

SEC[®] 7

journal

Software Engineering Center

ハイブリッドなコスト見積りモデルの 反復的な構築方法について

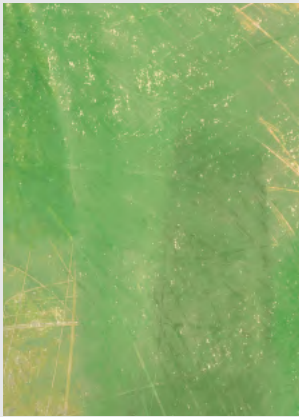
Adam Trendowicz ,Jens Heidrich ,Jürgen Münch

石谷 靖 ,横山 健次 ,菊地 奈穂美




独立行政法人 情報処理推進機構

<http://www.ipa.go.jp/>



SEC journal

Software Engineering Center
No.7目次

- 1 巻頭言
鳥居宏次 (奈良先端科学技術大学院大学 特任教授)
- 2 所長対談：Ron Bell 元英国政府保健省健康安全局 (HSE) 局員
IECとSECとが協力して
機能安全の国際的な相互認証の
スキーム確立を目指そう
- 6 所長対談：Richard Mark Soley OMG会長
産業競争力の強化には、
ソフトウェアの標準化と
スキル標準の確立が欠かせない
- 10 論文
ハイブリッドなコスト見積りモデルの
反復的な構築方法について
Adam Trendowicz, Jens Heidrich, Jürgen Münch, 石谷靖, 横山健次, 菊地奈穂美
- 22 海外レポート
ソフトウェア工学国際会議
ICSE 2006 上海に参加して
神谷芳樹
- 30 技術解説
ソフトウェアプロダクトライン開発の概要
岸知二
- 32 機能安全の枠組み
田邊安雄
- 40 アングル
最近のソフトウェアファクトリ論
松本吉弘
- 50 地域からの発信
東北地域の産業競争力強化に向けた取り組み
東北地域産業クラスター形成戦略「TOHOKUものづくりコリドー」
～モノ作りとITの融合による地域の競争力強化～
中山陽輔
- 54 組織紹介
韓国ソフトウェア振興院 
牧内勝哉
- 56 BOOK REVIEW
- 57 ソフトウェア・エンジニアリング関連イベントカレンダー
- 58 あとがき
- 59 お知らせ (論文募集 / SEC journal バックナンバー)

データ収集、「見える化」そして産学連携



奈良先端科学技術大学院大学 特任教授

鳥居 宏次

2004年11月のSEC発足シンポジウムにおいてデータ収集の重要性を述べたところ、小生の主張は30年前と変わっていない、と親しい参加者からのコメントがあった。その1年半前から、われわれは文部科学省の新事業として、「データ収集に基づくソフトウェア開発支援システム」（通称、EASE：Empirical Approach to Software Engineering）プロジェクトを実施していた。プロジェクト管理における勘と度胸と経験からの脱却を願って、事実に基づく工学的な管理技術の確立をねらった。事実を証明するデータしか頼れるものはない、との信念が30年間同じことを語る結果になってしまっていたようである。

企業データの「見える化」はSECならではのこと

データ収集については、いわゆる制約付き実験として、大学における学生実験時に実施する。ところが、ソフトウェア開発では産と学とで規模が違いすぎるし、担当者の知識量やスキルが違うために参考にならず、企業現場のデータこそ研究上も参考にできると考えられている。現実には企業側として、社員の能力、企業の品質保証の技術レベル、さらには、ソフトウェアの価格基準までわかってしまうのではないかという危惧があり、最近のSECのような特別の努力がない限りデータ収集は至難のわざである。

最近、SECには「プロジェクト見える化部会」がある。きっかけはソフトウェアの現場の様々な事象をさらけ出し問題を見るようにする、ということらしいが、やはり、見たい人に、見たいものを見る姿にして、見ても

らえる技術を明らかにする部会なのであろう。

実は本当に見たいのは単純には見えないものなのである。収集されたデータをうまく分析した結果を見たい。たとえば、外注先、さらにはオフショア開発における開発状況であり品質保証の作り込み能力であるが、今はまだ見る手段はない。しかし、これもコンピュータやネットワークの高速化、大容量化などの恵まれた環境にあって、もう少し技術が進めば何らかの方法で見えるようになるかと信じている。

「見える化」は産学連携にも期待

もう一言、産学連携について述べておきたい。大学の使命は教育と研究と社会貢献である。しかし、時代とともに産学連携は地域貢献のみならず、産業を支援する手段として技術移転や研究成果としてのシーズの提供も求められるようになった。ところが、最近は逆に教育にまで産業界からの支援が必要だとの風潮が出ている。ソフトウェアで実際の現場のことを教えるのに、企業出身者が非常勤の先生として教壇に立つとか、学生がインターシップとして現場で実戦経験を積む機会が与えられる、等である。実は教育は簡単そうだが非常に難しいもので、教育者である前に人格者でなくてはならず、学生への責任が伴うのは当然である。国立大学の法人化以来、教育現場でも教育評価が当然の時代になっているが、教育現場でも教壇に立つのが誰であれ、データに基づく「見える化」による評価が当たり前の時代になっている。

IECとSECとが協力して 機能安全の国際的な相互認証の スキーム確立を目指そう

航空機や自動車に利用される組込みソフトウェアの比率が高まる中、組込みソフトウェアの安全性が重視されている。イギリスで過去に起きたプラント事故を背景に、ソフトウェアを含む電子機器・製品の機能安全(Functional Safety)を保証するための国際標準規格「IEC 61508」が策定されている。2006年度のタスクフォース活動として機能安全に取り組むIPA/SECの鶴保征城所長と、長年にわたってIEC 61508の策定に尽力したRon Bell氏(元英国政府保健省健康安全局(HSE)局員)が機能安全の重要性と、機能安全の認証を普及させていくために必要な国際的なスキームづくりについて語り合った。



鶴保 征城(つるほ せいしろう)
1966年大阪大学大学院工学研究科電子工学
専攻修士課程修了後、日本電信電話公社
(現NTT)入社。NTTソフトウェア研究所
所長、NTTデータ通信株式会社取締役開発本
部長、同社常務取締役技術開発本部長、
NTTソフトウェア株式会社代表取締役社長
を歴任し、2004年10月独立行政法人 情報
処理推進機構 ソフトウェア・エンジニアリ
ング・センター所長に就任。

- ・社団法人 情報処理学会 会長(2001年～
2002年)
- ・XMLコンソーシアム 会長(2001年～)
- ・高知工科大学工学部情報システム工学科
教授(2003年～)
- ・日本BPM協会 副会長(2006年～)
- ・実践的ソフトウェア教育コンソーシアム
会長(2006年～)
- ・社団法人 電気情報通信学会 フェロー
- ・社団法人 情報処理学会 フェロー

