開発は約 26% に対し、差分／派生開発は約 73% である [3]。組込み系ソフトウェアでは、新規開発は約 42% に対し、差分／派生開発は約 54% である [3]。差分／派生開発では仕様変更の正確な影響分析ができないと、品質、コスト、そして納期が悪化する。正確な影響分析を行うためには「トレーサビリティ」が重要である。

以下の 2 つのソフトウェア開発における不具合データを調査対象としたトレーサビリティ確保におけるソフト開発データからの効果検証が行われた [4]。

- 一般組込み機器製品に搭載される通信ソフトウェア
- 車両搭載用通信プロトコルスタックソフトウェア

不具合対応工数／対策工数の比較では 10% から 53% の工数削減効果があると報告されている（図 13）。

一般社団法人 TERAS では、オープントレーサビリティツールプラットフォーム TERAS バージョン 2 の実証評価の参加を呼び掛けています。2013 年 12 月時点での参加企業は 103 で、図 14 のような分類となっています。

6 終わりに

2011 年から開発を開始し、実証評価会員からの要望を取り入れながら進めてきたオープントレーサビリティツールプラットフォーム TERAS のバージョン 3 を、2014 年 4 月にリリースする。

また、2014 年 3 月 12 日に平成 25 年度 TERAS 成果報告会が開催される予定である。

【参考文献】

[1]「情報技術セキュリティ評価のためのコモンクライテリアパート 3：セキュリティ保証要件」平成 17 年 12 月　翻訳第 1.0 版 IPA/ セキュリティセンター

[2]「日立認証局システム Enterprise Certificate Server Set による ISO/IEC 15408 認証取得について」2004 年 10 月 29 日　株式会社日立製作所 ソフトウェア事業部 セキュリティ対応センタ 栗田 博司

[3]「ソフトウェア産業の実態把握に関する調査　調査報告書―速報版―」2012 年 4 月 27 日　IPA/SEC

[4]「トレーサビリティ確保におけるソフト開発データからの効果検証実施報告書」2013 年 2 月　IPA/SEC

制御システムセキュリティへの対応

独立行政法人情報処理推進機構（IPA）
情報セキュリティ技術ラボラトリー 研究員
入澤　康紀

産業分野や重要インフラ分野などに用いられる制御システムのセキュリティへの対応の重要性を、脅威の高まりと、米国をはじめとした国際動向を背景に解説。制御システムのセキュリティに関する国際基準である IEC62443 の構成と内容、独立行政法人情報処理推進機構（IPA）の提案・推進する本基準を用いた認証制度の確立に向けたパイロットプロジェクトについて解説。

1　はじめに

1.1　制御システムの概要

制御システムは、生産工程やプロセスの制御の自動化など、様々な用途で工数の低減や生産性の向上を目的に利用されている。

最近の制御システムは、情報系のシステムとはファイアウォールなどで分離されている。制御システムのエリアでは、下記の図 1 に示すように、アプリケーションや管理システムなどが動作する上位のレイヤに Windows や UNIX 系の汎用のサーバやパソコンなどで構成されており、標準プロトコルが利用されている。実際の制御にかかわるコントローラやセンサーなどの下層部分は独自のプロトコルやハードウェア、OS などが利用される割合が高く固有の仕様により構成されている。

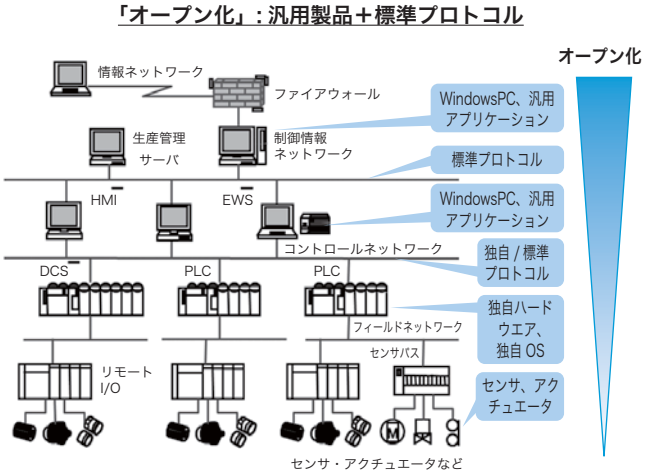


図 1　オープン化が進む制御システムの構成※ 1

1.2　制御システムセキュリティの課題

従来、制御システムは事業者ごとに固有の仕様部分が多く、詳細な内部仕様などを把握しない限り、外部からの攻撃は難しいものと考えられていた。しかし、近年、汎用プラットフォームや標準プロトコルが採用され、更にメンテナンスや管理の目的で外部ネットワークに接続されるなど、事業者及びシステム開発企業の利便性向上やコスト低減が図られている反面で、攻撃対象になりやすいという課題に直面している。さらに、攻撃の糸口となり得る産業用制御システムのソフトウェアの脆弱性の報告件数は大幅に増加している（図 2）。

加えて、これらの産業用制御システムに用いられるソフトウェアの脆弱性については、その特性から、他の一般のソフトウェアに比べて深刻度レベル※ 2（CVSS※ 3 による分類）の高い脆弱性が多くを占めているという特徴があり、対応の必要性が高いと考えられる（図 3）。

1.3　制御システムセキュリティの動向と国内の取り組み

エネルギー、水道、生産ライン、化学プラント、輸送・通信など、重要インフラの制御システムに影響を

【脚注】
※ 1　計測展 2011 TOKYO テクニカルセミナー資料 http://www.ipa.go.jp/security/vuln/documents/TechnicalSeminar2011.pdf
※ 2　CVSS を用いた、脆弱性の深刻度を同一の基準の下で定量的に比較。CVSS については次脚注参照。
※ 3　共通脆弱性評価システム CVSS(Common Vulnerability Scoring System) は、情報システムの脆弱性に対するオープンで包括的、汎用的な評価手法を提供している。
共通脆弱性評価システム CVSS 概説
http://www.ipa.go.jp/security/vuln/CVSS.html

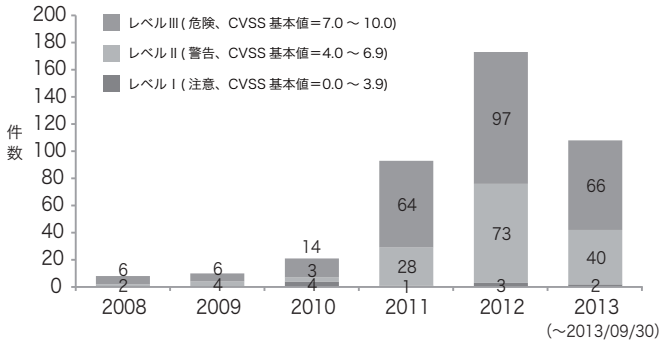


図２　産業用制御システムに関するソフトウェアの脆弱性の深刻度別件数

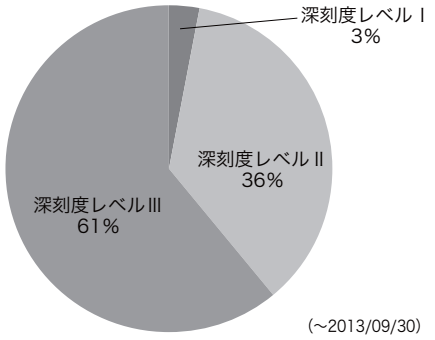


図３　産業用制御システムに用いられるソフトウェアの脆弱性の深刻度別の割合

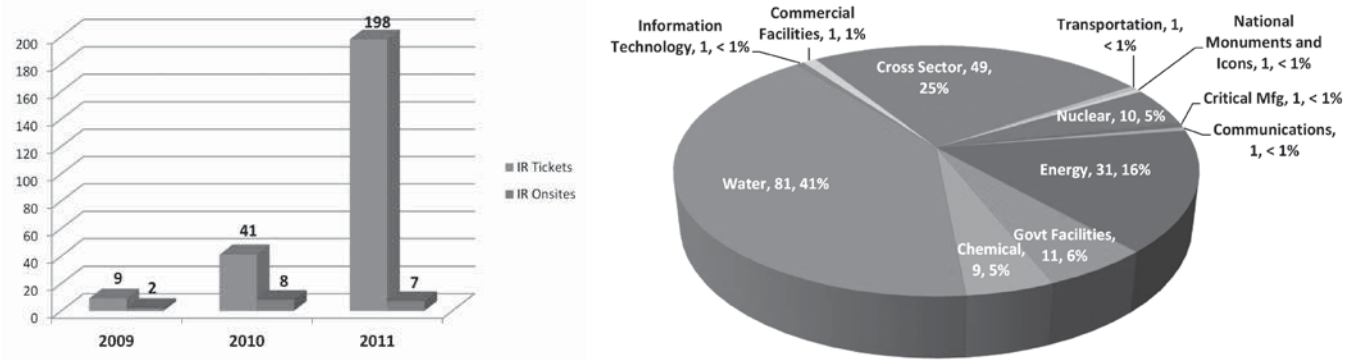


図４　ICS-CERT の 2009-2011 年インシデントレスポンス件数及び 2011 年の分野別報告割合

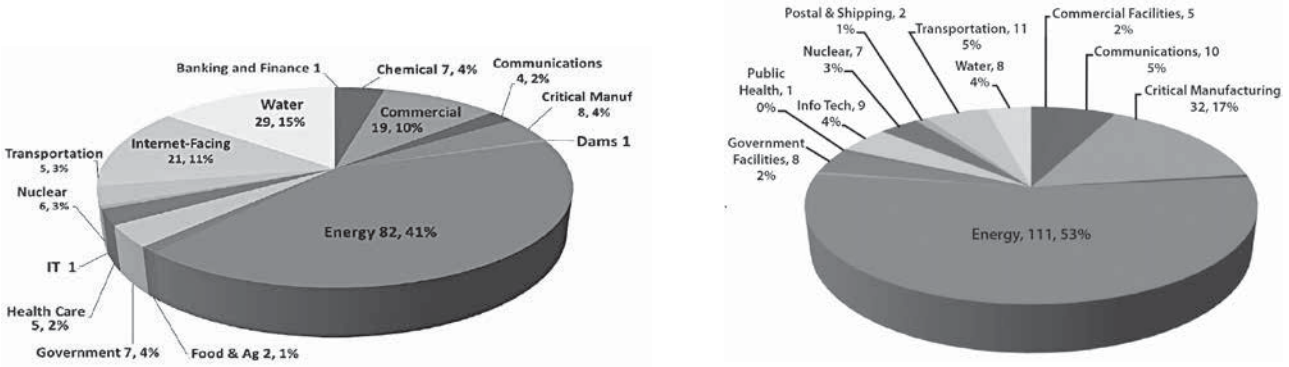


図５　2012 年（FY）及び 2013 年（10 月～翌 5 月）における分野別インシデントレスポンス件数

与えるインシデントが増加傾向にある。米国土安全保障省（Department of Homeland Security：DHS）の公表によると、過去 3 年間の米国内の制御システムインシデントの報告件数は、2009 年が 9 件、2010 年は 41 件、2011 年は 198 件と、急激な増加傾向にあることが報告されている※4（図 4）。

この報告の中では、水道、エネルギー分野でのインシデント報告件数が特に高い割合となっている。さらに、2012 年においても、インシデント報告件数は 198 件※5（※ 10 月～翌 9 月）と、依然として高い水準にあり、特にエネルギー分野において著しい増加が見られる（図 5）。また、2013 年においても 5 月末時点で既に 204 件※6（10 月～翌 5 月）が報告されている。

【脚注】

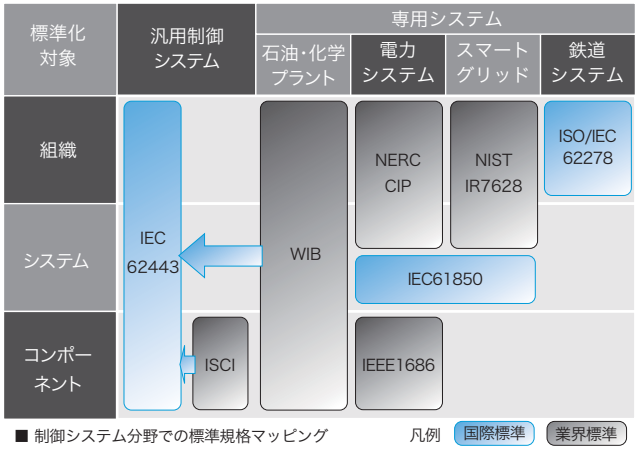
※4　http://ics-cert.us-cert.gov/sites/default/files/documents/ICS-CERT Incident Response Summary Report (2009-2011).pdf

※5　http://ics-cert.us-cert.gov/sites/default/files/ICS-CERT_Monthly_Monitor_Oct-Dec2012_2.pdf

※6　http://ics-cert.us-cert.gov/sites/default/files/ICS-CERT_Monitor_April-June2013.pdf

このような状況下において、制御システムへのサイバーセキュリティ対策は国家の安全保障、危機管理上重要な課題となってきている。

重要インフラの一端を担う制御システムが攻撃された場合、システムの停止や誤作動などにより、社会インフラへ大きな影響を及ぼす危険性がある。海外では、制御システムセキュリティに関する国際規格の整備が進むとともに、規格に基づく認証制度が確立されてきており、制御システムの輸出の際の要件にも加わり始めている。このような状況を鑑み、2010 年に経済産業省で実施された「サイバーセキュリティと経済研究会」の提言として「制御システムの安全性確保」が挙げられた※7。それを受け、2011 年には同省の下で「制御システム情報セキュリティ検討タスクフォース」が実施され、標準化や評価認証などの実現が検討されてきた※8。



■ 制御システム分野での標準規格マッピング

図 6　制御システム分野での標準規格マッピング

表 1　IEC62443 の構成及び標準化のステータス

分類	規格	現状ステータス	リリース
General	62443-1-1	Ed.1：発行済 Ed.2：ドラフト（CD）	Ed.1：2009.07 —
	62443-1-2	ドラフト（DC）	—
	62443-1-3	ドラフト（DC）	—
	62443-1-4	ドラフト（CD）	—
Asset owner	62443-2-1	Ed.1：発行済 Ed.2：1stDC リリース	Ed.1：2010.10 Ed.2：—
	62443-2-2	提案段階（NP）	—
	62443-2-3	ドラフト（DC）	—
	62443-2-4	ドラフト（CD）	—
System integrator	62443-3-1	発行済	2009.07
	62443-3-2	ドラフト（DC）	—
	62443-3-3	発行済	2013.08
Component Provider	62443-4-1	ドラフト（DC）	—
	62443-4-2	ドラフト（DC）	—

※ CD：Committee Draft、DC：Document for Comments、NP：New Work Item Proposal、FDIS：Final Draft for International Standard

2 制御システム分野に関連する標準規格について

制御システムのセキュリティの標準・基準には、組織やシステムのレイヤに対応したもの、業種や業界に対応したもの等、様々な標準・基準が提案されている。このような状況下において、汎用的な標準・基準として IEC62443 が注目され、一部事業者の調達要件に挙がってきている。一方で、制御システム分野では、既存の業界標準を用いた評価認証が ISCI※9、WIB※10 などで実施されている。ISCI では 2010 年より制御システムを構成する個々のコントローラなどのコンポーネントに相当する製品の認証制度を運用している。WIB では、事業者によるシステムの調達の際のセキュリティ要件を広く規定しており、石油・化学など一部の業界でその認証が利用されてきている。一方で、この評価認証で先行していた ISCI や WIB の要求事項（評価基準）が、IEC62443 のシリーズに標準案として提案される動きとなっている（図 6）。換言すると、組織からコンポーネントまでのレイヤをカバーする汎用の制御システムセキュリティに対する国際標準が、評価認証スキームを兼ね備えることになり、その適用や普及が推進されるものと考えられる。次項では、この IEC62443 の構成と内容について解説する。

3 工業用プロセス計測制御のセキュリティ規格（IEC 62443）

国際電気標準会議（International Electrotechnical Commission：IEC）では、電気、電子技術分野の国際標準・規格を作成し、その普及を図ることを目的としている。

【脚注】

※ 7　http://www.meti.go.jp/press/2011/08/20110805006/20110805006.html

※ 8　http://www.meti.go.jp/committee/kenkyukai/shoujo/controlsystem_security/report01.html

※ 9　ISCI：ISA Security Compliance Institute（米国の ISA セキュリティ適合性協会）の略称。ISA のメンバのコンソーシアムにより創設された EDSA 認証の制度運営元（スキームオーナ）である。http://www.isasecure.org/

※ 10　Working-party on Instrument Behaviour の略称、1982 年以降は International Instrument Users' Association と呼称。欧州石油メジャーが中心となり、制御機器ベンダに対するセキュリティ調達要件を規定している。