鶴保 SECは2004年10月、ソフトウェアエンジニアリングを日本のソフト業界に普及させることを目的としてIPAの中に設立されました。現在、産学から250名を超えるメンバが集まって、エンタプライズ系と組込み系のソフトウェアエンジニアリングに関する問題の解決策を研究しています。そのテーマの1つとして、組込み系ソフトウェアの信頼性という問題があります。とくに、日本ではエレベータの動作の不具合によって高校生が死亡するという事故が起きて、組込みソフトウェアの信頼性確保に対する関心が高まっています。私どもも、情報システムの信頼性、組込みソフトウェアの信頼性を担保するという観点から、機能安全に対して非常に興味を持っています。

Bell 初めに機能安全に関する国際標準規格であるIEC 61508が策定された背景を申し上げます。ヨーロッパでは、1980年代にマイクロプロセッサを機械の安全のために用いようという動きが始まりました。中でも、中心となったのはイギリスです。イギリスで安全性に対する関心が高まる中、業界と規制当局とが連携して安全確保のための体制が整えられました。国際標準化機関として、ISOもあります。ISOはどちらかというとな

技術系の規格で、IECは電気分野及び機械分野に特化した規格を策定しています。プログラマブルエレクトロニクスシステム(PES)を安全に使っていくことを目的としたタスクグループがIECに設置され、私が委員長を務めました。

タスクグループは、機能安全に関する検討を重ね、その結果をレポートとしてIECに提出しました。そのレポートには各種のレコメンドーションが含まれており、国際的な投票でコンセンサスが得られたので、さらに活動を進めていくことになりました。そこで、ソフトウェアを対象とするワーキンググループ及びシステムとハードウェアを対象とするワーキンググループが設置され、1986年から活動を開始しました。

システムエンジニアリングに関するIEC 61508の標準化作業は、私が委員長を務めて進めました。ワーキンググループには、ヨーロッパだけでなく、日本やアメリカの方も入っていただき、国際的な活動で標準化の作業を進めました。この規格は、PES全体をカバーする規格となっています。その対象には、ソフトウェアも含まれています。

鶴保 機能安全を確保するためには、ライフサイクルのすべてにわたってリスクを把握し、対策をとることが重要です。

Bell おっしゃるとおりです。IECのタスクグループでは、システムエンジニアリングの規格を策定する際にリスクベースのアプローチをとりました。そして、いかに安全なシステムを設計するかということを考え、安全なシステムを作るために何が必要なのか、具体的な要件を見ていくという作業を行いました。

その要件として挙げられたものを我々は「全安全ライフサイクル」と呼んでいます。そこに含まれるものは、仕様、設計、実現、運用、保守、改修、そして廃棄です。1987年から2000年にかけてIEC 61508の7つのパートが作成されました。全部合わせると500ページ近くになります。その成果が、2000年までに発表されたIECの規格です。

鶴保 機能安全の標準規格を策定するきっかけにはいくつかの事故があったと聞いています。HSE(英国の健康安全局)では、34の事故の原因を分析した「Out of Control(制御不能)」という報告書をまとめていますね。Out of Controlが機能安全のベースになっているのでしょうか。

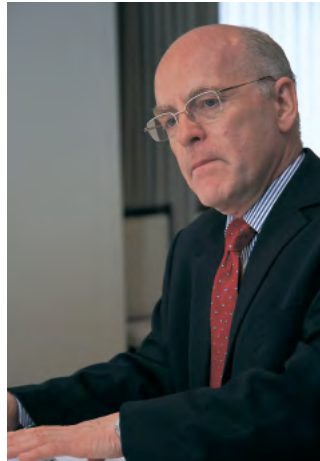
Bell どちらかというと、安全にかかわるすべての段階、すべてのフェーズ、つまり全安全ライフサイクルを強化しなければいけないというオリジナル議論に基づいて規格ができたというほうが正しいかと思いますが、スペックから設計までを含めた全機能安全を認識しなければいけないと考えた背景に、Out of Controlがあったことは事実です。とくに、スペックが正しくないと大きな問題が生じるということが事故事例の研究をする過程でよく理解できるようになりました。制御システムの主要原因を調査したHSEの事故事例分析によって、事故の故障要因の44%に仕様の欠陥があったこと、15%に設計・実装上の問題があったこと、20%に試運転後の変更の問題があったことがわかりました。事故原因の60%は機器を設置・運転する前にあったのですが、15%は運転・保守段階で、6%は設置・試運転の段階で問題があったのです。つまり、基本的にすべての機能安全のライフサイクルの様々な段階でフェイルが起きていることがわかりました。ですから、すべての段階でフェイルが起きないようにしていかなければならないと考えています。

認証取得の背景にはエンドユーザの要求がある

鶴保 先ほど申し上げましたが、日本ではエレベータの故障によって高校生が死亡するという事故が起きました。その後も、エレベータの事故が多発しています。エレベータ事故に関してどうお考えですか。

Bell まず、原因の究明を行うことが大切でしょう。その際に、仕様、設計、運用、保守等の段階すべてにおける根本的な原因の究明を行うべきだと思います。また、今回、事故を起こしたエレベータが規格に適合していたかどうか調査することが必要でしょう。さらに、エレベータの規格そのものについても適合性を確認することが必要ではないでしょうか。

鶴保 日本で、国際標準規格を取り上げて議論するとき、イギリスやその他のヨーロッパ諸国、あるいはアメリカの標準を翻



Ron Bell(ロン・ベル)
公認技士、国際電気標準会議(IEC)フェロー。機能安全に関わる制御システム(とくにコンピュータによるもの)の標準ガイドラインの開発に25年以上もの関与を続けている。英HSE局(政府保健省健康安全局)の電気制御システムグループの部長を務め、英仏海峡トンネル安全委員会のメンバーとしても活躍した。IEC 61508のワーキンググループの委員長を務め、現在は、改訂プロジェクトの委員長を務めている。また安全に関わるIEC諮問委員会(IEC Advisory Committee on Safety(ACOS))のメンバーとして機能安全を担当する他、IET(Institution of Engineering and Technology)の機能安全専門家ネットワークの委員長を務めるなど、多方面で機能安全に関わる活動を行っている。今年5月、HSEを退職して、ロン・ベルコンサルティングを設立した。

訳することが多いのですが、単に規格を翻訳するだけでは表面的な理解にとどまり、規格の本来の意図や内容が伝わりにくい面があります。したがって、SECとしては、翻訳と並行して日本の事故の事例研究を積み重ねる必要があると考えています。

Bell 私も鶴保さんとまったく同じ考えです。IEC 61508の規格はプロセス産業に特化して作られた規格ですが、製品用の規格というわけではありません。もっと基本的な規格です。ですから、規格の何条何項を順守して製品を作れば信頼性の高い製品が作れるというたぐいのものではありません。

鶴保 機能安全の認定・評価についてお話をうかがいたいのですが、現在、どのような状況でしょうか。

Bell IEC 61508は、ソフトウェアの認証もシステムの認証も、外部の独立した第三者機関によって認証される仕組みになっています。第三者機関が、ソフトウェアまたはシステムの審査を行って規格を満たしているかどうかを判断し、認証を発行することになっています。認証を取得する理由として多いのは、自分たちが作った製品の安全性に問題がないと自信を持つためということが挙げられます。それは、製品を実際に使用するエンドユーザの要求に応えるためです。例を挙げると、モータがあったとしてそのモータの機能安全が認証されていれば、化学プラントにそのモータを使用する際に安心と自信を持つことができます。

鶴保 システムは様々なソフトウェアやコンポーネントによって構成されています。複雑なシステムに対してもIEC 61508は安心や自信をもたらすのですね。

Bell PESの場合、さまざまな複雑なエレメントが入っています。例えば、スマートインスツルメント、ロジックプロセッサ、アクチュエータ等です。それらのエレメントが互いに通信できる必要があるとともに、PESのソフトウェアもエレメントとし

て機能します。そこで、それらのエレメントが認証されていれば安心につながり、また、認証されているエレメントをパッケージとしてインテグレーションする側もパッケージの認証を受けることになります。

鶴保 実際にどのような製品、システムが認証を受けているのでしょうか。

Bell トータルなシステムとして認証を受けているケースは少ないですね。化学プラントの防護系等に用いられる製品は非常に特異性が高く、認証を受けるにあたって非常に高いコストがかかります。しかし、製品認証ということであれば、スマートインスツルメント、PLC（プログラマブルロジック・コントローラ）等は認証を受けているものが多いと思います。洗濯機等に使われているマイクロプロセッサ等は特異なもので、認証を受けるときもコストがかかりますが、50万台販売できれば1台当たりのコストは小さくなります。

鶴保 これからIEC 61508規格の認証は普及していくとお考えですか。

Bell スマートインスツルメント等のエレメントの認証は今後増えていくと考えています。また、複雑なアプリケーション等も認証を受けるものが増えていくでしょう。とくに、複雑なアプリケーションでは、仕様と設計の認証が必ず要求されるようになると考えています。また、化学プラント等で使用されるシステムや機器の場合、保守の手順も認証が必要になるでしょう。

コンピテンシーマネジメント教育が重要になる

鶴保 機能安全に対するヨーロッパのソフトウェアエンジニアの関心はどのようなものですか。

Bell とくに、イギリスについてお話しさせていただきます。1980年代の前半からイギリスでは2つの専門機関が機能安全に大きな関心を持っています。1つが、British Computer Society、もう1つがInstitution of Electrical Engineersという機関です。ソフトウェアエンジニアリングとシステムエンジニアリングの両方の機関が機能安全に大きな関心をもって活動しています。この2つの機関は互いに協力し合っており、IEC 61508規格の策定にも大きくかかわっています。

鶴保 ソフトウェアエンジニアに対して機能安全に関する教育

やトレーニングは行われているのでしょうか。

Bell 機能安全に関する確立されたトレーニングプログラムはヨーロッパに存在しません。ですから、これから作っていかないといけないと考えています。ただし、各種の機関が主催している機能安全のトレーニングコースはあります。1日コース、2日コース、4日コースとあり、日本企業からの参加者がいるコースもあります。また、高度な内容の集中コースもあります。トレーニングコースの確立は大きな問題で、これから注目されていくことになるでしょう。

IEC 61508規格は非常に複雑なものです。また、処方箋のように企業の取り組み方を詳細に記述しているものではありません。そこで、実際にこの規格の導入・利用に携わった経験のある人の知識に頼るしかないのが現状です。この問題に対応すべく、イギリスでは今、コンピテンシーマネジメントコースが実施されています。組織レベルと個人レベルの2つの面でコンピテンシーマネジメント教育が必要になってくると思います。市場は、コースを受講してコンピテンシーが備わっていると認定された人に与えられる認証を求めています。

今後、求められるのはグローバルな認証スキームの確立

鶴保 航空機や自動車等、製品に占めるソフトウェアの比重が高まる中、SECとしても機能安全には関心を持っています。そこで、2006年度の活動として、ソフトウェアの故障モデルや安全度水準、人材スキル等をタスクフォースとして取り組んでいく計画です。今後のテーマとしてグローバル認証が重要になる



と考えていますが、グローバル認証に対するベルさんのお考えはいかがでしょうか。

Bell IEC 61508規格を効果的に利用してもらえようグローバルな認証スキームが必要になると考えています。例えば、ヨーロッパでシステムを開発している開発者が日本のソフトウェアコンポーネントを使いたいという場合、そのコンポーネントが日本で認証されていれば、ヨーロッパでも問題なく認証され、それをヨーロッパの開発者が利用できるというグローバルなスキームが必要になるでしょう。

鶴保 各国の認証機関の相互連携が求められますね。

Bell そうです。グローバルな認証スキームを実現するためには、IECの規格をもとにさまざまな認証機関が認証を行っていくことが欠かせません。具体的には、ピアレビューを行うことが必要です。また、認証を行う第三者機関を国が認めた認定機関が認定することも必要となるでしょう。この考え方はヨーロッパ的で、日本的ではないかもしれませんが。

鶴保 日本もヨーロッパに近い考え方を取っています。日本でも、公的機関が認証機関を認定するという考え方になるのではないかと思います。日本とアメリカの間では、例えば暗号モジュールの相互認証が検討されています。さらに、ご存じのように日本とヨーロッパの間でもコモンクライテリアという相互認証のスキームが採用されています。これらは、暗号とセキュリティというようにドメインが特化されています。

一方、機能安全の対象は幅広いので、どのように認証の制度を設計していくのか、どのようなスキームで進めていくのか。我々も大きな関心を持っています。

Bell 認証とともに、もう1つ、ピアレビューが注目されています。これは、業界主導型のレビューになると思います。例えば、EXスキームという危険な環境で使用される装置のレビューを行う方法が、IECで構築されているところです。

認証という方向とピアレビューというもう1つの方向とがあり、それぞれ強みと弱みをもっています。そこで、機能安全のためには、両方の方法を取ることが重要です。

鶴保 ピアレビューはボランティアで行われているのですか。

Bell あるグループに経験を積んだ人たちがいて、その中に新人が入ってきたときにレビューするというのが基本的な方法です。新人が入ってくることによって、確立された規格、標準が弱くなることを防ぐために、レビューを行おうという考え方は

最近登場してきたものです。

鶴保 ソフトウェア工学で行うピアレビューは、ドキュメントの作成者がレビューするのではなく、第三者の仲間がレビューすることを指しています。

Bell 機能安全に関するピアレビューは、新人をレビューすることと、組織をレビューするという2つの考え方で行われています。組織のレビューは、トレーニング組織や組織自体の能力についてレビューすることです。