IEC を構成する専門委員会（Technical Committee：TC）の一つである TC65 では、工業用プロセス計測制御に関する標準化を行っている。TC65 の配下にある WG10 では、ネットワーク及びシステムのセキュリティに関する標準化を推進している。本稿では、WG10 において策定中の規格である「IEC62443」の標準化活動について解説する。

IEC62443 は大別して 4 つの分類があり、発行済み、策定中の規格を合わせて総計 12 の規格が存在する（表 1 [ドラフト段階のステータスは 2012 年時点の参考値]。

[IEC62443-1 シリーズ]

IEC62443 の中で用いられる用語の解説や、制御システムのセキュリティ動向、SCADA モデルの一般論などを記載している。このシリーズは事業者や、システムインテグレータ、コンポーネントプロバイダなど、すべての関係者が共通して参照する規格となっており 4 つの規格から構成されている。現在は、IEC62443-1-1 のみが発行済みであり、他は現在策定中となっている。IEC62443-1-1 では、7 つの基礎的な要件（Foundational Requirement：FR）を規定している。

＜ IEC62443-1-1 ＞

用語、コンセプト、モデルの定義について記した技術仕様書（Technical Standard：TS）である。これには、IEC62443 に用いられる用語の解説や、制御システムの動向や状況、セキュリティ概念、及び SCADA モデルの一般論などを記載している。初版は 2009 年 7 月に発行済であるが、2013 年現在、第二版が策定中である。

＜ IEC62443-1-2 ＞

用語、略語について記した技術報告書（Technical Report：TR）である。IEC62443 に用いられる制御システムのセキュリティに関連する用語・略語集となっている。2013 年現在、草案段階であるが用語を 243 個、略語（Abbreviated terms and acronyms）を 117 個登録している。草案（Documents for Committee）の策定中である。

＜ IEC62443-1-3 ＞

システムの安全性評価基準の規定について記した文書（International Standard：IS）である。これには、評価基準（metrics）策定や利用のためのフレームワークなどを記載している。2013 年現在、草案（Documents for Committee）の策定中である。

[IEC62443-2 シリーズ]

事業者や運用者などの組織を対象としたセキュリティ要求事項などを規定した規格である。このシリーズは 4 つの規格から構成されており、現在 IEC62443-2-1 が発行済みとなっている。このほか、IEC62443-2-2、IEC62443-2-3、IEC62443-2-4 についてはドラフトの策定中となっている。

＜ IEC62443-2-1 ＞

制御システムのセキュリティプログラム確立方法について規定した文書（IS）である。CSMS（Cyber Security Management System）というセキュリティマネジメントプログラムの規格となっており、これは既存規格である ISMS をベースに制御システムのセキュリティに関する要求事項が記載されている。初版は 2010 年 10 月に発行済であるが、2013 年現在、第二版が策定中である。

＜ IEC62443-2-2 ＞

制御システムのセキュリティプログラムの運用ガイドラインについて規定した文書（IS）である。運用する際に必要となる対策ついて、セキュリティポリシー、組織（Organization of security）、資産管理（Asset Management）、人的資源セキュリティ（Human Resources Security）、物理環境セキュリティ（Physical and Environmental Security）など、を記載している。2013 年現在、草案（Documents for Committee）の策定中である。

＜ IEC62443-2-3 ＞

制御システムにおけるパッチ管理方法に関するガイドラインについて記した技術報告書（TR）である。制御システムへのパッチ適用に関する問題点を導入とし、事業者の要件、製品提供者の要件、パッチ情報交換時の要件などについて記載している。Annex としてパッチ報告の書式やパッチについての制御システムの事業者のガイドラインも含んでいる。2013 年現在、草案（Documents for Committee）の策定中である。

＜ IEC62443-2-4 ＞

制御システムの提供者に対するセキュリティ要求事項

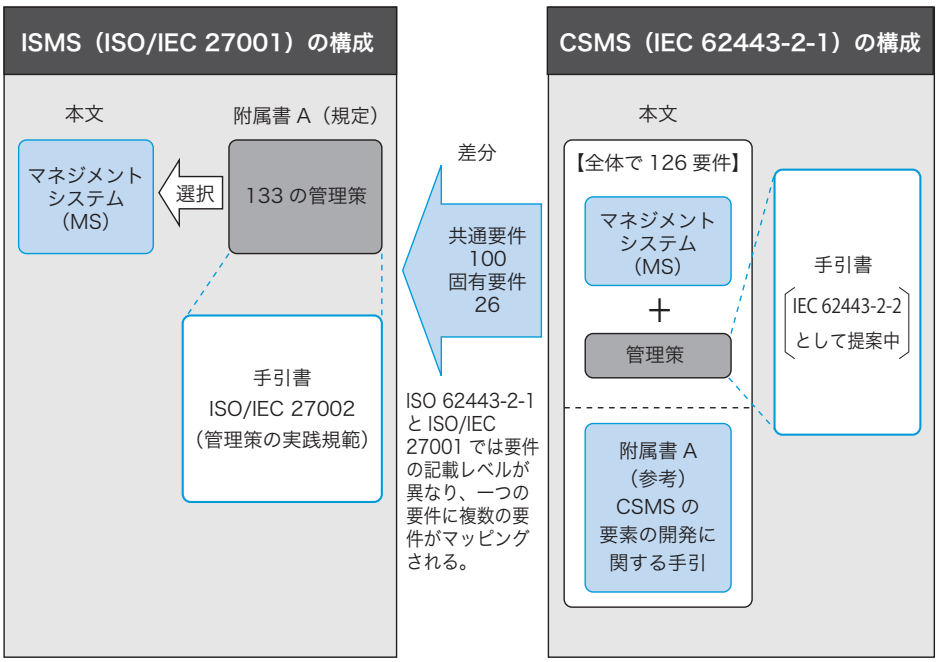
などを規定した文書（IS）である。業界で先行している認証を基に、事業者が制御システムのコンポーネントやシステムを調達する際に必要な要件などが本規格に提案されている。2013 年現在、草案（Committee Draft）の策定中である。

[IEC62443-3 シリーズ]

複数の機能や製品を組み合わせて運用している制御システムを対象とした規格である。このシリーズは 3 つの規格から構成されており、IEC62443-3-1 及び IEC62443-3-3 が発行され、IEC62443-3-2 はドラフト案の策定中である。IEC62443-3-3 については、IEC62443-1-1 で規定されている 7 つの基礎的な要件（FR1 から FR7）に対応する形で技術的なシステム要件を規定している。システム要件は、基本的な要件（System Requirement：SR）と強化策（Requirement Enhancement：RE）から構成されている。それぞれの要件には、セキュリティレベル（Security Level：SL）が割り当てられている。SL は、各要件を満たした場合に、どのような攻撃からシステムを保護できるかを示すものである。4 段階のレベルが規定されており、最も高度な要件を満たすものをレベル 4 としている。

表 2　セキュリティレベル（SL）の定義

SL	対策可能な攻撃（攻撃者）の特徴				
	悪意の有無	攻撃手段	使用リソース	スキルレベル	動機
1	無	-	-	-	-
2	有	単純	低	一般的	低
3	有	洗練	中	システム固有	中
4	有	洗練	高	システム固有	高



出典 :IPA「制御システムにおけるセキュリティマネジメントシステムの構築に向けて」2012 年 10 月

図 7　ISMS と CSMS の関係

＜ IEC62443-3-1 ＞

一般的なセキュリティ技術のうち、制御システムで適用可能なものについて、解説などを記載した技術報告書（TR）である。セキュリティ技術の解説書という位置づけであり、認証、フィルタリング／ブロッキング／アクセス制御、暗号／データ保護、管理・監査・証跡、ソフト管理（脆弱性対応を含む）、物理セキュリティ、人的セキュリティなどを記載している。初版は既に 2009 年 7 月に発行済である。

＜ IEC62443-3-2 ＞

ゾーン（領域）やそれらを連結するコンジットに関するセキュリティについて規定する文書（IS）である。ゾーン及びコンジットやセキュリティ要求事項の定義などが規定されている規格である。ゾーンやコンジットを適切に確立することに目的としている。2013 年現在、草案（Documents for Committee）の策定中である。

＜ IEC62443-3-3 ＞

制御システムのセキュリティ機能要件を規定した国際標準（IS）である。IEC62443-1-1 で規定されている 7 つの基礎的な要件（FR1 から FR7）に対応する形で技術的なシステム要件を規定している。システム要件は、基本的な要件（System Requirement：SR）と強化策（Requirement Enhancement：RE）から構成されており、SR や RE ごとにセキュリティレベル（Security Level：SL）が割り当てられている。SL は、各要件を満足した場合に、どのような攻撃からシステムを保護できるかを示すものである。4 段階のレベルが規定されており、巧妙で大

規模な攻撃にも対処可能なレベルをレベル 4 としている。この SL の定義を表 2 に示す。初版は 2013 年 8 月に発行済。

[IEC62443-4 シリーズ]

制御システムの一部である個別のコンポーネント単位が準拠の対象となる規格である。このシリーズの規格として、現在 IEC62443-4-1 及び IEC62443-4-2 の 2 つが存在するが、いずれもドラフト案の策定中となっている。

＜ IEC62443-4-1 ＞

コンポーネントの開発要件を規定した国際標準（IS）である。セキュアなコンポーネントを開発するための方法を規定しており、ISA Secure の EDSA（SDSA）をベースにしている。内容は、ソフトウェア開発のライフサイクルを 12 の段階に分けて、それぞれのセキュリティに関する要求事項を記載している。2013 年現在、草案（Documents for Committee）の策定中である。

＜ IEC62443-4-2 ＞

コンポーネントのセキュリティ要件を規定した国際標準（IS）である。デバイスに搭載されるセキュリティ機能を規定。ISA Secure の EDSA（FSA）をベースにしており、セキュリティ機能の実装評価に関する要求事項を記載している。2013 年現在、草案（Documents for Committee）の策定中である。

なお、既に発行済みの規格（IEC62443-1-1、2-1、3-1）に対しては、IPA により作成された英日対訳版が、日本規格協会から発刊されている※ 11。

4 我が国の評価認証への取り組み

4.1 日本発のセキュリティマネジメントの認証プログラム「CSMS」のパイロットプロジェクトの開始

制御システムにおけるセキュリティ対策の必要性の高まりに対応するため、制御システムを利用する事業者のセキュリティマネジメントシステムの確立が非常に重要となってくる。本規格の IEC62443-2-1^{ed1} は、制御システムのセキュリティマネジメントを規定しており、CSMS※ 12 と定義している。IPA では前述の「制御システム情報セキュリティ検討タスクフォース」において、この CSMS に基づく認証制度の実現を提案してきた。現在、経済産業省及び一般財団法人日本情報経済社会推進協会（JIPDEC）において、日本発の制御システム向けセキュリティマネジメントシステム適合性評価制度の確立が進められている。本制度では、CSMS（IEC62443-2-1^{ed1}）規格の要求事項を用いて、世界的にも突出して高い認証実績を有する ISMS 適合性評価スキームに沿った評価認証を実施することが予定されている。また、既に ISMS を取得している事業者等においては、認証の重複を省くなどの効率的な認証スキームの推進も今後検討していくことで、制御システム分野にセキュリティマネジメントシステムを普及、浸透させるきっかけとなる事が期待される。

経済産業省は、「グローバル認証基盤整備事業」による政策に基づき、JIPDEC を主体とした CSMS のパイロットプロジェクトを 2013 年度に実施している。本パイロットプロジェクトでは、認証機関が審査に用いる認証基準、及びこれに基づく認証機関に対する認定基準を策定し、これらを用いて実証実験を目的とした試行の受審となるパイロット認証を実施している。このパイロット認証においては、CSMS の認証機関候補が制御システムを利用する事業者（受審企業候補）に対して実際に CSMS への適合性を評価し、実証実験としてその結果を認証基準やスキームへフィードバックすることが盛り込まれている。本パイロットプロジェクトは ISMS 創設時に実施された「ISMS パイロット事業」と同等の位置付けとして実施されており、本プロジェクトにおいて抽出された課題などを解消した上で、2014 年度を目途に正式な認証制度が施行される見込みとなっている。

ISMS は、準拠する標準である ISO/IEC27000 シリーズが ISO によって改正が進められているところであり、IEC62443-2-1 の標準化においても、これに沿った整合が図られる見込みである（図 7）。IEC62443-2-1^{ed2} の案では、ISMS の箇条体系に沿った形式でのリリースが予定されており、認証制度についてもこれに伴って改正されていくことによって、より一層、ISMS と親和性の高い CSMS 認証が確立、普及していくことが期待される。

【脚注】

※ 11　一般財団法人日本規格協会 http://www.jsa.or.jp/

※ 12　CSMS：Cyber Security Management System の略称。制御システムにおけるセキュリティマネジメントに関する要求事項を規定している。

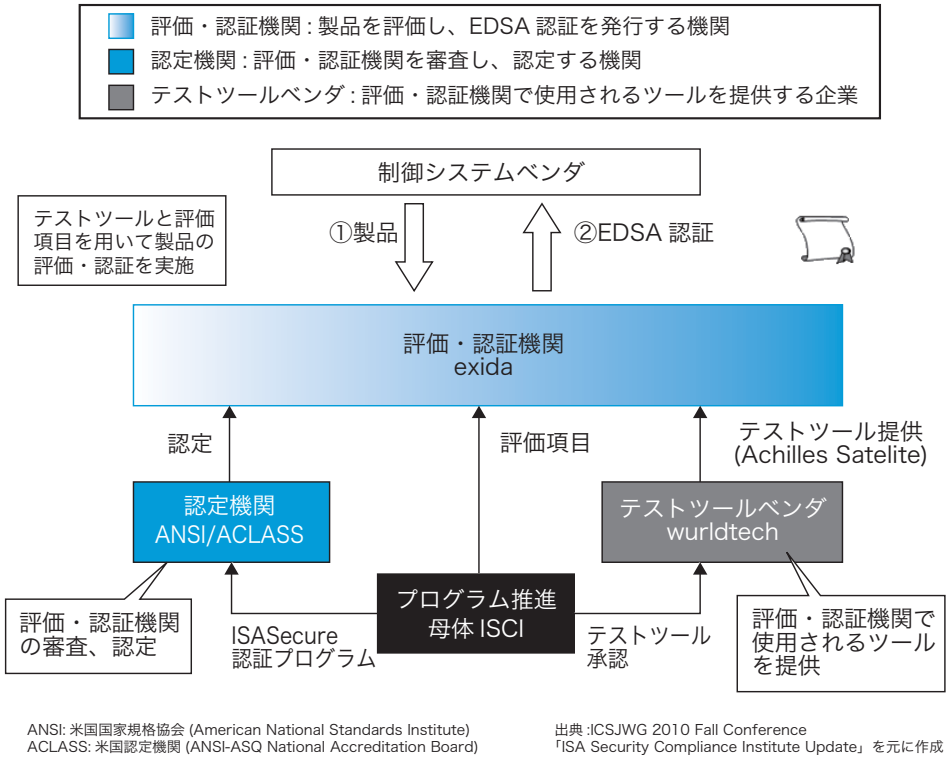


図 8　EDSA スキームの概念図

4.2　組込み機器のセキュリティ保証プログラム「EDSA」の国内導入

ISCI がスキームオーナを担っている組込み機器のセキュリティを保証する認証プログラムである EDSA は、IEC62443-4（制御システムセキュリティ）シリーズの標準化の場へその要求事項が提案され、承認される見込みとなっている。EDSA 認証は ISA※13 のメンバー（民間企業主体）により創設された ISCI がスキームオーナとなり、運営されている認証スキームである。EDSA 認証の主な評価項目は「通信の堅牢性試験（CRT※14）」、「セキュリティ機能の実装評価（FSA※15）」、「ソフトウェア開発のライフサイクルの各フェーズにおけるセキュリティ評価（SDSA※16）」に大別される。なお、SDSA の要求事項には IEC61508（電気 / 電子 / プログラム可能電子安全関連システムの機能安全）、ISO/IEC15408（IT セキュリティの評価基準）などが引用されている。EDSA 認証において、現状は米国 ANSI※17 のみが唯一の認定機関となっており、同じく米国の exida.com, LLC が唯一の評価・認証機関を担っている。EDSA スキームにおいて、評価・認証機関は、スキームオーナの認可する評価ツールを用いることとされている。現状、スキームオーナより認可されているツールは wurldtech 社ツール（Achilles Test Platform）及び Codenomicon 社ツール（Codenomicon Defensics）がある。現状の EDSA スキームの概念図を図 8 に示す。