鶴保 プラントや要素部品の半導体、あるいは金融システムや航空管制システムといった各種のドメインの問題と機能安全の関係についてはどのように考えていますか。

Bell 2つあると思います。第一に、組込みソフトウェアやIC等を作っているシステムコンポーネントの製造業がドメインとして挙げられると思います。第二に、業界的に見ると石油化学工場、オートメーション、原子力、航空産業、防衛産業がドメインになってくると思います。ハードウェア、ソフトウェアを含めたサブシステムやコンポーネントを作っていてIEC 61508に準拠している場合は、様々な産業で使われることになるでしょう。スマート系の装置やソフトウェア、ASICを作っている企業はIEC 61508によって認証を受け、認証を受けたコンポーネントや製品が様々な市場で利用されていくことになると思います。

鶴保 日本では、組込みソフトウェアに対する関心が高まっています。また、組込みソフトウェアの市場も拡大しています。安全性が認証されたソフトウェアが利用されることは、安全性への寄与と同時にソフトウェアの流通を促進させることへの寄与も期待できます。

Bell いろいろな業界で利用されているソフトウェアコンポーネントの認証を受ける場合、業界ごとに認証を受けるとすると、企業は煩雑な作業を強いられることとなります。1回の認証で済むことが望ましいのです。そうした仕組みづくりに対して、SECがこれまで培われてきたすばらしい膨大な専門知識を生かしていただけることを期待しています。国際規格の作業部会等にもぜひご参加いただいて、これからの国際規格の策定に大きな影響を与えていただければありがたいですね。

鶴保 今後、我々もIECと連携して、機能安全に関する活動を進めていきたいと考えています。

産業競争力の強化には、ソフトウェアの標準化とスキル標準の確立が欠かせない

自動車や家電等、製品におけるソフトウェアの比率が増す中、ソフトウェアの生産や向上及び品質の確保、そしてソフトウェア技術者のスキルの向上が大きな課題となっている。世界的な標準化団体のOMG (Object Management Group) のRichard Mark Soley会長とIPA/SECの鶴保征城所長がソフトウェア技術の標準化及びスキル標準のあり方について語り合った。



鶴保 征城(つるほ せいしろう)
1966年大阪大学大学院工学研究科電子工学専攻修士課程修了後、日本電信電話公社(現NTT)入社。NTTソフトウェア研究所長、NTTデータ通信株式会社取締役開発本部長、同社常務取締役技術開発本部長、NTTソフトウェア株式会社代表取締役社長を歴任し、2004年10月独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター所長に就任。

- ・社団法人 情報処理学会 会長 (2001年～2002年)
- ・XMLコンソーシアム 会長 (2001年～)
- ・高知工科大学工学部情報システム工学科 教授 (2003年～)
- ・日本BPM協会 副会長 (2006年～)
- ・実践的ソフトウェア教育コンソーシアム 会長 (2006年～)
- ・社団法人 電気情報通信学会 フェロー
- ・社団法人 情報処理学会 フェロー

鶴保 SECは、ソフトウェア関連の産業界と学界とが共同でソフトウェアエンジニアリングに関する技術開発とその普及を推進していく産学連携をサポートすることを目的に設立されました。経済産業省と協調して、産学の触媒機能を担っています。Soley 1989年に設立されたOMGも同様に、ベンダ、エンドユーザ、政府機関、そして教育機関の団体によって構成されています。OMGは、ソフトウェアの移植、再利用、相互運用を可能とする標準を開発してきています。よく知られている成果としてMDA (Model Driven Architecture)、UML (Unified Modeling Language)、CORBA (Common Object Request Broker Architecture)があります。また、OMGの活動の90%は、様々な産業を対象にしたもので、金融、医療、テレコム、軍事ロジスティクス等、25の産業分野をカバーしています。

OMGに参加している団体・企業は550社ほどあり、その半分がソフトウェアのエンドユーザ、半分がベンダです。さらに多くの政府機関と大学関係の組織が標準の開発に取り組むとともに、OMGのスタッフが標準に関する資格認定作業を行っています。地域的に会員の内訳をみると、北米が半分、西ヨーロッパ

パが1/3、日本が8%を占めています。その意味でもOMGにとって日本は重要です。

鶴保 SECの活動目的は、技術開発そのものではなく、日本の産業界の競争力強化にあります。そのために、最も効果的な方法として産学官の連携を促進させることに取り組んでいます。つまり、企業単独や産業界だけではなく、大学の知見も取り入れて、手法の統一や定量データの蓄積を進めているのです。その成果はある種の標準と呼べるものですが、ダイレクトに国際的な標準を目指しているわけではありません。結果として標準となることは期待できますが。

OMGは、国際的なスタンダードづくりを目指しているのですね。

Soley おっしゃるとおりです。我々は、研究機関ではなく、国際標準化団体です。我々の活動は、既にラボから外に出た技術で標準化が必要と思われるものを対象としています。

鶴保 SECの活動対象は、エンタプライズ分野のソフトウェアと組込み分野のソフトウェアの両者です。組込み分野では、組込みソフトウェア開発力の強化を目的とした組込みスキル標準(ETSS)を作成しています。OMGとはETSSが縁となって互いに協力していこうということで合意していますね。

Soley OMGとしては、ETSSに大きな関心を持っています。OMGは、2002年にスキルマネジメントや専門知識の認定に対する需要が非常に大きいと認識しました。そこで、UTI (UML教育研究所)とパートナーシップを結んで3年間にわたって資格認定テストの開発をしてきました。それが、UMLプロフェッショナルの認定試験「OCUP」です。OCUPは、ソフトウェアを構築・運用・管理する上で必要な設計表記法に関する知識を問うもので、既に1万人以上が受験しています。

現在、UTIと組込みシステム、リアルタイムシステムのエンジニアリングの資格認定プログラムの開発を進めています。その過程で、組込みシステムのスキルやスキルマネジメントに詳

しい組織を調べたところ、SECの存在を知りました。SECのETSSグループの考え方・方法は、我々が組込みシステムのエンジニアを認定しようとしていた方法と非常に似ていると思いました。

鶴保 そもそもETSSというのは、日本が最も得意とする組込みシステム領域の技術者の育成や製品の信頼性確保を考えたときに、スキルはどのようにあるべきかを産学連携によって一から見直したのです。その点がOMGのメンバから評価されて、SECとOMGの意見が合致したのでしょう。

Soley 我々は、資格認定の前提となるスキルモデルを調査しました。米国政府やフランス政府のモデル、IBMのような大企業のモデルも調査しました。その内容はそれぞれ異なっているのですが、それらはいずれも特定の技術に絞り込みすぎているのです。それに対して、ETSSについて大原茂之先生（東海大学教授）から学んだときに、これは単に組込みシステムだけでなく、もっと一般的なものに使えると感じました。建設業のスキルであれ、無線のスキルであれ、非常に広範囲に使えるということがわかりました。

鶴保 ETSSは組込みシステムから始まっていますが、おっしゃるとおり、広範囲に使えると思います。というのは、組込みシステムの適用領域そのものが非常に広いからです。適用される製品としては、携帯電話、自動車から始まって、デジタル家電等、様々です。また、例えば自動車の中でもエンジン制御とカーナビのソフトウェアとは違います。したがって、ETSSではターゲットシステムが千差万別であるという状況を踏まえて、なおかつスキルを定義しなければいけないという問題に真正面から取り組んだので、ソーリーさんが言われる評価につながったのでしょう。

標準化によってソフトウェアの市場は拡大する

鶴保 OMGは国際的な標準化活動に取り組んでいますが、その中で最も成功したものを挙げるとすると何でしょうか。

Soley UMLの標準化です。OMGがUMLを標準として採用したのは1997年のことです。それ以前には、20社から12のオブジェクトモデリング言語が出ていました。10年が経過した現在、



Richard Mark Soley
(リチャード・マーク・ソーリー)
MITコンピュータサイエンス研究所で人工知能を研究、博士号を取得。在学中より、ピクチャーテル、シンボリックなどベンチャー企業の設立に参加。1989年、OMG設立に参加、以来、戦略と技術に関わる全業務を総括し、世界最大のソフトウェア標準化コンソーシアムに育て上げた。1997年、会長に就任。CORBA、UML、MDAなどOMGの主な標準技術のすべてに主導的な役割を果たしている。

オブジェクトモデリング言語の標準はUMLだけです。20社の総売上げは3000万ドルだったのですが、UMLは、インタラクティブ開発環境（IDE）やオープンソース製品、独自仕様の製品すべてに使われ、現在の製品市場は約40億ドルになっています。UMLは、標準によって市場が成長するよい例だと思います。最初は、コンソーシアムの標準から始まって、やがてデファクトスタンダードとなり、ISOの標準となったのです。

CORBAも非常に成功したケースです。CORBAは現在、20億台のコンピュータシステムで実行されています。組込みシステム、デスクトップ、すべてのJava、IBMのWebSphere、BEAのWebLogic等を実装されています。また、全世界のテレコムノードがすべてCORBAになっています。皆さんがお持ちの携帯電話にもOMG標準が実装されていると思います。

また、ヨーロッパの航空管制にもOMG標準が実装されています。また、日本、オーストラリア、韓国、NATO諸国において、これから10年の軍用の無線は、すべてOMGの標準に基づいたものになります。このように、我々は非常に多くの標準規格をもっています。

鶴保 CORBAは大きな成功を収めていますが、ベンダの中には、自社の技術を広げていきたいという考えを持つところも存在します。ベンダが標準化を阻害する面もあるのではないのでしょうか。CORBAの標準化に際しても非常に苦労があったと思いますが、いかがですか。

Soley それは秘密です（笑い）。ベンダには、標準を回避したいという理由があると思います。自分自身で価格を設定したい、あるいは自社で市場を支配したいという思いがあるでしょう。ただ、それはユーザの要求とは違います。一方で、ユーザが望むものは、必ずしも標準のものとは限りません。例えば、銀行間の支払いプロトコルは銀行間ごとに異なっています。これはベンダが原因となっているのではなく、ユーザ側の要求によるものです。

我々としては、1つの標準規格を押しつけるのではなく、様々なもの間を橋渡しして、相互運用性や移植性、再利用性

を高めていくことを目的に活動を行っています。各ベンダにはエゴがあるということもあるのですが、市場が非常に拡大し、大きなマージンが得られると認識してもらえると各企業とも一緒に標準化活動をしていくことになります。

鶴保 技術的には、XMLはうまくできていますね。自分が理解できない部分は無視して、理解できる範囲だけで解釈すればいいということになっています。技術的な進歩もあるのではないのでしょうか。

Soley IT市場で心配なことは、非常に急いで新しい技術を追いかけていることです。XMLは確かにある問題を解決することができますが、すべての問題を解決することはできません。XMLを含む基盤技術も変化しています。現在の傾向としては、サービス指向アーキテクチャ（SOA）への移行が進展し、エンタプライズ・サービス・バス（ESB）が登場しています。しかし、これらがすべての問題を解決するわけではありません。また、モデリングも同様です。ですから、すべての技術や方法を持ち寄って、それらを橋渡しする標準について合意することが必要なのではないのでしょうか。

OMGの認定資格を ETSSに適合させることを検討

鶴保 UMLは世界における標準化の成功例ですが、地域的な普及状況はどのようなものですか。

Soley 西ヨーロッパでは90%、北米では70%、日本はまだ普及率が低く10%と聞いています。モデリング市場には大きな変化があります。主要な開発環境ではUMLが整備されていま

す。もちろん、日本の開発者もその開発環境を使うことができます。ただ、ソフトウェア開発の考え方を変えるということも必要になってきます。とくに、組み込みシステム・リアルタイムシステムの開発にはそれが言えると思います。そこで、資格認定ということが重要になります。モデリングに移行した場合、技術者を採用する企業側も、資格が認定されている技術者なら、モデリング技術をきちんと知っていることがわかるからです。OMGでは、OMG-Compliant Realtime and Embedded Specialist（OCRES）という資格を開発しています。そして、大原先生のチームと、どのようにすればOCRESをETSSの枠組みに適合させることができるか、話し合いを進めており、我々としてはOCRESをETSSに適合させようと考えています。OCRESの開発と普及において、ETSSが非常に重要であり、また、組み込み技術の専門知識を活用するという点ではベストな方法でしょう。

鶴保 ETSS自身、内容的にまだ変化しているものです。ETSSとOMGが考えるOCRES資格とがどのような関係になるかは、これから詰めていくことになりそうですね。

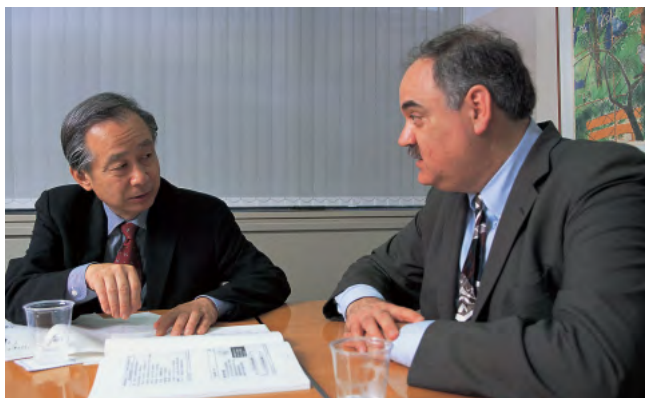
Soley OCRESもまだ最終的な完成には至っていません。我々も問題を作成してベータテストを行っている段階にあります。それによって、スキルとして内容的にETSSとマッピングが取れるかを確認していこうと考えています。