現状の EDSA 認証スキームは北米主体で先行しているが、北米だけでなく国際的な製品の調達要件に挙げられ始めている。このため、国内の企業からも EDSA 認証取得に関する要望があるため、国内においても認証が可能となるよう取り組んでいる。ISCI のメンバである IPA では、2012 年 10 月から、日本の JAB※18 を EDSA 認証スキームの認定機関として登録するための活動を実施しており、2013 年 3 月には米国 ISCI、ANSI、JAB、IPA の 4 者会合にて IPA より交渉を行い、正式に日本スキームの確立が承認された※19。これにより、国内での認証機関の認定、及び製品認証取得が実現する見込みとなった（図 9）。IPA は、国内での認証制度の実現計画を策定し、国内に同スキームの認定機関及び認証機関の設置を提案及びその支援を実施した。この結果、2012 年に設立された技術研究組合制御システムセキュリティセンター（CSSC）内に、同制度の認証機関が設置され、現在その正式な認定を取得する作業が推進されている。

【脚注】

※ 13　ISA：International Society of Automation（国際計測制御学会）の略称。
※ 14　Communication Robustness Testing
※ 15　Functional Security Assessment
※ 16　Software Development Security Assessment
※ 17　ANSI：American National Standards Institute（米国国家規格協会）の略称。米国内の工業製品の規格を策定する団体。EDSA 認証スキームにおける認定機関を担っている。http://www.ansi.org/
※ 18　JAB：Japan Accreditation Board（公益財団法人 日本適合性認定協会）の略称。適合性評価制度全般に関わる認定機関としての役割を担う組織。http://www.jab.or.jp/
※ 19　制御機器認証プログラム「EDSA」国内認証制度の確立および規格書対訳版の公開について http://www.ipa.go.jp/about/press/20130415.html

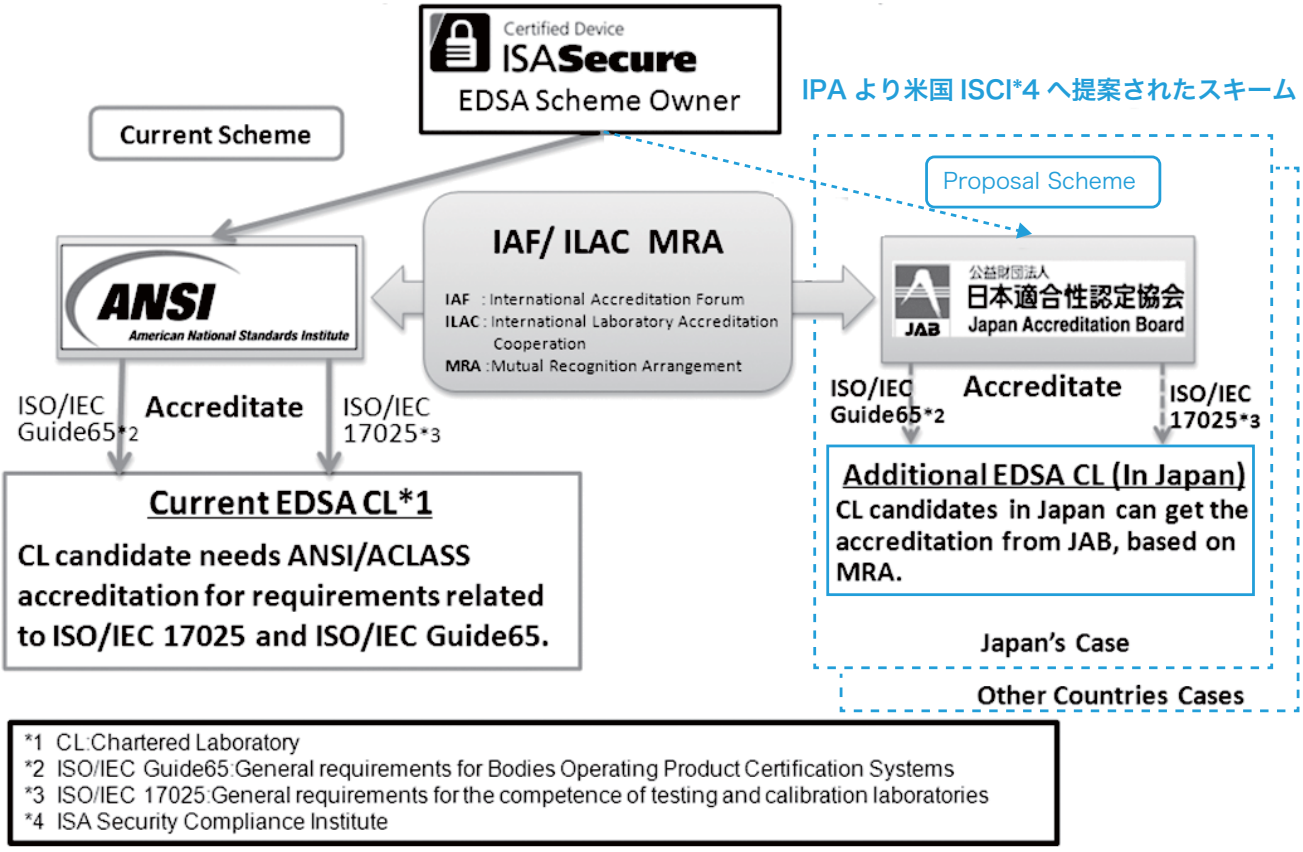


図 9　EDSA 認証スキームへの日本参画構想

5 今後の展望

我が国内では、制御システムのセキュリティへの対応として、IEC62443 の活用を軸に、その普及啓発が進められていく見通しである。これをベースに、日本発の制御システムのセキュリティマネジメントシステムの評価認証制度が開始され、制御機器などの個別のコンポーネントに関しても米国に次ぐ形で同じく評価認証制度が開始される予定である。これらはいずれも 2014 年度から施行される見込みとなっており、今後のセキュリティ対策の発展が期待される。マネジメントシステムについて、制御システムを用いる各分野に共通で汎用的な要求事項が求められており、これに適合するための指針となるガイドの策定が併せて進んでいる。また、将来的には業界ごとにその指針を詳細化したガイドが策定及び普及されることが望まれる。一方で、「システム」のレイヤに対応した IEC62443-3-3 が発行されており、これを用いた認証制度として SSA（System Security Assurance）の策定が ISCI により進められている。こちらも我が国として、その導入の可否が検討段階に入っている。

制御システムへの脅威、業界・国際規格の動向とこれに基づく認証制度の策定状況については流動的となっているため、引き続き、関係業界・各国などと連携を保ちながら、我が国として国内の制御システムのセキュリティ向上と制御システム製品の国際競争力の強化の観点から、タイムリーに標準の活用と評価認証制度の拡充を図っていくことが望まれる。

【参考文献】

[1] IPA,"CSMS/EDSA 認証導入に向けたパイロットプロジェクト ", http://www.ipa.go.jp/security/fy24/reports/ics_sec/ics_annex.pdf

[2] 入澤康紀 , IPA," 制御システムセキュリティ標準「IEC62443」 ", http://techon.nikkeibp.co.jp/article/FEATURE/20130130/263280/

[3] 入澤康紀 , IPA," 政府による制御システムのセキュリティへの取組み ", http://techon.nikkeibp.co.jp/article/FEATURE/20130128/262781/

[4] IPA," 制御システムにおけるセキュリティマネジメントシステムの構築に向けた解説書の公開 ", http://www.ipa.go.jp/security/fy24/reports/ics_management/index.html

[5] 入澤康紀 , IPA,"EDSA の認証プログラム ", http://techon.nikkeibp.co.jp/article/FEATURE/20130130/263302/

[6] 入澤康紀 , IPA," 制御システムのセキュリティマネジメントシステム ", http://techon.nikkeibp.co.jp/article/FEATURE/20130130/263303/

パッケージソフトウェア品質（PSQ）認証制度創設１年目の展開

ISO/IEC 25051:2006

一般社団法人コンピュータソフトウェア協会（CSAJ）PSQ 認証室
中野　正　　鈴木　啓紹

「パッケージソフトウェア品質認証制度」は、一般社団法人コンピュータソフトウェア協会（CSAJ※1）が 2013 年 6 月に運営を開始した、日本で初めてのパッケージソフトウェアを対象とした第三者適合性評価による製品認証である。本制度では公正性を担保するため、独立行政法人情報処理推進機構（IPA）が規定した「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」と呼ぶ）を参照している。
ここでは本制度の概要と、これまでに実施された２回の認証判定における認証取得企業からの評価等について概説する。

1 はじめに

利用者においては、効率的かつ戦略的な情報システムを構築するうえで、パッケージソフトウェアが個別（スクラッチ）開発ソフトウェアと比較して、選択から導入までのスピードやコストといった点で優位性があるという認識が浸透しつつある。

日本国内のパッケージソフトウェア製品は、製造者の品質に対する意識が高く、高品質が維持されている。この背景には、利用者からの「品質が高くて当然」という日本市場の特性と、社会的信用を落とさないために入念な品質管理を企業文化として一般化していることが要因と考えられる。

パッケージソフトウェア製品は、多くの場合、製造者（供給者）の開発部門と品質管理部門による独自の基準と方法で品質を担保した品質保証が行われている。よって、国内パッケージソフトウェア業界では「国内パッケージソフトの品質保証はそのベンダ自身が発信するのみ」といった点について、かねてから危惧されてきた。

一方で、グローバル化が進む昨今、「品質の見える化」が重視され、欧米では第三者による品質保証情報の提供が一般化している。つまり世界では、どの品質基準に従い、どのように製品化し、誰が品質を確認したのか説明するように求められているのである。それに伴い、国際基準では早くから品質に関する要求事項を整理し、第三者機関による適合性評価を定めて規格化が進められていた。

日本のパッケージソフトウェア業界も世界に向けて「安心・安全・高品質」を証明するための第三者適合性評価の品質認証制度を設立し、パッケージソフトウェア製品の品質を客観的に利用者に提示できるようにする必要があった。

そこで、CSAJ では、品質の見える化、品質説明力強化を目的とした第三者認証を実現するために、パッケージソフトウェア製品に対して最低限の安全性あるいはビジネス上の重要な品質要求事項に関して業界標準となりえる品質基準を策定した。この基準に則り、使用者にとって安全性・信頼性の指標となり、かつ、高品質なパッケージソフトウェアの普及を更に促進するための制度として「パッケージソフトウェア品質（PSQ※2）認証制度」を創設した。

2 制度概要

PSQ 認証では、業界団体である CSAJ が、IPA の制度ガイドラインに沿って、認証機関※3 として公平・公正な第三者認証制度を実現する仕組みづくりをしている。

認証機関は適合性判定業務に特化するために評価業務を専門性を持った第三者へ委託し、判定を公正に実施するために外部有識者で組織した判定委員会を設置している。さらに認証制度全体の公正性を担保するために、内部でのマネジメントレビューと外部の公正性委員会によるチェック機構を設け、内外から適正な認証業務が行われているか確認している。そして、申請者は CSAJ 会員 / 非会員を問わず対象とすることで、パッケージソフトウェア業界内に開かれた制度となっている。

認証制度のフローとしては、認証対象製品を有するパッ

【脚注】
※ 1　Computer Software Association of Japan
※ 2　Packaged Software Quality
※ 3　制度ガイドラインの制度責任主体に相当

ケージソフトウェアベンダからの申請を認証機関が取りまとめ、認証機関が承認した評価機関に対象製品の評価作業を依頼する。

評価機関は、後述する評価基準に基づき、ドキュメントによる評価及び申請者を訪問する現地調査で構成された評価を実施し、その結果を評価報告書にまとめて認証機関に提出する。

この評価報告書に基づき、認証機関内に設置されている、国際規格に対する有識者を中心とした外部有識者で組織された判定委員会で、評価基準に対する適合性を判定する仕組みとなっている（図 1）。

また、認証機関では恒常的に本制度の改善を図るため、技術的な改善点を検討する技術委員会、申請フローや対象製品の定義など制度に影響を及ぼす改善点を検討する認証制度委員会を設置している。

＜ PSQ 認証制度で参照した規定類＞

- 公正性が担保された制度設計を行うための基準として IPA が規定した「ソフトウェア品質説明のための制度ガイドライン」を参照
- 認証機関は「製品認証機関に対する一般要求事項の適用に関する指針（ISO/IEC Guide65:1996（JIS Q 0065:1997））」及び「適合性評価－製品認証の基礎（ISO/IEC Guide67:2004（JIS Q 0067:2005））」のシステム 1b に基づき規定を策定。ISO/IEC Guide65:1996（JIS Q 0065:1997）は、後に ISO/IEC 17065:2012（JIS Q17065:2012）「適合性評価－製品、プロセス及びサービスの認証を行う機関に対する要求事項」に改定され、本制度ではこちらを継続して参照している）
- 評価機関は「試験所及び校正機関の能力に関する一般要求事項（ISO/IEC17025:2005（JIS Q17025:2005））」に基づき認証機関が承認する第三者で構成。（独立評価機関相当の外部専門企業に委託）

3 評価対象と評価基準

個別開発（スクラッチ）のシステムは利用者が仕様策定からかかわって構築するのに対して、パッケージソフトウェア製品はほとんどの場合、一定の標準仕様に基づいた完成品を利用者が選択し、導入する。したがって、パッケージソフトウェア製品は、開発プロセスに関する間接的な品質よりも、完成した製品に関する直接的な品質を確認する方法が適切であると判断した。後者の品質確認方法について調査した結果、ISO/IEC25051:2006（JIS X 25051:2011）の規格を参照することとし、評価基準として具体化する作業を行った。

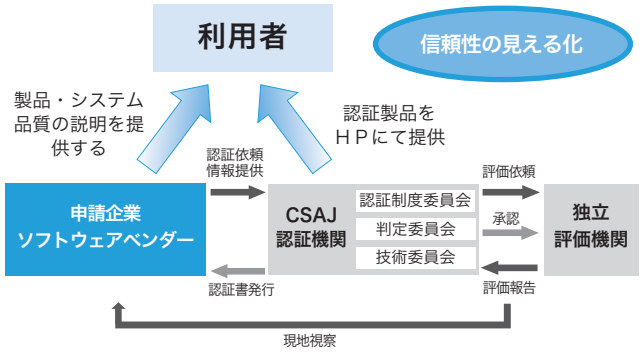


図 1　PSQ 認証の仕組み

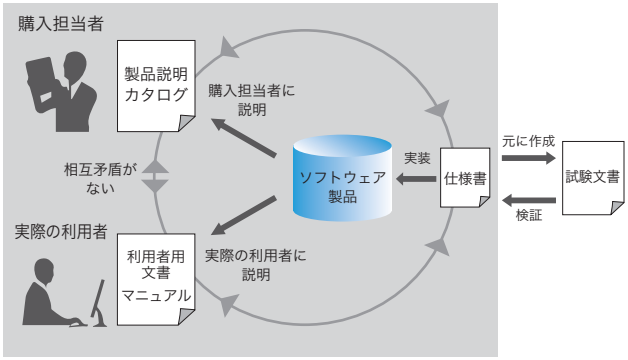


図 2　PSQ 認証での文書間の相関関係

ISO/IEC25051 に基づいたパッケージソフトウェアの品質は、「製品説明※4」「利用者用文書※5」「試験文書」とソフトウェアが実装する機能で評価する（図 2）。評価のために評価機関が実際の環境下でソフトウェアを稼働させたり、開発マネジメントの観点から間接的に品質を評価するものではない、という点が本制度の特徴であるといえる。言い換えれば、提供側における品質基準や開発プロセスではなく、できあがった製品に対する直接的な品質を見るものであり、あくまでソフトウェアの購入予定者、利用予定者にとって、要求した通りの機能が正しく提供されているか、また、選定に必要な情報が十分に提供されているかを評価している。

製品説明の評価基準

- 入手／参照のしやすさ、利用のしやすさ
- 利用者の要求に対するソフトウェアの整合性を判断できる情報を含んでいるか
- 製品の特徴、機能、性能、制約などの情報が明確かつ正確に記載されているか

【脚注】

※ 4　パッケージ外装表示、データシート、ウェブサイト情報など購入前に参考とする情報

※ 5　ソフトウェアのインストール及び使用に必要な情報

利用者用文書の評価基準

- 利用する上で必要な情報が正しく記載されているか
- 製品説明と矛盾や不一致がないか
- 想定している利用者にとって理解しやすいか

試験文書の評価基準

- 利用する上で製品説明や利用者用文書に記載されている製品の機能が正しく実装されていることを、試験によって確認されているか
- 製品の性能や使いやすさなどが試験によって確認されているか
- 試験を実施した結果や合否判定が具体的かつ明確に記載されているか

また、評価基準の中では（ISO/IEC9126-1:2001（JIS X 0129-1:2003））で定義されたソフトウェア品質特性を用いている※6。ソフトウェアに対する品質要求事項を品質特性に照らし合わせることで「このソフトウェア製品が達成すべき品質は何か」を明確化することができる仕組みとなっている（図 3）。

この図では、ソフトウェア製品に求められる品質をソフトウェア製品の品質特性と品質副特性に分類し、階層構造で整理している。図中の「内部品質」「外部品質」とは、以下を指す。