鶴保 日本ではソフトウェア技術者を目指そうと考える学生が減少しています。米国ではどのような状況でしょうか。

Soley 全体として、ソフトウェア技術を専攻した卒業生の数は減っています。米国には、移民が多数います。高等教育を受けた移民もいるので、彼らを活用することによって人材の不足を補っています。また、インドや中国にアウトソーシングして人材不足に対応しています。

鶴保 ソフトウェア技術者の育成という問題に関して、スキル標準のもつ意義は大きいと考えます。日本の今の若い人は、就職して自分が成長できるという実感がもてないのでIT産業の人氣が下がっていると思います。世の中には3K的な仕事、苦しい仕事、大変な仕事がたくさんあるのですが、若い人は、仕事を通じて成長したという実感がもてる会社や産業に惹かれます。IT産業の会社は、若い人が成長する実感がもてるよう努力する必要がありますが、そのためには技術的なスキル標準をしっかり固めないといけないと思います。

Soley 認定試験の実施によってエンジニアの数が増えること



を期待されているのです。他の地域に比べて、日本の方が資格認定を重視している、とくにエンジニアを雇う側が重視しているという点は、文化的な違いかもしれません。日本では、雇うのは簡単で解雇するのは難しい。ヨーロッパは両方とも難しいですね。

OMGとSECが協力して 標準化の問題を解決しよう

鶴保 日本では、OMGは技術的な標準化に取り組んでいる団体として認識されています。しかし、スキル標準のようなマネジメントの領域を手がけているという認識はあまりありません。

Soley OMGにとってスキル標準は新しい分野です。しかし、我々は多くの技術と技術インフラを持っています。また、他の標準化団体に比べて迅速に行動できるとともに、多くの実績を保有しています。例えば、ビジネスプロセスモデリングの領域ではいち早く標準化を行っています。技術の標準化だけでなく、マネジメントについても標準化を行っています。

鶴保 マネジメント分野に関しては、他の標準化団体の方が強力なのではないかという意見もありますが、どうお考えですか。

Soley 我々の組織は、他の組織にはない非常に迅速なプロセスをもっています。また、ユーザとベンダの両者が会員となっているという大きな長所があります。さらに、ISOとも良好な関係をもっています。ただし、我々が唯一の団体であるということをおっしゃっているわけではありません。OMGは、他の機関とも強力な関係を維持して、多くのプロジェクトを共同で行っています。しかし、OMGのメンバ企業はスキルマネジメントの標準化に非常に強い関心を持っています。同時に、我々は最大のメンバシップを持っています。ですから、OMGだけがスキルマネジメントの標準化ができる機関ではないと申し上げましたが、ベストな機関であると考えています。同様に、SECもスキルマネジメント標準を検討する唯一の機関ではありませんが、ベストな機関だと思えます。

鶴保 OMGの軸足が、技術だけでなくマネジメントを包含するようになってきていると考えてよいのでしょうか。

Soley OMGは、以前は片足を分散型オブジェクトコンピュー

ティングに、もう一方の足をモデリングに置いていました。今では、もっとたくさんの足があり、テレコムに足を置き、金融システムにも足を置いていました。今度は、大きな足をビジネスプロセスマネジメントに置いてあります。このように、OMGにはたくさんの足があるのです。

鶴保 国際標準化団体の会長としてSECにアドバイスをいただけますか？

Soley スキルマネジメント標準の分野でSECと一緒に作業できることを非常に誇りに思っています。さらに、SECのこれまでの経験について非常に尊敬しています。また、OMGがSECによく知られていることをとてもうれしく思っています。SECに対するアドバイスはとくにありません。というのは、SECには非常に優秀な方がそろっているからです。SECにはいろいろなチャンスがあると考えています。

我々が行っているのは、世界中のソフトウェア開発の効率を高めていくということです。SECは、主に日本市場をご覧になっていますが、もはや日本市場は世界市場と同じなので、日本はソフトウェアの輸出国であるといえます。ですから、両者の興味と利害を合わせていけば、必ず、両組織のメンバにとって価値のある方向が見つかるかと確信しています。鶴保さんから、OMGに対してメッセージをいただけるでしょうか？

鶴保 現在、標準化に対して、日本の力が落ちているといわれています。この10年、産業界の力が弱くなったといわれています。日本の産業界は、リエンジニアリングを優先してきました。その結果、標準化の優先順位が下がっています。

しかし、スキルマネジメントを標準化し、ソフトウェア技術者を育成するという問題は、日本の国際競争力の原点です。標準化における日本のリーダーシップを復活させなければいけないと思っています。日本の場合、産業界は官頼みの傾向がありました。

この10年、官の側は、「産のできることは産で」というスタンスをとってきました。両者の考え方の狭間に標準化に対する取り組みが落ち込んでしまっています。それを拾い上げて立ち上げていきたいと思っています。

Soley OMGとSECが一緒になって標準化の問題を拾い上げていきましょう。

ハイブリッドなコスト見積りモデルの反復的な構築方法について

Adam Trendowicz † Jens Heidrich † Jürgen Münch †
石谷 靖竹 横山 健次竹 菊地 奈穂美竹, †††

下記に掲載する論文は、SECとドイツ・フランホーファ協会実験的ソフトウェア工学研究所（IESE）¹で2004年11月から2005年8月にかけて実施した見積り手法に関する共同研究の成果をまとめたものである（内容については一部削除・修正）。2006年度のICSE（International Conference for Software Engineering）²に研究論文として採択され、2006年5月、ICSE2006においてAdam Trendowicz氏が発表した。なお、本論文の翻訳はACM（Association for Computing Machinery）より許可されている³。（翻訳：石谷靖）

コスト見積りは、ソフトウェア開発企業にとって非常に重要な活動である。そして、見積り技術の導入に際しては、見積り精度が重要な判断材料となる。しかしながら、精度の評価はただ一度の試行で決定されることが多く、当該見積りモデルの改良の可能性は、採用の判断に当たって考慮されないことが多い。さらに、見積り方法の多くは、反復的な構築による見積りモデルの改良を明確には意識していない。これは、重要なコスト要因またはデータ間の矛盾を見落とす危険性を増加させる原因となる。本論文は、CoBRA⁴法に基づくコスト見積りモデル構築に対して、反復的な分析、フィードバックサイクル及びその評価を系統的に導入し、拡張したプロセスを提示するものである。沖電気工業株式会社のソフトウェア開発部門でモデル改良サイクルを実施し、見積り精度は、当初のモデルで120%の誤差であったが、最終的なモデルで14%に向上した。本論文では、反復的なモデル構築アプローチによって得られた知見を示す。

Development of a Hybrid Cost Estimation Model in an Iterative Manner

Adam Trendowicz †, Jens Heidrich †, Jürgen Münch †, Yasushi Ishigai ††, Kenji Yokoyama ††, Nahomi Kikuchi ††, †††

Cost estimation is a very crucial field for software developing companies. The acceptance of an estimation technique is highly dependent on estimation accuracy. Often, this accuracy is only determined after an initial application. Possible further steps for improving the underlying estimation model typically do not influence the decision on whether to discard the technique or deploy it. In addition, most estimation techniques do not explicitly support the evolution of the underlying estimation model in an iterative manner. This increases the risk of overlooking some important cost drivers or data inconsistencies. This paper presents an enhanced process for developing a CoBRA cost estimation model by systematically including iterative analysis and feedback cycles, and its evaluation in a software development unit of Oki Electric Industry Co., Ltd., Japan. During the model improvement cycles, estimation accuracy was improved from an initial 120% down to 14%. In addition, lessons learned with the iterative development approach are described.