- 内部品質 ……ソフトウェア開発工程の各（中間）成果物から測定される品質
- 外部品質 ……開発された製品自体を動作させて測定される品質

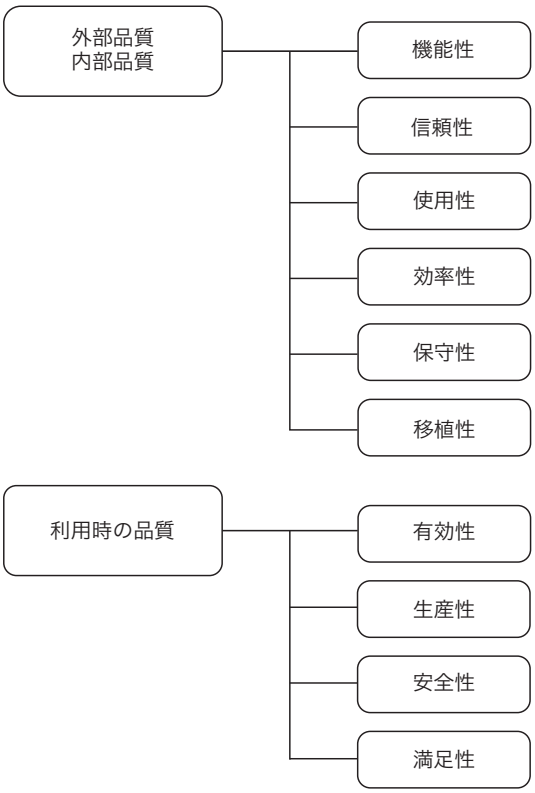


図 3　品質モデル

利用時の品質特性は、ソフトウェア製品が実際に使われている状況における品質を考慮したモデルである。

パッケージソフトウェアの企画段階から「この製品が達成すべき品質は何か」が考慮・検討されることはまだ少ないかもしれないが、今後、本制度の普及に伴ってソフトウェア品質特性への理解と対応が進むことが望まれる。

4 品質説明力強化に加えて利用者視点によるプロセス改善へ発展

判定委員会は現状、四半期に一度開催されており、2013 年は 8 月と 11 月に開催された結果、以下 9 社 11 製品が適合と判定された。

製品名	申請企業
サイボウズガルーン	サイボウズ株式会社
Speedy Call	ネクストウェア株式会社
土留め工の性能設計計算（弾塑性解析 II+）	株式会社フォーラムエイト
置換基礎の設計計算 Ver.2	株式会社フォーラムエイト
iOptMICS（アイオプトミクス）	東京システムハウス株式会社
勘定奉行 V ERP8（勘定奉行 i8）	株式会社オービックビジネスコンサルタント
PCA 給与 X	ピー・シー・エー株式会社
SMILE BS2 販売	株式会社 OSK
SMILE BS2 会計	株式会社 OSK
Power Steel	日本ナレッジ株式会社
CORE Plus NEO 通販	日本事務器株式会社

判定終了後、認証取得者に対してアンケート / インタビューを実施した。評価の過程を通じて申請者が感じた PSQ 認証取得のメリット / 効果を以下にまとめる。

PSQ 認証制度は、ISO/IEC25051 に基づき当該ソフトウェアの信頼性を担保することでユーザ視点の製品説明力強化を促すことを主な目的としているため、一連の評価を受けた申請者側担当者から

- 社外に対する品質のアピールが容易になった

という声が聞かれることは予想されたが、以下の評価もいただいている。

【脚注】

※ 6　JIS X25051:2011（ISO/IEC 25051:2006）は JIS X 0129-1:2003（ISO/IEC9126-1:2001）の品質特性が用いられていたが、ISO/IEC9126-1:2001 が ISO/IEC 25010:2011 へ、JIS X 0129-1:2003 が JIS X 25010:2013 に引き継がれたことで、ISO/IEC 25051 の改定時にはその参照先を JIS X 25010:2013（ISO/IEC 25010:2011）に変更の予定

・ 申請書類の作成と試験文書の確認に労力を費やしたが、結果として開発工程や管理体系の見直し・改善に役立った。

そもそもの申請理由として

・ すべての工程を社内で行ってきたが、特にパッケージについてはこれといった指針がないため、外部の人に見てもらういい機会と捉えた。
・ 試験文書の管理体系をはじめとする社内の現状のプロセスが正しいかどうかを確認する機会になる。

と言っているように、パッケージソフトベンダがそれぞれで行っている開発工程 / 試験体系に対する外部評価が必要とされていたことがわかる。

長い場合は数十年もシリーズを継続している製品では、

・ 古くから開発を続けてきた製品は、その長い間に工夫や改良されてきたものをこの審査で改めて整理したことで、本来の開発プロセスの基本的な方法や考え方といったものから歪みが生じていたことが分かった。
・ パッケージであるため、コア機能評価がかなり以前のテストでフォローされていたこともあり、試験ドキュメントを整理することが大変だった。

といった声が挙がった。結果として

・ 文書管理体系が作業的にやりやすいほうへ偏っていたために、部外からの視点で見た場合に整理できていたとは言えない状況。

のまま現地審査を迎えることになり、製品によっては 2 日にわたる審査も行われることとなった。

「機能の粒度と試験文書の粒度をマッチできるかがポイント」という言葉に言い表される通り、利用者に提供される機能は必ず試験が実施されているものの、継続的にソフトウェア品質を向上させるためには、試験文書をいつでも（社内の）誰でも参照できる状態が望ましいと考える。

この他、「申請に向けた作業としては、副特性を理解するところからスタートする必要があった」、「機能品質特性の申請書で、各特性に該当するか否かの判断に苦慮した」、「不要だと感じる項目が無いだけに、一つ一つの項目への記載に苦労した」と言っているように、各機能における品質を品質特性に合わせて具体化・整理することについては多くの申請者から「申請から評価の過程における苦労」として挙げられた。

品質特性と試験文書はすなわち ISO/IEC 25051 の評価基準であり、これをクリアしたからこその PSQ 認証取得製品であるといえるだろう。

こうした評価を経たことで「従来の要求仕様に対する開発プロセスではなく、購入し利用する上で良いソフトウェアであるかどうか、つまり、製品の出来上がりの姿に対する開発プロセスや管理方法の基準となりうる認証制度である」と、利用者視点によるプロセス改善へ発展する兆しが見て取れる。

これは PSQ 認証制度で品質特性に触れたことによる、従来の開発工程とは異なる視点と指標への「気付き」であると考えている。従来の上流から下流へといった流れだけではなく、川下であるユーザ視点からソフトウェアに必要な品質を探るといった改善がよりよいソフトウェア開発につながり、ひいては企業経営全体にプラスの影響を及ぼすことが可能となる。

PSQ 認証によって、従来自分たちが正しいと思っていたプロセスも、利用者視点になることで異なる指標が必要になる、ということを理解いただけたと考えている。

5 今後について

国内はもとより国際的にも展開力をもった高品質なパッケージソフトウェアの拡大のため、PSQ 認証制度の普及を更に推進する。

昨今、情報システムが「所有」から「利用」へと移行しようとしている。この流れを受け、認証機関として SaaS/ クラウドを認証対象とした制度対応が急務である。評価基準の参照先である ISO/IEC25051 における SaaS/ クラウドを対象とした改定と同期し、認証対象の拡大を計画している。

また、本認証制度の " 海外展開を進める国産パッケージソフトウェア製品の後押し " という目的を実現する手段のひとつとして国際相互承認がある。そこで、CSAJ は ISO/IEC 国際会議 JTC1 SC7/WG6（ISO/IEC 25051 を含む 25000 シリーズを検討・策定）へ委員参画し、国際会議のメンバーである韓国（GS（Good Software）認証※7 やフランス（NF（NormeFrancaise）認証）など、海外における ISO/IEC25051 準拠の第三者認証制度を運営する国々と情報交換を開始している。

【脚注】

※ 7 検討開始当初、各国の認証制度についても調査を進める中で、候補としてドイツの「Verisoft XT プロジェクト」と韓国の「GS 認証」が挙げられた。結果、より思想の近い GS 認証を対象とすることを決め、同認証取得企業であるユニオン & EC 社の日本事業担当者や KOTRA の協力を得て参考としていた。

その後、韓国 TTA（Telecommunications Technology Association）Software Quality Evaluation Center の Shin 博士を招聘し、関係者を集めた GS 認証に関するセミナーを開催している。その内容については CSAJ のホームページを参照されたい。
http://www.csaj.jp/info/11/110218_ttashin.html

ソフトウェアプロダクトラインのエンタプライズ・システムへの適用と評価



中村　伸裕[†1 †3]　谷本　收[†2]　楠本　真二[†1]

ソフトウェアプロダクトラインは，ソフトウェアを共通性と可変性が事前に整理された再利用資産から開発する手法で，派生ソフトウェアを効率よく開発できる．組込ソフトウェア開発での事例は多いが，エンタプライズ・システム開発には適用が難しいことが指摘されている．本論文ではエンタプライズ・システムへのソフトウェアプロダクトラインの適用の試みについて述べる．ソフトウェア資産としてビジネスロジックではなく画面部品を中心に構築することにより各アプリケーションで開発するソースコード量を削減し，プログラム開発のコスト削減を実現した．また，操作性の高い部品を提供することで利用者の満足度を高めるとともに開発者に部品利用の動機付けを行った．その結果，ソフトウェア資産は 10 年以上の期間にわたり全開発プロジェクトで再利用されており，組織全体で開発したソースコード量は大幅に減少し，開発コストも低減することができた．

Application and evaluation of Software Product Line for enterprise systems

Nobuhiro NAKAMURA[†1†3], Osamu TANIMOTO[†2], and Shinji KUSUMOTO[†1]

Abstract

Software Product Line (SPL) method is one of the reuse technologies that software is developed by adding specific features to a common set of core assets in a prescribed way. Though there are many case studies that applied SPL to embedded software developments, it has been pointed out the difficulties of applying it to enterprise software developments. This paper describes a successful study introducing SPL to enterprise software developments. We constructed mainly GUI components, instead of business logic components, as software assets. By using the components, we have reduced the amount of source code newly developed and attained development cost reduction. We also motivated developers to use the components by providing high operable components for users. As the results of introducing SPL, the software assets have been using for over a decade, and we could greatly reduce the amount of source code size and the development cost.

1. はじめに

ソフトウェアの大規模化と複雑化に伴い，高品質なソフトウェアを短期間に効率よく開発することが求められている．再利用はこれを実現する一つの有効な手法であると考えられ多くの研究が行われてきた [1]．再利用の形態はサブルーチン，モジュール，オブジェクト，コンポーネントと

【脚注】
† 1　大阪大学　Osaka University
† 2　住友電工情報システム株式会社
Sumitomo Electric Information Systems Co., Ltd.
† 3　住友電気工業株式会社
Sumitomo Electric Industries, Ltd.

進化し，2000 年代にはソフトウェアプロダクトライン (以下，SPL)[2][3] の考え方が広まっている．

SPL の 2 つの特徴は (a) ソフトウェア資産を構築するドメイン開発とソフトウェア資産を活用するアプリケーション開発の 2 つのプロセスを分離する，(b) ソフトウェア資産のどの部分が共通でどの部分が可変なのかを明示的に示すことである．SPL は従来のアドホックな再利用ではなく計画的な再利用を実現することが目的となっている．SPL は組込み系の分野では多くの事例が報告 [4][5] されているが，エンタプライズ・システムの事例報告は少ない [6]．

住友電気工業の情報システム部門 (以降，当組織) は社内で利用する生産管理，在庫管理，販売管理，購買管理，物流管理，会計，人事などの事務処理システムの開発を主な業務としており，継続的に品質，コスト，納期の改善に取り組んでいる．本論文では 1999 年から開始した，SPL に基づくソフトウェア部品の再利用により社内の業務システム開発で作成するソースコード量を削減することで開発コストを低減させる取り組みと，2003 年から 2012 年までの 10 年間の実績評価について報告する．ソフトウェア部品へは業務ロジックを対象とするアプローチと画面部品を中心とするアプローチが考えられたが，これまでの経験から後者の方が開発するソースコード量の削減に効果があると考えた．業務システムで利用する画面はそれぞれ表示するデータ項目が異なり，データ長や入力方法が異なるため簡単に部品化することができない．この問題を解決するためにデータ項目をオブジェクト化する項目オブジェクトの考案し画面部品を開発した．また，操作性の高い画面部品を提供することで利用者の満足度を高め，開発者の再利用に対する心理的な抵抗を解消した．さらに演習を中心とした 3 日間のトレーニング・コースを定期開催することで全社展開することができた．これらの取り組みの結果，IPA/SEC が発行している文献 [7] に掲載されている生産性と比較して，3 ～ 5 倍の高い生産性と利用者の高い満足度が実現できている．

2. ソフトウェアプロダクトライン (SPL)

ソフトウェアの再利用は品質，コスト，納期の改善策として期待されてきた．初期段階ではサブルーチンが作成され，モジュール，オブジェクト，コンポーネントと進化し [8]，2000 年頃から計画的な再利用を目的として SPL の考え方が広がっている．従来の製品ごとの開発プロセスと SPL の開発プロセスとの違いは，ソフトウェア資産を開発するドメイン開発とソフトウェア資産を活用して個々のアプリケーション（システム）を開発するアプリケーション開発（以下，ＡＰ開発) の 2 つのプロセスが体系化されていることである．ドメイン開発の主要プロセスは，(1a) ドメイン要求開発，(1b) ドメイン設計，(1c) ドメイン実現，(1d) ドメイン試験であり，これらのプロセスから複数アプリケーションの共通的な要求，アーキテクチャ，コンポーネント，試験に関する成果物といったソフトウェア資産が構築される．ＡＰ開発は，(2a) アプリケーション要求開発，(2b) アプリケーション設計，(2c) アプリケーション実現，(2d) アプリケーション試験のプロセスで構成され，各プロセスがソフトウェア資産を活用する．また，上記 8 プロセスがうまく計画管理できるよう，個々のアプリケーションのリリース計画などを管理する製品管理プロセスがあり，合計 9 プロセスで構成されている．各プロセスの内容は 6 章，7 章で実施内容と共に説明する．なお，本稿では業務システム開発がＡＰ開発に相当する．

3. SPL の導入目的

1999 年にオブジェクト指向言語 Java をサーバーサイドで利用する技術が登場したことをことによりソフトウェア部品の開発環境が容易に入手できるようになった．当組織では以前から開発コスト削減の手段としてオブジェクト指向言語によるソフトウェアの再利用を検討しており，1999 年にソフトウェア部品の開発を進めることになった．当時は個々の開発チームで共通部品を開発し再利用を進めていたが局所的な再利用のため大きな成果は得られていなかった．ソフトウェア資産を構築し再利用によるコスト削減効果を得るためには各開発プロジェクトが開発する成果物量を削減し，開発工数を削減する必要がある．また，ソフトウェア資産の開発投資に対する効果を最大化するためには開発したソフトウェア資産は全プロジェクトに展開し，長期間利用する必要がある．このような全社的な再利用を進めるためには SPL で示されているようにドメイン開発チームとＡＰ開発チームを分離し，ソフトウェア資産の開発にも要求定義から試験のプロセスを実施する必要があると考えた．

SPL 導入の目的は，(a) ＡＰ開発プロジェクトが開発する成果物量を削減できるソフトウェア資産を構築，(b) ソフトウェア資産を長期間，全社展開することでコスト削減効果を最大化することである．

4. SPL 導入の課題

4.1. エンタプライズ・システムへの SPL 適用の課題

文献 [6],[9] ではエンタプライズ・システム開発における SPL 適用の課題が述べられている．ここでは当組織の状況を説明する．

(1-1) 移り変わる実装技術と非機能要件の高度化

ソフトウェア資産を構築する上で資産が永続的に利用できることは重要な要素である．当組織では OS，ミドルウェアに OSS を積極的に採用することで，ベンダーの事業戦略（事業撤退を含む）の影響を受けにくい環境を整えている．また，開発言語は複数のベンダーが提供している Java を採用しており，長期間の利用が期待できる状態となっている．また，操作性やリアルタイム性，24 時間稼働などの非機能要件は製造業ということもあり，大きな問題となっていない．全社員が利用する勤務管理システムでは利用者が数千名，同時利用は数百名の規模であり，事業部ごとに開発するシステムでは同時利用者は 100 名以下であることが多い．

従って，一般の PC サーバーの能力で処理可能な負荷である．

(1-2) RDBMS への依存度

一般に，SQL による RDBMS の処理がボトルネックになることが指摘されている．当組織では同時利用者数が少ないことに加え，サーバー能力が不足気味であった 1990 年代から，正規化されたテーブル構造を実装した上で十分な応答速度を確保するチューニング技術を蓄積しているため，SQL をそのまま利用しても問題が発生していない．