1 はじめに

高品質のソフトウェアに対する需要の急成長とともに、ソフトウェア開発は世界的に主要なマーケットの1つになっている[GARTNER2005-1, GARTNER2005-2]。ソフトウェアの社会への浸透は進み、その変化と複雑さは、絶

えず進展している。マーケットの変化は激しく、より一層の高機能性、高信頼性、高性能がソフトウェア製品に要求されるようになっている。ソフトウェア開発では、プロセス、開発方法やツール等で進歩しつづけなければ、競争に勝ち残ることはできない。また、競争力維持のために、ソフトウェア開発組織は、予算内・予定期間で合意されたレベルの品質を確保しつつ、開発費の低減や期

† ドイツ・フランホーファ協会実験的ソフトウェア工学研究所（IESE）, Fraunhofer Institute for Experimental Software Engineering

†† IPA/SEC, Software Engineering Center

††† 沖電気株式会社, Oki Electric Industry Co., Ltd.

間の短縮も要求される。そして、ソフトウェアを調達する側でも、開発か購入かの判断や、ソフトウェア製品の開発費の妥当性を判断するための根拠が必要である。

ところが、非現実的なコストで、予定より遅れ、予算を超過して終わるか、完成しない場合も多く[TSG2003]、信頼できる見積りの必要性は高い。

ソフトウェアコストの見積りは、他の産業の見積りよりも難しいと考えられている。これは、主として同じ製品を繰り返し製作するのではなく、新しい製品を開発することが多いからである。さらに、ソフトウェア開発は、人間の活動であり不確実なものである。これらの要因が、とくに早期のフェーズで、見積りが困難な原因になっている。課題には、例えば、プロジェクト規模、未知なものを含めた数多くのコスト要因、異なる組織でのコストモデルの適用可能性、信頼できる見積りモデルを構築するための十分な数のデータの入手などがある。これらの多くの問題を扱うために、開発過程のより深い理解と、ソフトウェアコスト見積り技術、手法及びツールの構築・評価に関する研究が多くされている[BRIAND2002]。

見積り方法を選択するにあたっては、適用するのに必要とされるデータ、または、適用のための労力といった適用可能性を評価するが、この2つは通常相反するものである。より少ない労力で見積りモデルを作ろうとすると、より多くの過去のプロジェクト実績データが必要とされる⁵。代表的なものにデータに基づいた見積り方法がある。他方、専門家知識に基づく方法は測定データをほとんど必要としないが、見積りモデル構築に多くの労力がかかり、企業は、少ない労力か、少ないデータ要求かの判断に迷うことになる。例えば、実績データに基づいた方法を使う組織の大部分は、分量の適切なデータを持っていない。ハイブリッドな方法（データ及び専門家の知識に基づく見積りモデル構築方法）は、必要なデータと構築に必要なとする労力に妥当なバランスを与え、妥当な労力で見積りの信頼性を実現する[BRIAND1998, RUHE2003, CHULANI1999]。適用する状況に応じて、各

手法の適用可能性を注意深く評価することが重要である。

組織的な改善活動として、技術的かつ組織的なプロセスの弱みの把握と適切な対応のために、継続的なモニターがなされるべきであるが[BASIL2001]、コスト見積りも、組織上の1プロセスとして、継続的な改善の対象とすべきである。とくに、新しい見積り方法導入時には、現存するプロセスに統合し、適用環境に合わせるために、試行を何度か繰り返す必要がある。実際には、この事実は見落とされがちで、最初のパイロットの結果のみに基づいて導入か否かが決定される。結果として、初めて適用されたとき満足な結果が得られなかった（通常、コストパフォーマンスの点から）との理由で、内容を吟味せず大雑把な感覚によって、有望な方法を採用しないことになりがちである。

本論文では、沖電気工業株式会社において実施した、CoBRA（Cost Estimation, Benchmarking and Risk Analysis）に基づくコスト見積りモデルの反復的な構築を示す。CoBRAは、企業の特性にあったコスト見積り方法として選択した。CoBRAは、専門家の知識と企業の実績データに基づいて透明性の高いモデルを構築する方法を提供するハイブリッド方法である。また、ソフトウェアプロジェクトの定量的なリスクを評価し、ゴール指向的な測定プログラムを支援する。

今回、CoBRAモデルの構築プロセスを拡張し、データや専門家の知識に基づいてモデルの改良を検討する分析・フィードバックサイクルを組み込んだ。フィードバックサイクルを通じて、データにおける矛盾を発見し、誤りや欠落しているコスト要因を特定し、結果的に見積りモデルを反復的に改良した。

2 CoBRA法の基本的な考え方

CoBRA[BRIAND1998]は、プロジェクトコストが、理想的なプロジェクトでのコストとコストオーバーヘッドという2つの要素から成るという考えに基づいている。

1 <http://www.iese.fraunhofer.de/>

2 <http://www.isr.uci.edu/icse-06/>

3 Copyright © ACM, Inc., 2006. Translated by Permission.

4 CoBRA はフラウンホーファー協会実験的ソフトウェア工学（IESE）の登録商標。

5 データが多く利用できればできるだけ、専門家の関与が少なく済み、企業がかけるコストを下げることができる。

$$Cost = \underbrace{Nominal\ Productivity \cdot Size}_{Nominal\ Cost} + Cost\ Overhead \quad (1)$$

ただし、

$$Cost\ Overhead = \sum_i Multiplier_i (Cost\ Factor_i) + \sum_{i,j} Multiplier_{ij} (Cost\ Factor_i, Indirect\ Cost\ Factor_j) \quad (2)$$

理想的なコストとは、理想的なプロジェクトに必要なコストである。理想的なプロジェクトとは、組織または事業体において、仮説的な理想の環境で、最適条件の下で実施されるプロジェクトを指す。つまり、プロジェクト特性のすべてが、開始時において最善なものである。例えば、プロジェクト目的が明確に定義され、関係者すべてが確実に理解し、すべてのキーパーソンが適切なスキルを持つような場合である。コストオーバーヘッドは、理想のプロジェクトからの不完全さを補うために必要な追加コストである。状況を補償するのにチームトレーニングが行われること等はその例である。