(1-3) 個別案件主体とレガシーシステムの存在

案件単位で利用する技術やコスト，納期の制約があり，SPL 適用の障害になることが示されている．当組織では新技術を評価し，社内展開する部署が設置されているため，案件単位で個別に技術を選定することがなく，SPL を導入しやすい環境となっている．

(1-4) 工数削減できるソフトウェア資産の構築

文献 [9] の事例では SPL の取り組みが初期段階で充分な結果が得られていない．よりコスト削減効果があるソフトウェア資産の構築が求められている．

4.2. SPL 導入の組込み系との共通課題

文献 [10] では組込み系，エンタプライズ系共通の課題として以下のものが示されている．

(2-1) 品質管理のためのソフトウェア構成管理

開発したソフトウェア資産はＡＰ開発で利用されたあとも機能拡張や欠陥除去が行われ，複数バージョンの資産が開発される．欠陥除去は配付されたすべてのバージョンに対して実施されるべきであるが，構成管理をうまく行わないと局所的にしか欠陥除去が行われない可能性がある．

(2-2) 再利用に対する人間的側面

NIH(Not Invented Here) は自分達が開発したものでないと再利用したがらない傾向を表す言葉であるが，NIH 症候群を考慮したプロジェクト運営が必要である。

5. SPL 推進方針

4. で示した (1-1), (1-2), (1-3) の課題は既に解消していたため，(1-4), (2-1), (2-2) の課題に取り組んだ．

5.1. 開発量が削減できるソフトウェア資産の開発

ソフトウェアの再利用により開発コストを削減するためにはＡＰ開発において開発するソースコード量が削減できるソフトウェア資産が必要となる．文献 [6][9] の例ではユーザーインターフェースではなく，ビジネスロジックに焦点を当てている．一方，我々の Web システム構築の経験ではビジネスロジックよりも画面出力に関するソースコードの方が多いことがわかっている．我々は画面部品を開発することで (1-4) の課題を克服することにした．しかし，ユーザーインターフェースである画面は利用者の要求により多くのバリエーションがあり，部品化が難しい．また，部品化により画面デザインの自由度が低下すると，操作性の悪化により利用者の満足度が低下する恐れもある．画面部品の開発にあたっては操作性のよい画面が作成できる機能を持たせ，部品化により利用者の満足度を向上させることにした．

5.2. ソフトウェア資産の展開

(2-1),(2-2) はソフトウェア資産展開の課題と位置付けた．開発したソフトウェア資産は全社に展開し，長期間利用することでコスト削減効果を最大化したい．また，ソフトウェア資産の欠陥除去などの保守コストも抑制したい．機能拡張により基本アーキテクチャが変更すると欠陥除去のための調査コストやソースコードの修正コストが増加することが予想されるため，ドメイン要求分析を実施しすることで安定したアーキテクチャを構築することにした．その上で (2-1) の品質管理の課題をドメイン開発チーム内の構成管理で解決する体制を検討する．また，(2-2) の人的側面の課題は開発者が積極的に使いたいと思うよう，利用者の満足度を高められる操作性の高い画面部品を提供することで解決することにした．さらにトレーニングなどの支援体制を整備することでソフトウェア資産が容易に利用できる環境を整えることにした．

6. ドメイン開発の実践

6.1. ドメイン要求開発

ドメイン要求開発ではソフトウェア部品を抽出し，安定したアーキテクチャを構築するために想定されているアプリケーションの固定化できる共通点と個別に変更できる可変点を抽出する．

(1) 共通要件の抽出

想定するドメインは企業内の事務処理システムであり，生産管理，在庫管理，販売管理，購買管理，物流管理，会計，人事といった幅広い業務を支援するソフトウェアである．図 1 に事務処理システムの構造を示す．サーバー内にデータベースを構築し，正規化されたテーブル構造が実装されている．端末には Web ブラウザがインストールされており，サーバー内のプログラムにアクセスすることで業務を行う．プログラムは注文登録や出荷指示など，１つの業務に対応したもので複数の画面で構成されている．１つのシステムが対応する業務は 30 ～ 50 種類程度のものが多い．

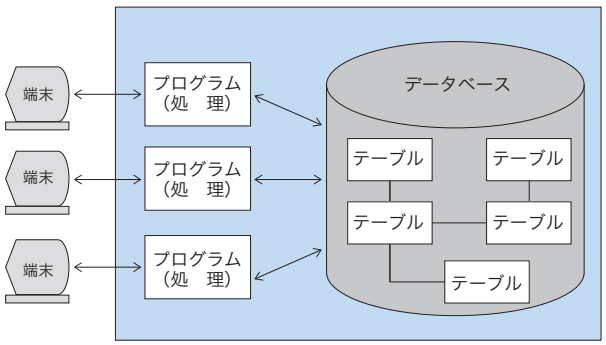


図１　事務処理システムの構造

上記構造を前提とし，以下の共通的な要件が存在している．

(R1) エンドユーザーはブラウザでシステムにアクセスし，業務で利用する画面を表示する．

(R2) エンドユーザーは画面を操作して，データベースにデータを登録，照会，変更，削除する．

(R3) その際，システムは入力された値のエラーチェックを行う．

(R4) システムは必要に応じて，データベースのデータを加工してブラウザに表示する．

(R5) システムは必要に応じて，入力されたデータを加工してデータベースに登録，変更する．

(R6) システムは処理結果を端末に表示する．

(2) 画面の共通点と可変点の抽出

図２，図３に業務で使用する画面の例を示す．図２は商品を登録する画面である．図３は商品の注文を登録する画面である．この２つの画面の共通点は，(a) タイトル，(b) メニュー，(c)1 つ以上のデータ入力ブロック (図３では (c-1), (c-2))，(d) 登録ボタンが配置されていることである．可変点は，共通点の中に多く含まれており，以下のものが抽出できる．

- タイトルに表示されている文字が “ 商品登録 ” と “ 注文受付 ” で異なる．
- データ入力項目は図２では１ブロック，図３では２ブロックで構成されている．
- データ入力ブロックに含まれるデータ項目（商品コード，受注番号等）が異なる．

部品化の課題となるのはデータ入力項目であり，可変点は以下の通りである．

- データ項目の名称
- データ入力領域の桁数
- データ入力の方法（TEXT, CHECKBOX 等）

(3) 画面遷移の共通点と可変点の抽出

画面遷移についても再利用を行うため，共通点と可変点を抽出した．基本的な画面遷移を図４に示す．利用者がアプリケーションのメニューからプログラムを選択すると，プログラム内のメニューから (a) 登録入力画面または (d) 検索条件入力画面を表示することができる．その後は矢印で示された流れで画面を進めることができ，登録，照会，変更，削除の業務が行えるようになっている．この画面遷移で可変点は，(h) 変更確認画面，(k) 削除確認画面で，アプリケーション要求に応じて省略することができる．

6.2. ドメイン設計

ドメイン設計ではＡＰ開発で使用するアーキテクチャを設計する．今回のソフトウェア資産は Web システムを前提としており，端末とサーバーとの通信が毎回切断されるという条件の中で可変点に対応できる構造にする必要がある．図５に利用者が画面に入力したデータをデータベース

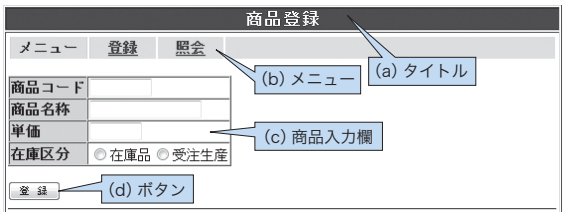


図２．商品登録画面の例

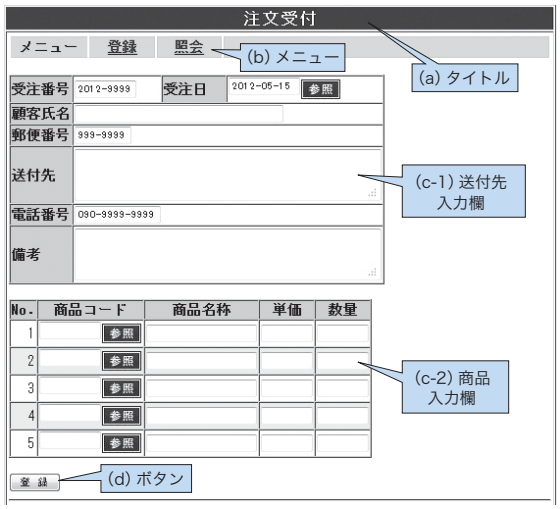


図３．注文受付画面の例

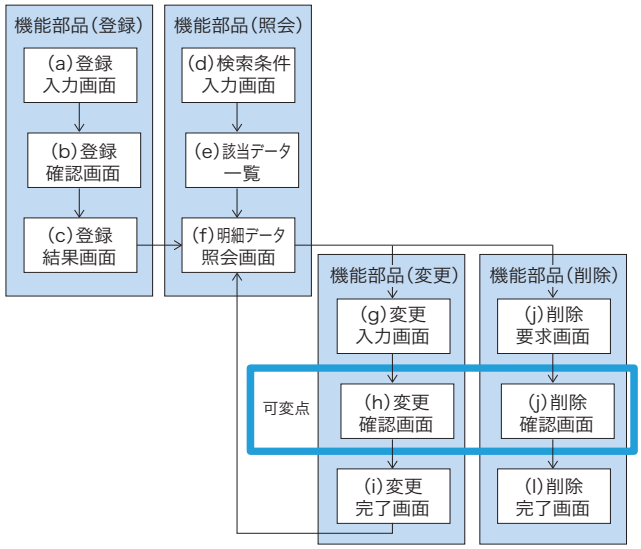


図４．画面遷移の共通化

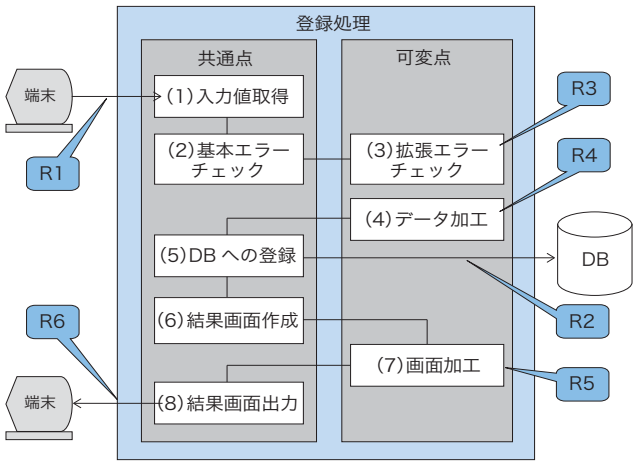


図５．データをデータベースに登録する処理の流れ

```
<form action="…">
<table>
<tr><th> 商品コード </th>
<td><input type="text" name="prod_cd" size="8"></td></tr>
<tr><th> 数量 </th>
<td><input type="text" name="order_qty" size="5"></td></tr>
</table>
<input type="submit" value=" 登 録 ">
</form>
```

図 6．受注品目 , 受注数量入力画面の HTML

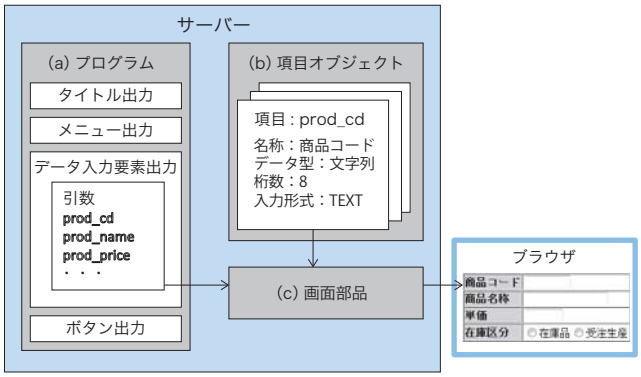


図 7．画面部品の構成

に登録する際の処理の流れを示す．この流れは 6.1 の要件 (R1) から (R6) に対応している．端末の登録ボタンを押すと，サーバー内の登録処理が起動される．まず，共通部の (1) 入力値取得は端末から送信されたデータをデータ項目ごとに分解し，変数に保管する．次に，(2) 基本エラーチェックでは変数に格納されたデータが予め定義されたデータ型（文字型，数値型，日付型等）に合っているか，日付の場合は閏年を考慮して存在する日かなどのエラーチェックを実施する．その後，アプリケーションが追加のエラーチェックを要求されているのであれば，(3) 拡張エラーチェックを実行し，要求に対応する．この様なアーキテクチャを作成し，(R1),(R2),(R6) を共通点とし，(R3),(R4),(R5) は可変点として処理が追加できるようにした．

6.3. ドメイン実現

ドメイン実現では，ソフトウェア資産の詳細設計と実装を行う．エンタプライズ・システムでのソフトウェア資産実装の課題は対象となる事業部や対象業務の違いで使用しているデータ項目が異なることである．画面はデータベースの項目を表示するため，同じ注文入力の画面でも事業部ごとにデータ項目が異なり，再利用が難しくなる．同様にデータベースとの入出力を行うデータ項目も異なるためそのままではソフトウェア資産の再利用ができない．

6.3.1. 画面に関する可変点抽出と抽象化

ブラウザにデータの入力画面を表示するための HTML を図 6 に示す．この画面では，受注品目と受注数量の入力項目のブロックが 1 つずつ表示される．この HTML を出力する再利用可能なプログラムを考える場合，下線を引いた部分が可変点となり，残りの部分は共通点である．従って，共通点の方が可変点より多く，再利用の効果が期待できる．ＡＰ開発者がソフトウェア資産に追加する情報は，画面に表示し利用者が認識するためのデータラベル，プログラム内で処理するためのデータ識別子（当組織のルールでは英数字），桁数の 3 種類が必要となり，具体的には “ 商品コード , prod_cd, 8 桁 ”, “ 数量 , order_qty, 5 桁 ” の 6 項目設定が必要である．このような単一の単純な画面では画面表示機能の部品化は簡単である．しかし，我々の組織で標準的なシステムでは 200 以上の画面があり，それぞれ平均 5 つのデータ項目が使用されているとすれば合計 3000 件 (5 項目× 3 種類× 200 画面) の定義が必要となる．また，実際のシステムの画面はもう少し複雑で，図 2 に示したように RADIO ボタンで選択できたり，プルダウン形式で値を選択できたりするものもある．その選択肢も画面ごとに設定する必要があり，再利用のための作業が増加する．

画面表示の機能を部品化するためには，ＡＰ開発チームの作業量を削減する必要がある．図 7 にこの課題を解決するための仕組みを示す．商品コードなどのデータ項目は，データベース設計時に項目名称や実装用の識別子，桁数などが設計されているため，ＡＰ開発チームは，各画面を設計・実装する際にデータベースの設計情報を参照している．図 7 の (b) 項目オブジェクトはデータベース設計で設計された設計情報をサーバー上のメモリにオブジェクトとして実装したものである．例えば，商品コードは商品登録画面や注文入力画面でも通常同じデータラベル，同じ桁数となるため画面ごとに設定する必要はない．図 7 で今回作成する (a) プログラムには画面に出力したいデータ項目の実装用識別子 prod_cd を設定する．利用者がブラウザに画面を表示する際，prod_cd をキーとしてメモリ中の項目オブジェクトを検索し，項目オブジェクトからデータラベル，入力桁数を取り出すことで画面を表示することができる．更に，項目オブジェクトの入力形式に RADIO ボタンを指定し，選択肢を登録しておけば，利用者が RADIO ボタンで入力できる画面を出力することができる．項目オブジェクトに必要な基本情報はデータベースの設計情報から自動生成可能であるため，ＡＰ開発者は必要に応じて追加情報を登録するだけでよい．また，プログラム開発では，データラベルや桁数を気にすることなく，表示したいデータ項目の実装用識別子を指定するだけで画面部品の再利用が簡単にできる．図 8 に項目オブジェクトを利用する主な画面部品を示す．部品は入力画面と表示画面に分類でき，更にデータを 1 件表示するものと複数件のデータをリスト表示するものに分類できるため，合計 4 種類できる．その他，マトリックス形式で表示する部品も存在するが，ごく一部の機能でのみ利用されている．

項目オブジェクトの考案により画面部品を実現することができた．業務で利用する画面はこれらの部品を組み合わせることで開発することができるが，これらの部品を組み合わせて図 3 に示したような画面全体を中間部品として開発できるようになった．さらにこの中間部品を使って図 4 の画面遷移や図 5 の共通点の部分も部品化できるようにな

りＡＰ開発チームが作成するソースコード量の削減が期待できるソフトウェア資産が構築できた．

6.3.2.　画面部品の高機能化

画面部品の開発は開発工数削減が期待できる一方で画面デザインの制約となる．制約が強ければ利用者の満足度を低下させたり，開発者が再利用に対してネガティブな印象を持ったりすることになる．再利用を全社展開するためには使い勝手の良い画面を従来よりも少ない工数で開発できるようにする必要がある．

Web ブラウザは本来閲覧用のソフトウェアであり，データ登録作業を効率的に行えるように設計されていない．そのためＡＰ開発者は JavaScript などでプログラムを作成して操作性を改善する必要がある．今回開発した画面部品では図９に示すようにカーソルキーやリターンキーで項目移動ができるようになっている．このような機能はマルチブラウザ対応にする必要があり，セキュリティ面の配慮も必要となる．また，ブラウザのバージョンアップにも継続的に対応する必要がある．このような継続的に保守が必要な高機能部品をドメイン開発チームが品質保証して提供することで，ＡＰ開発チームに対してソフトウェア資産を活用した方が低コストでよいシステムが開発できるという動機づけを行っている．

6.3.3.　構成管理

開発したソフトウェア資産はソースコードでＡＰ開発チームに提供し，ＡＰ開発チームでカスタマイズする方法とドメイン開発チームが機能追加してバイナリ形式のライブラリで提供する方法が考えられる．我々は後者を採用した．その理由は欠陥除去すべきバージョンの特定作業が容易で品質保証の点で優れており、新機能を全社展開するのにも好都合であるからである．一方，複数のＡＰ開発が並行して進行している状況ではドメイン開発チームはタイムリーに新機能の提供や欠陥除去が行える高い開発能力が要求されるという制約がある．ドメイン開発チームはリリースしたソフトウェア資産の各バージョンを一元管理し，欠陥が発見された場合，各ＡＰ開発チームが利用しているバージョンのソフトウェア資産に対して欠陥除去を行い，新しいライブラリを提供する．各ＡＰ開発者はライブラリ間の互換性を心配することなく開発作業を継続することができる．

6.4.　ドメイン試験

ドメイン試験では，ドメイン開発で作成した成果物の試験を行う．また，ドメイン試験で開発した成果物をＡＰ開発チームに提供する．当組織でのドメイン試験の課題は品質保証であった．ソフトウェア資産はＡＰ開発チームのニーズに合わせて，短い周期で新機能のリリースを行っている．この際，他の機能への悪影響があれば，ＡＰ開発チームの開発効率を悪化させる恐れや，本番稼働中のシステムのトラブルを引き起こす危険があり，高い品質管理が求められる．日々増加するソフトウェア資産を人手でテストすることはできないため，自動テストツールを活用したテストを

(1) データ入力部品（1 件）

受注番号	2012-9999	受注日	2012-05-15 参照
顧客氏名			
郵便番号	999-9999		
送付先			
電話番号	090-9999-9999		
備考			

(2) データ入力部品（複数件）

No.	商品コード	商品名称	単価	数量
1	参照			
2	参照			
3	参照			

(3) データ表示部品（1 件）

受注番号	2012-9999	受注日	2012-05-15
顧客氏名	住友　太郎		
郵便番号	999-9999		
送付先	大阪市中央区北浜1-1		
電話番号	090-9999-9999		
備考			

(4) データ表示部品（複数件）

No.	商品コード	商品名称	単価	数量
1	LD10L	LED電球 電球色 850lm	3,000	2
2	LD10N	LED電球 昼白色 1000lm	3,500	3

図８．主な画面部品

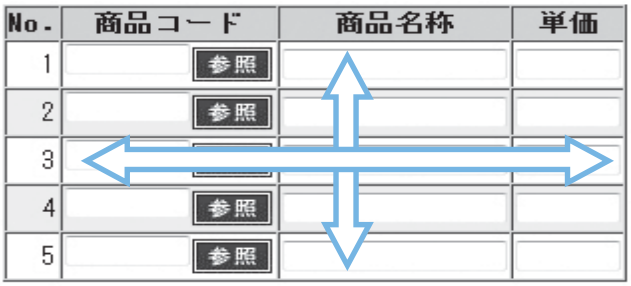


図９．カーソル操作

行っている．作成したテストシナリオは約 1000 種類あり，テスト項目数は約 10000 である．ＡＰ開発チームへリリースする際，一晩かけて自動テストすることで互換性が確保されていることを確認し，不具合流出を防止している．なお，ＡＰ開発チームへの試験成果物の提供はできていない．

7.　アプリケーション開発の実践

7.1.　トレーニング

演習を中心とした３日間のトレーニング・コースを開発した．定員は 10 名で月１回の定期開催に加え，プロジェクトの状況に応じて追加開催した．新入社員は入社後の新人研修で全員がトレーニングを受け，研修課題のプログラムを数本開発する．現在は開発者全員がトレーニングを受けた状態で開発を行っており，定期開催のトレーニングは協力会社から新たに開発に加わる開発者向けに行われている．

7.2. アプリケーション要求開発

アプリケーション要求開発では，ソフトウェア資産を活用し，個別アプリケーションの要求を開発する．事務処理システムでは，ユーザーインターフェースである画面・帳票や，データベース更新時の計算ロジックなどを明確にし，利用部門と合意する．ソフトウェアの再利用を効果的に行うためにはこの段階で再利用可能なソフトウェアで実現可能な仕様になっている必要がある．今回の取り組みでは画面部品が再利用の対象となっているため，画面設計が再利用の度合いを決めるポイントとなる．ソフトウェア資産を活用した実装経験がある開発者が要求開発を行うのが望ましいが，最初の３年間は実装経験のない開発者が要件定義を行うケースも多いため，ドメイン開発チームのメンバーが設計された画面をレビューし，再利用ができるよう指導していた．現在では要求開発の担当者が入社後に実装を経験しているケースが多く，望ましい状態に近づいている．

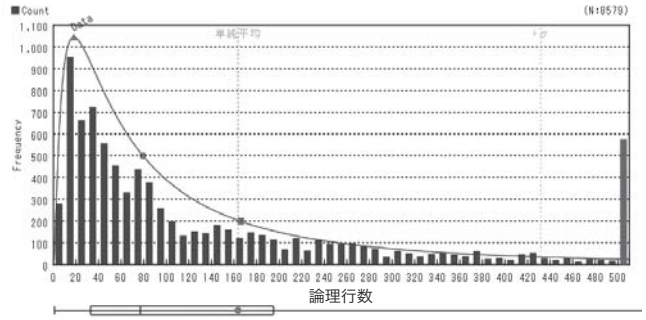


図 10. 作成したプログラムのライン数の分布

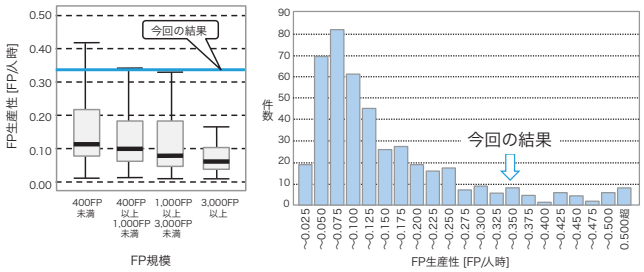


図 11. 生産性のベンチマーク結果との比較

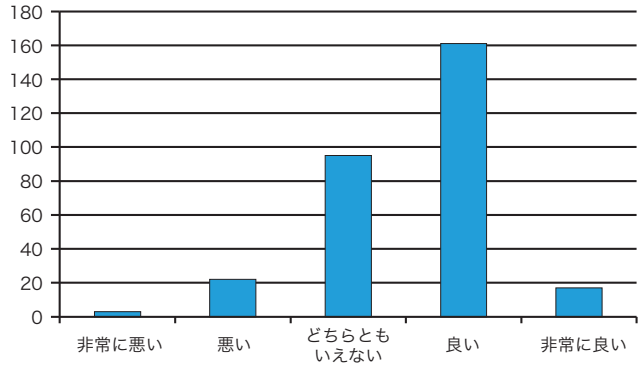


図 12. 操作性に関するアンケート結果

7.3. アプリケーション実現

アプリケーション実現では，アプリケーションの詳細設計と実装を行う．当組織ではソフトウェア資産の可変点の設定は XML 形式のファイルで行い，ソフトウェア資産で実現できない機能は Java でコーディングする．作成したコードはプラグイン方式で既存のソフトウェア資産から呼び出され，要求にあった機能が実現できるようになっている．また，ソフトウェア部品は体系化されており，開発者はネーミングルールにより必要な部品が簡単に特定できるようになっている．

8. 評価

ここでは 5 章の SPL 推進方針で示した課題と解決策の評価を行う．ソフトウェア部品の開発は 1999 年より行っているが 2003 年に部品化の範囲を拡大したため評価対象は 2003 ～ 2012 年の 10 年間で開発した約 300 の業務システムとする．

8.1. 構築したソフトウェア資産の評価

今回の取り組みでは再利用性の高いソフトウェア部品を構築することで，ＡＰ開発チームが開発するソースコードの量を削減し，開発コストを削減することが目的であった．

(1) 開発コード量削減の評価

開発したソフトウェア部品によるコードの削減量を評価するため，これまでにソフトウェア資産を再利用して開発したプログラム 13,174 本を調査した．その結果，4,595 本 (34.9%) のプログラムは，既存の可変点に対する設定のみで実現できていた．残りの 8,579 本 (65.1%) は可変点に対して追加コーディングを行っている．追加コーディングを行ったソースコードのライン数の分布を図 10 に示す．ライン数は空白行やコメントを取り除いた論理行数である．平均値は 164 行であり，中央値は 77 行であった．文献 [11] のデータを利用して，機能当たりのライン数を計算すると 2084 行となり，ソフトウェア資産の再利用によりコード量が約 92% 削減されており，狙い通りのソフトウェア部品が構築できたと考えられる．

(2) 開発生産性の評価

コスト削減の成果を評価するため，文献 [7] のデータと比較する．今回のソフトウェア資産を活用した開発での開発生産性は 0.33FP[※1]/ 人時であった．文献 [7] に掲載されている FP 生産性に今回の結果を加えたものを図 11 に示す．今回開発したシステムの規模は 1500 ～ 5000FP のものが多く，左側の箱ひげ図で見ると一般的な開発の中央値に比べ３～５倍程度の生産性が実現できている．右側の分布を見ても比較的高い生産性が達成できていることが分かる．ただし，生産性は要求される品質に応じて再利用とは無関

【脚注】

※１　FP: ファンクションポイント [12]

係にテスト工数が増加することも考えられるため，この評価結果は参考値と考えるべきである．

(3) 利用者満足度の評価

今回の取り組みでは操作性の高い画面が実現できる高機能な画面部品を提供することで NIH 症候群を回避することが 1 つの対策であった．部品機能の強化による利用者の満足度を評価するためにシステム稼働してから 3 ヶ月後に実施している利用者向けアンケート調査の結果を集計した．その結果を図 12 に示す．有効回答は 298 件であった．縦軸は回答数である．“ 良い ”，“ 非常に良い ” という回答が 188 件 (63%) あり，利用者のニーズに応えられていると考えられる．

8.2. ソフトウェア資産展開の評価

開発したソフトウェア資産は 10 年間社内の全開発プロジェクトで再利用されており，当初の狙い通り展開できている．ソフトウェア資産展開の課題は品質保証のための構成管理と NIH 症候群の回避であった．発見された欠陥の除去はドメイン開発チームが一括して行っているため特に問題は発生してない．また，ＡＰ開発チームが独自に開発すると手間がかかる高機能部品を提供することで NIH 症候群も回避でき積極的にソフトウェア資産を活用する取り組みが進んでいる．

9. 考察

9.1. ソフトウェア資産構築に関する考察

今回調査した 13,174 本のプログラムでは 318 個のソフトウェア部品が延べ 28,444 回再利用されている．再利用された部品の利用頻度を図 13 に示す．横軸は部品を示し，縦軸に利用割合と累積値を示している．最も利用回数の多かった機能部品は 1,929 回使用され，全体の 6.78% を占めていた．上位 15 個の部品で 50.4%，34 個で 70.5%，55 個で 80.3%，120 個で 90.0%，200 個で 92.9%，318 個で 93.8% であった．200 位以下の部品の使用率は 0.02% 以下と低い値であった．当組織では部品が必要になる前に網羅的に部品を提供する戦略をとったが投資効率の観点では開発するソフトウェア資産の範囲は利用頻度の高いものに絞り込む戦略も考えられる．

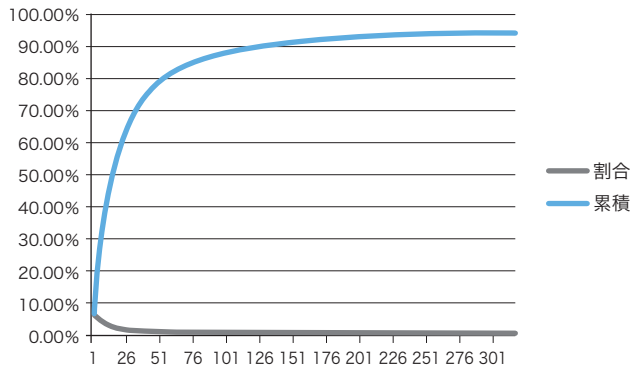


図 13. 部品の利用頻度

9.2. 継続的なソフトウェア資産の機能拡張

SPL ではドメイン開発チームとＡＰ開発チームを分離し，協業することが望ましいとされている [2]．当組織では操作性の高い部品が提供できたこともあり，各ＡＰ開発チームは汎用的な部品を独自に開発せず，ドメイン開発チームに依頼するようになった．要件定義や外部設計の段階からドメイン開発チームに機能追加の相談があり，ＡＰ開発チームと協業しながらソフトウェア資産の拡張と他プロジェクトへの展開ができる体制になっている．開発者にとって魅力あるソフトウェア部品の提供がソフトウェア資産の拡張・展開サイクルがうまく回る重要な要因だと考えられる。

10. まとめ

本論文では SPL のエンタプライズ・システムへの適用事例を示した．ビジネスロジックではなく画面部品を開発することにより開発する成果物量を削減することで開発コストの削減が実現できている．また，画面部品の機能を強化することで利用者の満足度も向上することができた．ただし，今回の取り組みではＡＰ開発のプログラム開発工程のみが対象であった，今後の課題は設計や試験工程に対するソフトウェア資産を構築し，更にコスト削減を図ることである．

【参考文献】

[1] William B. Frakes and Kyo Kang, “Software Reuse Research: Status and Future”, IEEE Transactions on Software Engineering, Vol.31, No.7, pp. 529-536, Jul 2005.

[2] Software Engineering Institute, "A Framework for Software Product Line Practice, Version 5.0”, http://www.sei.cmu.edu/productlines/frame_report/index.html, 参照 2013.4.1.

[3] Klaus Pohl, Guenter Boeckle, Frank J. van der Linden, "Software Product Line Engineering: Foundations, Principles and Techniques", Springer, 2005. (邦訳 " ソフトウェアプロダクトラインエンジニアリング―ソフトウェア製品系列開発の基礎と概念から技法まで ")

[4] Kentaro Yoshimura, Dharmalingam Ganesan, and Dirk Muthig, “Defining a strategy to introduce software product line using the existing embedded systems”, Proc. of 6th ACM & IEEE International Conference on Embedded Software, pp.63-72, Oct 2006.

[5] Kentaro Yoshimura, Fumio Narisawa, Koji Hashimoto, and Tohru Kikuno, “Factor analysis based approach for detecting product line variability from change history”, Proc. of MSR 2008, pp.11-18, May 2008.

[6] 石田 裕三 , " エンタープライズ・システムにおけるソフトウェアプロダクトラインの適用 ", 情報処理 , Vol.50, No.4, pp.303-310, Aprl 2009.

[7] 情報処理推進機構 (IPA) ソフトウェア・エンジニアリング・センター (SEC), “ ソウトウェア開発データ白書 2010-2011, p.234,” 情報処理推進機構 , 東京 , 2012.

[8] Linda Northrop, "Software Product Lines: Reuse That Makes Business Sense", ASWEC2006, http://www.sei.cmu.edu/library/assets/ASWEC2006.pdf, 参照 2013.4.1

[9] Yuzo ISHIDA, “Challenge for the SPL Approach in Enterprise Software Development,” NRI Information Technology Report 2007, Vol.8, pp.1-11, 2007.

[10] 野中 誠 , " ソフトウェアプロダクトライン開発のマネジメント：課題と技法 ", 情報処理 , Vol.50, No.4, pp.289-294, Aprl 2009.

[11] 日本情報システム・ユーザー協会 , “ ユーザー企業ソフトウェアメトリックス調査 2011”, 日本情報システム・ユーザー協会 , 2011.

[12] A. J. Albrecht, "Function point analysis", Encyclopedia of Software Engineering, Vol.1, pp.518–524, 1994.

制御システムセキュリティセンター活動紹介

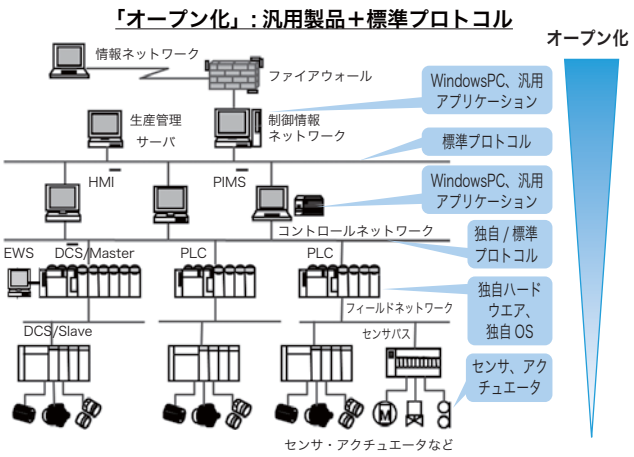
～セキュアな制御システムを世界へ未来へ～

技術研究組合制御システムセキュリティセンター（CSSC）
専務理事 研究開発部長 CSSC認証ラボラトリー長
小林　偉昭

1 はじめに

制御システムは、最近では Windows や UNIX などの汎用 OS の採用、さらに Ethernet や TCP/IP の標準プロトコルが採用されている。このため情報システムで起きているセキュリティの脅威（サイバー攻撃）が増大してきている。図 1 に示すような汎用製品と標準プロトコル採用によるオープン化が進展しているという認識を持つことが大切である。プラント設備（生産ライン制御等）におけるオープン化の割合は、外部ネットワークとの接続が 36.8%、設備内の OS の利用状況として Windows が 88.9%、UNIX 系が 13.7% と報告されている。

制御システムへのサイバー攻撃の衝撃的な事例が、2010 年イラン核施設への Stuxnet である。制御システムのセキュリティの安全神話は崩壊してしまった。米国の国土安全保障省（DHS）の ICS-CERT※1 からのレポートによると、米国における重要インフラ事業者に対する攻撃のインシデント報告件数は、図 2 に示すように 2010 年が 34 件であったのが、2013 年には 257 件と約 8 倍となっている。

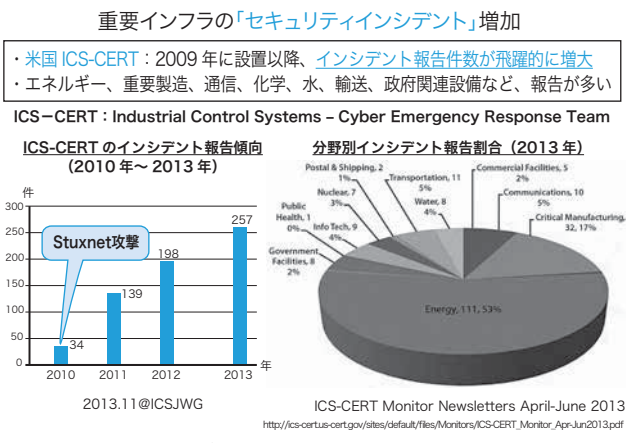


■ 例：プラント設備（生産ライン制御等）におけるオープン化の割合

- 外部ネットワークとの接続　36.8%
- 設備内の OS の利用状況　Windows：88.9% UNIX 系：13.7%

経済産業省サイバーセキュリティと経済 研究会中間とりまとめ（案）：
http://www.meti.go.jp/committee/kenkyukai/shoujo/cyber_security/report01.html

図 1　制御システムの状況



ICS-CERT ウェブサイト、ICSJWG2013Fall 情報をもとに作成

図 2　重要インフラのセキュリティインシデント発生傾向（米国 ICS-CERT）

2 CSSC の設立経緯と概要

2.1　CSSC の設立経緯

2010 年の Stuxnet によるイラン核施設へのサイバー攻撃を受けて、経済産業省は 2010 年 12 月に「サイバーセキュリティと経済研究会」を立ち上げた。研究会は、サイバー攻撃に対する情報共有、制御システムのセキュリティ確保及び人材育成が、これからのセキュアな社会インフラを守るためには必要だと提言した。これを受けて、2011 年 10 月に「制御システムセキュリティ検討タスクフォース」を立ち上げた。ここでの目標の一つが、日本の社会インフラのセキュリティ確保である。もう一つは、ベンダが製品を輸出するときに、国際標準に対応したセキュリティを実装し、競合力を高めることである。

【脚注】

※1　ICS-CERT：米国国土安全保障省（DHS）が運営する制御システムに特化したインシデント対応機関。制御システムに関する国内のインシデント報告を受け、専門家による分析・対応サービスを提供する。
http://www.us-cert.gov/control_systems/ics-cert/

このタスクフォースの検討を受け、2012 年 3 月に民間企業の技術研究組合として「制御システムセキュリティセンター」が発足した。2012 年 7 月にお台場に東京研究センターを、2013 年 5 月に宮城県多賀城市に東北多賀城本部を設置し、「セキュアな制御システムを世界へ未来へ」という目標を掲げて活動を開始した。

東北多賀城本部にはファクトリーオートメーション FA、プロセスオートメーション PA、ビルオートメーション BA などの 7 つの制御システムの模擬プラントを設置し、これらのテストベッドを中心に、制御システム向けの高セキュリティ機能や評価認証技術を研究・開発し、普及啓発コンテンツを整備している。この経緯を図 3 に示す。

2.2　CSSC の概要

CSSC の組織と概要を図 4、図 5 に示す。運営委員会のもとに 4 つの委員会を設置し、下記の活動を進めている。

- 制御システムにおける可用性を高める高セキュア化技術の研究開発
- 広域連携システム（スマートコミュニティ等）における高セキュアシステム技術の研究開発
- システムセキュリティ検証・認証技術の研究開発と認証実証事業への展開
- 国際標準化と国際連携
- 制御セキュリティテストベッドの研究開発と人材育成や普及啓発への展開
- サイバーセキュリティ事業を震災復興、減災に展開

さらに、2013 年 8 月には CSSC 内に CSSC 認証ラボラトリーを新設し、2013 年 9 月に日本適合性認定協会 JAB に認証機関として認定申請し、2013 年 11 月には ISCI（ISA Security Compliance Institute）へ加入し、制御機器の認証実証事業を推進している。

グローバルな社会インフラの制御システムセキュリティ向上に貢献していくために、米国の国土安全保障省 DHS や欧州の ENCS（European Network for Cyber Security）などと国際連携を進めている。

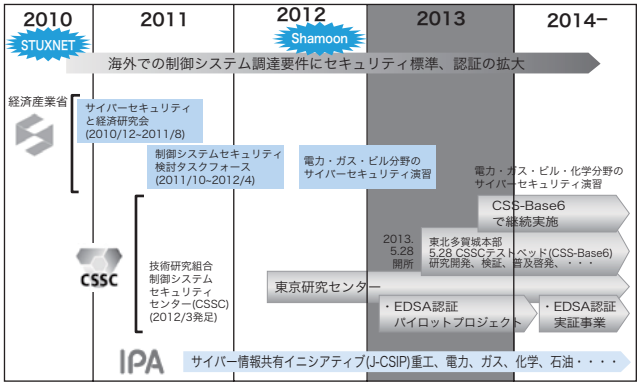


図 3　制御システムセキュリティへの日本の取り組み状況

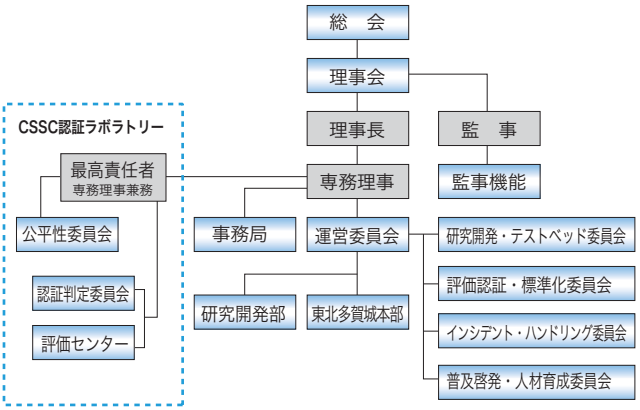


図 4　CSSC の組織体制

CSSC の概要			
名称	技術研究組合 制御システムセキュリティセンター （英文名）Control System Security Center （略称）CSSC	組合員 （50 音順）	全 23 社（2013 年 12 月現在）　　*：創設時メンバー 8 社 アズビル株式会社 *、エヌ・アール・アイ・セキュアテクノロジーズ株式会社、エヌ・ティ・ティ・コミュニケーションズ株式会社、オムロン株式会社、独立行政法人産業技術総合研究所 *、独立行政法人情報処理推進機構、国立大学法人電気通信大学、株式会社東芝 *、東北インフォメーション・システムズ株式会社、株式会社トヨタ IT 開発センター、トレンドマイクロ株式会社、日本電気株式会社、一般財団法人 日本品質保証機構、株式会社日立製作所 *、富士通株式会社、富士電機株式会社、マカフィー株式会社、三菱重工業株式会社 *、株式会社三菱総合研究所 *、三菱電機株式会社、森ビル株式会社 *、横河電機株式会社 *、株式会社ラック
	※経済産業大臣認可法人		
設立日	2012 年 3 月 6 日（登録完了日）		
所在地	【東北多賀城本部 (TTHQ)】 宮城県多賀城市桜木 3-4-1 （みやぎ復興パーク F-21 棟 6 階）		
	【東京研究センター（TRC）】 東京都江東区青海 2-4-7 （独立行政法人産業技術総合研究所 臨海副都心センター別館 8 階）	連携団体 （予定含む）	一般社団法人 JPCERT コーディネーションセンター、一般社団法人日本電機工業会、公益社団法人計測自動制御学会、一般社団法人電子技術情報産業協会、一般社団法人日本電気計測器工業会、一般財団法人製造科学技術センター、電気事業連合会、一般社団法人日本ガス協会、一般社団法人日本化学工業協会

図 5　CSSC の概要

3 CSSC の活動

研究開発と評価認証・検証技術について紹介する。

3.1 研究開発

2012 年度には、CSS-Base6 にセキュリティでの攻撃（サイバー攻撃）が発生した時、臨場感を持った疑似体験ができるよう研究・開発し、PA、FA、BA や広域連携システム及び電力事業者やガス事業者向けの７つの模擬プラントを構築した。2013 年度は、この７つの模擬プラントを活用して経営者向けの啓蒙活動を実施している。

①排水・下水用 PA 用模擬プラント（写真 1）

圧力、流量、温度などのプロセス量をバルブや電磁流量計等で管理するもので、このプラントでは沈殿槽を模擬した汚水排水設備でのサイバー攻撃が体験できる。

②化学用 PA 用模擬プラント

流量調節弁などを制御する水槽の水位レベル制御設備でサイバー攻撃が体験できる。

③ BA 用模擬プラント

ビル全体の温度管理、エレベータ管理、空調管理、照明管理など多くの管理システムが防災センターで監視されている。この模擬プラントでは照明管理の制御のサイバー攻撃が体験できる。

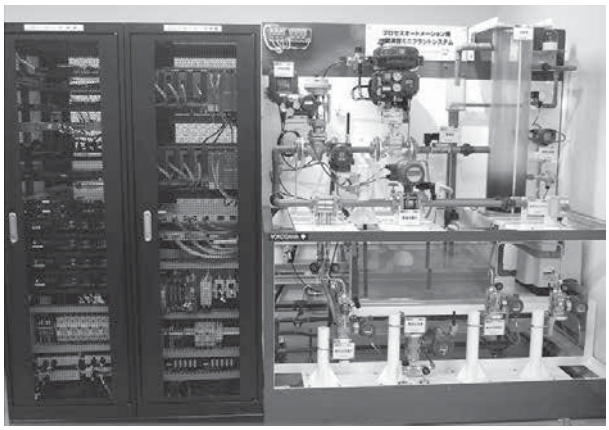


写真１　排水・下水用 PA 用模擬プラント

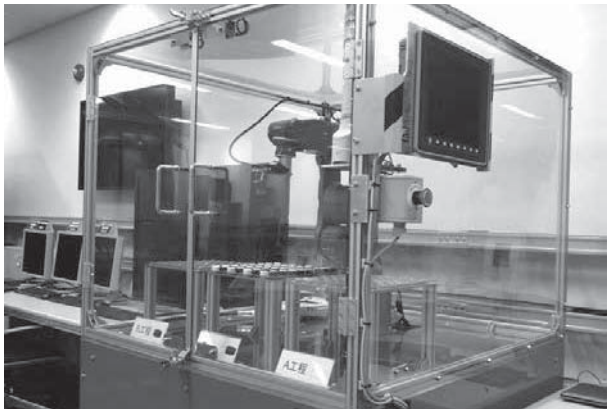


写真２　FA 用模擬プラント

④ FA 用模擬プラント（写真 2）

自動車工場の生産現場を模擬し、部品供給ロボット制御を使用したサイバー攻撃が体験できる。

⑤火力発電所用模擬プラント

火力発電所の管理用シミュレータを利用してサイバー攻撃発生時の体験ができる。

⑥ガス事業者用模擬プラント

エアタンク圧力一定装置を利用して、ガス事業者向けの各種サイバー攻撃が体験できる。

⑦広域連携用模擬プラント

スマートグリッドやスマートコミュニティを構成する配電設備へのサイバー攻撃が体験できる。

2014 年度には、2013 年度の研究開発の成果を反映することにより、これらの模擬プラントを利用してサイバー攻撃への対策を検討することも可能にする予定である。

テストベッドの設計・研究・開発に加え、システムを止めない、暴走させないための可用性を高める研究や万が一の場合にも最悪の事態を避けるなどの制御システムのセキュリティを高める研究を推進している。

3.2 評価認証・検証技術

製品を海外に輸出するときに、セキュリティ標準に準拠していることが国際石油メジャーからの要件となってきている。また、国内の社会インフラを構築する制御システムに対しても一定のセキュリティ標準に準拠していることが製品選択する時の基準になることが期待されている。

上記を目標に、制御システムセキュリティ標準の要件検討から図 6 に示すように当初の評価認証の対象を次のように決定し、推進している。

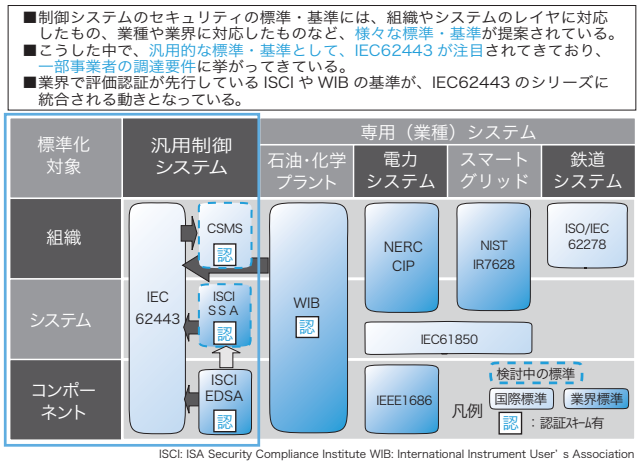


図６　制御システム分野での標準化に関する技術動向

・IEC62443（Industrial Network and System Security）を汎用的国際標準として選択
・標準確立、評価認証を一体で推進する
・IEC62443 対応の認証標準として ISCI で先行している標準に対応する

なお、業界対応に様々な標準があり、北米の電力システム用や欧米のスマートグリッド関係の標準化も進んでいることから、この分野の調査も進めている。

（1）国際標準 IEC62443 の概要

IEC62443 は、四つのレイヤからできている。

・1 番目が総論である。
・2 番目が管理・運用・プロセスの標準である。JIS Q 27001（ISO/IEC 27001）ISMS（Information Security Management System）は、情報システム向けのセキュリティマネジメントであるが、制御システム向けのセキュリティマネジメントが IEC62443-2-1 の CSMS（Cyber Security Management System）である。
・3 番目は、制御システムについてのセキュリティ標準である。
・4 番目はコンポーネントで、製品対応のセキュリティ標準である。

IEC62443 には 13 の標準がある。独立行政法人情報処理推進機構（IPA）は、既に国際標準になっている 3 つの標準を翻訳し、2012 年 10 月に日本規格協会から出版している。

（2）評価認証の推進状況

次に 3 つの評価認証への取り組みとその現状・方向性を紹介する。EDSA（Embedded Device Security Assurance）と SSA（System Security Assurance）については、ISCI が先行して認証標準を決めているが、IEC62443 に取り込まれていく予定である。図 6 参照。

＜ EDSA 認証＞

ISCI が、スキームオーナとしてグローバルな EDSA 認証を推進している。EDSA は次の 3 種類の標準で構成されている。

・CRT（Communication Robustness Testing）は、通信レベルでの評価の標準である。ファジングという手法によりランダムなデータをぶつけて、矛盾が起きないかを主に検証する。
・FSA（Functional Security Assessment）は、セキュリティの機能についての評価の標準である。
・SDSA（Software Development Security Assessment）はソフト開発プロセスの評価の標準である。

現在、米国の 2 社の四つの製品がこの認証を取っている。日本の製品ベンダが海外に行って認証を取るのではなく、日本で、日本語で認証を取りたいという要求がある。このため CSSC では、2013 年度から EDSA 認証パイロットプロジェクトを開始し、2014 年度から認証実証事業を開始し、普及を目指している。CSSC で認証を取得すると国際相互承認され、グローバルに認証された製品として認められる。

＜ SSA 認証＞

制御システムの認証を実現する SSA が現在検討されている。ISCI では 2014 年初めまでに実現しようとしている。

＜ CSMS 認証＞

日本情報経済社会推進協会（JIPDEC）が認定機関になって、情報システム向けの ISMS の活動を推進している。JIPDEC は、制御システムへ CSMS を展開するため、2013 年度に CSMS 認証パイロットプロジェクトを開始している。

（3）検証技術

制御機器に対する評価検証の技術及び評価体制を整備するため、ワーキンググループを設置し、ISCI　EDSA の 3 つの標準に対するテスト仕様書を整備した。2013 年度の EDSA 認証パイロットプロジェクトにより、更に実態に合った仕様書にしていく。

現在 CSSC は、複数の PLC や DCS などの制御システムを整備し、さらにファジングテストツール等も複数製品活用している。これらのツールは、CSS-Base6 で組合員による利用も可能にしている。

4 おわりに

発足時の組合員は 8 組織だったが、2013 年 12 月末現在 23 組織に拡大している。さらに賛助会員制も立ち上げて、研究開発成果の普及に努めている。重要インフラ事業者を始めとする制御システム関係者の CSSC への参画を期待している。また、CSSC の認証実証事業などを多賀城市をはじめとして連携を進め、復興支援にも貢献していく所存である。

今後とも CSSC は、「セキュアな制御システムを世界へ未来へ」という目標を掲げてグローバルに活動を進めていきます。

Embedded Technology 2013(ET2013)出展報告

SEC 企画グループ
荒川　明夫

IPA/SEC は、2013 年 11 月 20 日から 22 日にかけてパシフィコ横浜で開催された Embedded Technology 2013（ET2013）に出展した。同会場内アネックスホールでは、ET2013 の主催者である一般社団法人組込みシステム技術協会（JASA）と共催し、IPA セミナーを実施した。

1. 展示会概要

Embedded Technology（ET）とは、一般社団法人組込みシステム技術協会（JASA）が主催する組込みシステム技術に特化した専門展であり、組込みシステム開発にかかわる技術者や開発者向けに最新技術などの情報を発信している。

2. 出展概要

IPA/SEC では、事業成果の普及・啓発を目的として、2004 年より本展示会に出展している。本年は、「基盤技術が支える高信頼・安全 / 安心・セキュリティ」というコンセプトを掲げ、ソフトウェア・エンジニアリング関連の事業成果を更に普及・啓発するとともに信頼性の高いソフトウェアを開発するための取り組みについても紹介した。関連する事業のパネル展示やデモの実施に加えて、ブースプレゼンを行い、IPA 職員や SEC 連携委員、地域連携団体・組織からの発表を実施した。

3. IPA ブース

展示では、以下のコーナーを設置し、該当する事業成果のパネル展示や関連資料の配布を行った。

- 高信頼化
- 安全 / 安心
- セキュリティ
- 基盤技術
- 地域団体との連携

【高信頼化コーナー】では、SEC の事業紹介をはじめ、形式手法・モデルベース開発・データ白書など信頼性の高いソフトウェア開発の参考になる情報を展示した。

【安全 / 安心コーナー】では、2013 年 6 月に CSAJ と共同記者発表を行った「ソフトウェア品質説明のための制度ガイドライン」の紹介のほか、PSQ 認証やコンシューマデバイス、第三者がソフトウェア品質を確認するための制度の提案など、SEC の取り組みの紹介を行い、多くの来場者からの質問に、IPA 職員が回答する場面が見られた。

【セキュリティコーナー】では、IPA/ セキュリティセンターで取り組んでいるファジング、自動車の情報セキュリティ、制御関連のセキュリティに関するパネルや関連資料、動画を展示した。

【基盤技術コーナー】では、組込み関連の事業成果を主軸とし、ソフトウェア・エンジニアリングに関する展示のみならず、IPA/ 情報処理技術者試験センターで実施している i パス（IT パスポート試験）の紹介を行い、興味を寄せる来場者が多く見られた。

【地域団体との連携コーナー】では、SEC が連携を実施している全国各地の業界団体の事業を紹介した。来場者からは個別の団体についての質問をいただいたり資料を求められるなど、多くの来場者からの関心を寄せられた。

また、IPA ブース内でブースプレゼンを実施した。20 分間のショートプレゼンテーションで、IPA/SEC の事業成果をはじめ、IPA/ セキュリティセンターや SEC と連携している地域団体・組織からの発表もあり、3 日間で計 39 セッションを実施した。

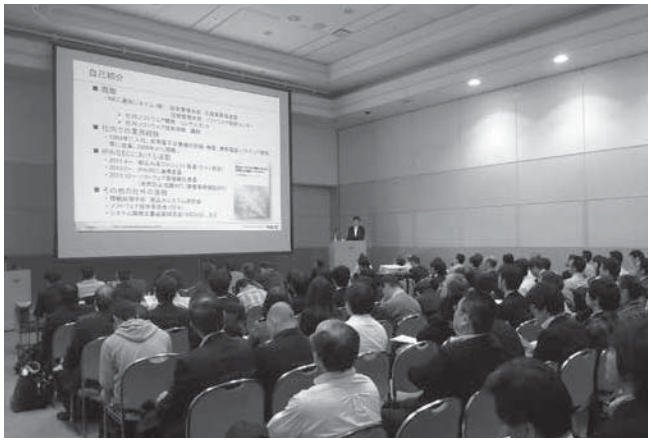
4. IPA セミナー

展示会場に隣接されたアネックスホールで、11 月 21 日、22 日の 2 日間、開催した。

1 日目は「ソフトウェア高信頼性への道程」をテーマに、利用者視点でのソフトウェア信頼性に見える化、モデルベースシステムズエンジニアリングの解説、組込みシステムセキュリティ、形式手法適用による品質確保などについての講演を行った。

2 日目は、「重要インフラシステム障害への取組み」がテーマであり、重要インフラシステム障害対策に関する IPA/SEC の取り組み、組込み製品の信頼性・安全性向上への取り組み、組込みソフトウェア信頼性向上のためのテスト事例、生産性・品質向上への取り組みなどについて、対策や事例の紹介を行った。

両日とも、午前に基調講演、午後はテクニカルセッションというかたちで、計 8 セッションを実施し、延べ 700 名の方にご参加いただいた。



5. 記者発表

ET 開催初日の 20 日には、「日本のモノづくり技術を国際標準提案へ」と題し、下記2つのテーマについて ET2013 の会場で先行発表を行った。

- 日本のすり合わせ開発と融和性の高い開発方法論の国際規格を提案
  ～高信頼なコンシューマデバイスを効率的に開発～
- 組込みソフトウェア開発向けコーディング作法ガイド（ESCR）を改訂
  ～ C 言語規格 C99 に準拠し、最新版 MISRA C に対応する～

これらの内容については、IPA ブース内展示コーナーで該当パネルを設置し、ブースプレゼンも実施した。

6. 今後の展示会出展に向けて

今回の ET は、全体来場者が昨年実績を下回る厳しい状況ではあったが、IPA ブース・IPA セミナーの来場者は、昨年実績を 1 割以上、上回る結果となった。

展示会場での記者発表や SEC の事業内容をアニメーションにした幕間を映写するなど、展示会に向けて新たな試みを実行し、成果に結びついた。事業成果普及対象者の方や関係者の方から、直接フィードバックやご意見をいただけることは展示会の醍醐味であり、出展する意義だろう。2014 年 7 月には、大阪で ET-WEST2014 が開催され、IPA/SEC も出展を予定している。この展示会では、今回記者発表を行った ESCR 改訂版（書籍）の提供や高信頼に関する様々な事業の進捗や成果を紹介する。

ET2013　IPA/SEC ウェブサイト
http://www.ipa.go.jp/sec/events/20131120.html
- IPA セミナー・IPA ブースプレゼンの講演資料がダウンロードできます。
- IPA セミナーの動画を公開しています。

Column

夢でなくなった物体瞬間移動！？

IPA 顧問
松田　晃一

3D プリンターのニュースをしばしば目にするようになってきた。当初の物珍しい話題というよりも、モノづくりの現場での実用に関するニュースが増えている。例えば医療の現場だ。CT や MRI のデータを基に患者自身の臓器を正確に再現した生体模型を 3D プリンターで作ることができる。医師は、この精巧な模型を使って、あらかじめ手術のイメージトレーニングを繰り返し、最適な手術手順を確認する、といった使われ方だ。普通には見えない、触れない物を、リアルな模型で手にすることができるメリットは大きい。

3D プリンターはモノづくりの現場に

プラモデルやフィギュアを作るといったホビーの世界や研究開発現場での試作品の製作といった用途はもちろん、自動車や航空機の部品や金型づくりなど製品そのものの製造に本格的に使われる時代を迎えているようである。3D データさえあれば熟練した職人の手を借りなくても、日本のお家芸である精密な金型まで 3D プリンターで短期間に作ることができるとのこと。複雑で精巧な製品がボタン一つで作れるデジタルモノづくりへの変革が着々と進みつつある。

3D データは誰のもの？

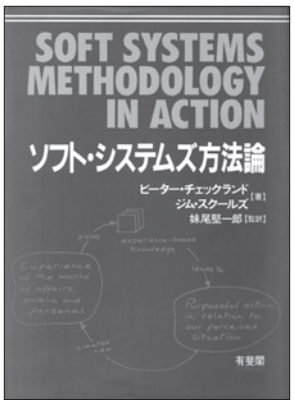
このような時代にキーとなるのは、職人の熟練した技ではなくて、3D データだ。3D データを制するものが、モノづくりを制することになる。このためにも、3D データの保護や権利確保の問題は今後大きな問題となろう。3D プリンターで見える形に表現されたモノは意匠権や著作権で保護できるかもしれないが、その元となる 3D データはどうやって保護するのか。立体造形をスキャンして作った 3D データは、もとの造形を作った人の所有なのか？スキャンして 3D データに変換した人の所有なのか？CAD でデザインして 3D データを作った場合は？などなど、なかなか悩ましい問題である。

いずれにしても、目に見える、形のあるモノを保護するこれまでの考え方を、そのままデジタルデータに適用すると様々な矛盾を抱えることになる。これまでには無い新しい権利保護の仕組みが必要だ。3D データに関する知的財産の管理について、多くのステークホルダのそれぞれが適正な権利を確保できる合理的な仕組みが作られることをぜひ望みたい。そのような仕組みができれば、3D データを中心とした新しいビジネスが次々立ち上がってくることが期待できる。技術的には十分実現可能となった電子ブックのサービスが、著作権などとの関係でなかなか立ち上っていない轍を踏まないようにしたいものである。

モノがネットで瞬間移動

振り返って見ると、CD や書籍、DVD といったモノの形で流通した音楽、小説、映像などは、今やデジタルデータとしてネットを介して流通するようになった。同様に、立体の造形がモノとしてではなく 3D データとして流通し、自宅の卓上や近所のファブラボ（そのうちコンビニの店先になるかもしれないが）の 3D プリンターのボタンを押すとモノとして手元に取り出すことができる時代が夢ではなくなったのだ。3D スキャナーと 3D プリンターが身近にあれば、実物をスキャンした 3D データをネットで交換するだけで、コピーを手に入れることができる。SF の世界で空想されていたテレポーテーション（物体瞬間移動）が、現実になったかのようだ。

明治 2 年 12 月（1869 年）に東京と横浜の間で初めて電信が始まった頃、「電線に頼信紙や荷物をぶら下げる者がいた」とのこと。一瞬にして相手方に届く電信を初めて経験した当時の人々は、通信文や荷物が電線を伝わってそのまま運ばれると信じたようだ。いつ荷物が移動するのか、ずーっと電線を見上げていた笑い話が、140 余年の後の現代では現実になろうとしている。



ソフト・システムズ方法論

ピーター・チェックランド
ジム・スクールズ　著
妹尾 堅一郎　監訳

ISBN: 978-4-641-07573-5
株式会社有斐閣刊
A5 判・430 頁
定価 5,040 円（税込）
1994 年 7 月刊

ソフトウェア開発上流工程におけるビッグピクチャ

もう 10 年以上前になるが、国際的 NPO の日本事務局長をしていた。組織運営に苦慮していた時に偶然この本に出会った。もともと本書は情報理論に基づくソフトウェア・システム開発のためのものであったが、組織はシステムであり、組織構造、ステークホルダ、諸活動の関連などを見直すには格好の手法であり、Davenport のプロセス・イノベーション（Process Innovation, T.D. Davenport, Harvard Business Press, 1993）と共に非常に助けになった。

最近、ソフトウェア・システムの高信頼性が議論されるにあたり、ソフトェア開発ライフサイクルの上流あるいは超上流に目が向けられるようになり、本書が引用されることが多くなったように感じている。この本の翻訳版の発行が 1994 年で約 20 年前のことである。オリジナルのソフト・システムズ方法論（SSM と以降省略）そのものは、1970 年代に開発されたということだからもう 40 年。ソフトウェア開発における方法論の総称としてソフトウェア工学という名称が世に出たのは 1960 年代終わりだからほぼ同じ年数を経た方法論である。本書では、SSM とは何かに加えて、SSM の産業界、国家医療制度、官庁への適用例が紹介されている。また、他の研究事例も紹介されており、何回も読み直したい本の一つである。

（新谷　勝利）

弱いロボット

岡田 美智男　著

ISBN: 978-4-260-01673-5
医学書院刊
A5 判・224 頁
定価 2,100 円（税込）
2012 年 09 月刊

ゴミ箱ロボットでこれまでの製品企画をリフレッシュ

日本のロボットは国際的にも産業競争力を持つと言われている。近年では、サービスロボットや自動運転などニュースを賑やかせている。いずれも高度なセンシング技術を利用し、高度な機能を実現している。本書で紹介されるロボットは、これらの先進的技術を利用したロボットではない。なんとソーシャルなスキルを持つロボットなのである。

掃除ロボットは日本でも大ヒットしたが、本書に登場する代表的ロボットはゴミ箱ロボットである。しかしゴミを拾う機能はない。掃除するロボットに求められる機能は、ゴミを見つけゴミを拾うことである。これらはセンシング技術を駆使することで実現できるが、高度な認識処理やメカの機構が必要となる。ゴミ箱ロボットは、ゴミを探してトボトボ歩きまわり、ゴミを見つけると人に助けを求めるように体をゆらす。そしてゴミを拾ってゴミ箱に入れてもらうと会釈のような動きをする。単体でなく他者や環境も踏まえたシステム企画・設計は、今後のシステム開発において重要なキーワードであろう。

ゴミ箱ロボットのセンサーの数などでは語れない。自分でできない機能を他者に委ね、協力して掃除する。ソーシャルな関係のなかで、ロボットと人が共存する。ゴミを拾えないロボットがいるから、ゴミがないようにしておく。

何でもインターネットに繋げ、リッチな UI で操作可能にするのではなく、何が利用者にとって心地よいかを考えた製品やシステム企画をしたい。

本書はこれまでの性能競争から「ことづくり」コンセプトメイキングに転換するため、思考のリフレッシュにお勧めの書籍である。

（渡辺　登）

編集後記

SEC journal35 号をお届けします。今回の小川紘一先生との所長対談は、校正を重ねるごとに文字数が増え、予定の五割増となりました。その結果、‘ソフトウェア・リッチ’時代と定義された環境での取り組むべき課題が熱く語られ、明確に読みとれます。ぜひ、ご一読ください。

また、この号では、SEC が取り組んでいる情報処理システムの信頼性向上へ向けて、既に活動されている企業や団体から投稿していただき、幅広い高信頼化への取組みが紹介されています。

SECjournal 論文賞 2013 が発表されました。前年は該当論文が少なく、順延しましたが、今年は 8 編の対象論文から 4 編がノミネートされ、表彰委員会にて、現場で役立ち、メッセージ性に富む 3 編が選ばれました。受賞者の皆様おめでとうございます。なお、最優秀賞、優秀賞、所長賞にはそれぞれ副賞として、100 万円、50 万円、20 万円が贈られます。

SECjournal 論文賞は企業・団体・個人を問いません。皆様の投稿をお待ちしております。

（編集長）

編集部より

次世代のソフトウェア・エンジニアリング等に関して、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

初日の出の富士山＠江の島　　（撮影：k-endou）

SEC journal® 第 9 巻 第 4 号（通算 37 号） 2014 年 1 月 31 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

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。

SEC journal No.35

IPA 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 jounal No.35
第 9 巻第 4 号（通巻 37 号）
2014 年 1 月 31 日発行

ISSN 1349-8622

独立行政法人情報処理推進機構