CoBRAではコストは、コストに影響を及ぼす要因によりモデル化（要因モデル）される。要因モデルは、プロジェクトマネージャ等の専門家の知識から作成する。図1の要因モデル例の矢印は、直接的または間接的な関係を示す。+と-は、コスト要因が全体のコストに対して、正または負に影響することを示し、要因の変化に応じてプロジェクトコストが増減することを示す。例えば、要求の不安定性が増加するならば、コストも増大する。矢印は要因間の相互作用も示す。例えば、統制のとれた要求管理と要求の不安定性の場合、統制された要求管理は、ソフトウェアコストに対する不安定性の負の影響を補償

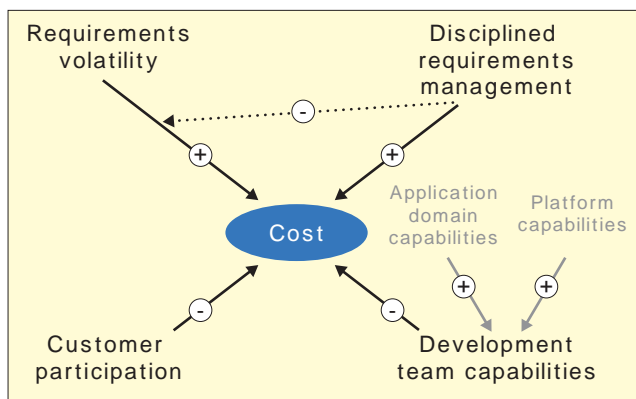


図1 要因モデル例

する効果があることを示す。

間接的影響によるコストオーバーヘッドは、式(2)に示す和の第2の項で表される。一般的に、CoBRAでは多層の間接的な影響を認めている（統制された要求管理に対する影響やその影響に対する影響など）。しかしながらモデルが複雑になり、評価が難しく、モデル構築の労力が増すため、実際には、考えられるすべての要因を要因モデルに組み入れることは必ずしも勧められない[BRIAND1998]。

各コスト要因の定量化は、コストへの影響度合いや互いの影響度合いの専門家の評価に基づく。影響は、理想的なコストに対して相対的に示される。各コスト要因に対して、専門家は、当該コスト要因について最も悪い極端な場合を想定して、理想的な場合に対するコストの増加分を評価する。評価の不確実性を定量化するため、上記の要因ごとに最大（最悪）、最小、最も有り得るコストの割合の回答が求められる。

CoBRAの第1の要素である理想的な場合のプロジェクトコストは、要因モデルに含まれない特性（開発タイプ、ライフサイクルタイプ等）が類似の過去プロジェクトデータに基づく。これらのプロジェクト特性は見積りの前提条件となる。過去のプロジェクトデータは、コストオーバーヘッドとコストとの関係を決定するために使われる（式(1)）。これは2変数の関係なので、多くの測定データを必要とせず、また、基本的にはプロジェクトの規模と工数が必要である。規模には、すべての成果物を含む全体のプロジェクト規模を反映する必要がある。なお、一般的に規模の単位は、コード行数またはファンクションポイントである[LOTHER2001]。またプロジェクトごと

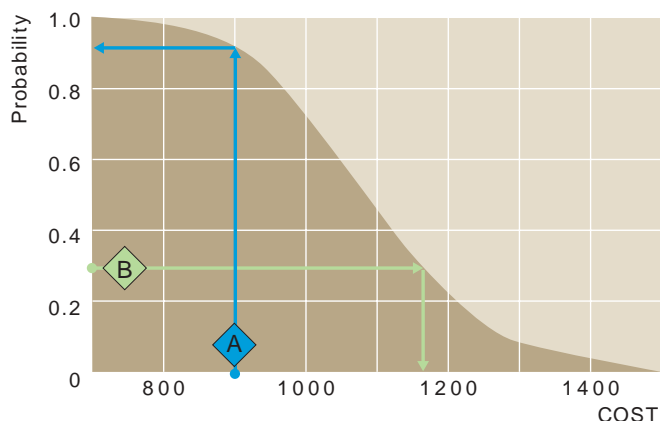


図2 累積費用分布の例

のコスト要因の情報は、各プロジェクトに関係した専門家に確認する。

コストオーバーヘッドのモデルは、定量化された要因モデル、過去のプロジェクトデータ、そして、現在のプロジェクト特性に基づきシミュレーションアルゴリズム（例：モンテカルロ法またはラテン方阵法）を用いて生成される。得られた確率分布は、コスト見積りのような計画管理活動、コスト関連のプロジェクトリスクの評価、または、ベンチマーキングに使われる[BRIAND1998]。

図2に、累積費用分布を使った、リスクの確率に対するコストの算出と決められたプロジェクトコストに対して実績が超えてしまう確率の算出方法を示す。

例えば、プロジェクトの予算が900であり、コスト分布が図2のとおりとすると、予算を超過する可能性は90%となる。この確率が許容リスクを超えていれば、プロジェクト予算は認められない(シナリオA)。次に、プロジェクトマネージャが超過の危険を最小限にしようとする場合(シナリオB)では、例えば実績値が見積り値を超えるリスクを30%と設定したときは、予算を1170より低くすべきでないことがわかる。

他の見積り手法と比較し、CoBRAは測定データが少なくても済み、また規模及び工数の単位に対する制約がない。CoBRAでは、組織の特性に合わせて見積りモデルを策定できるので、適用可能性等（見積り精度、正確さ、一貫性等）が高い[BRIAND1998]。

3 産業での事例研究

本節では、CoBRAに基づく反復的なモデル構築の方法と試行の結果について示す。

3.1 反復的なモデル構築

コスト見積りモデルは、図3に示す枠組に従い、最初に見積り分析準備段階として、入力データの完全性、一貫性、正確さを検証した。続いて、見積りモデルを作成し、モデルの品質を評価するために見積りモデル構築後の分析を行った。今回は過去のプロジェクトデータに対してクロスバリデーション法（1つのデータを残りのデ

ータから見積りする方法）によって検証した。見積り精度及び安定性の評価として、平均相対誤差及び相対誤差の中間値（MMREとMdmRE）[CONTE1986]、誤差25%のプロジェクトの割合⁶（Pred(.25)）、相対誤差の標準偏差（Std.Dev）を用いた。さらに、モデル出力（残余）と入力データの間の関係を分析した。

モデルの改良は、見積りモデル構築前及び見積りモデル構築後の分析に基づいて、関連するプロセス（例えば、情報収集プロセス）に対するものと、見積り方法そのものに対するものを決定した。

以下、反復的な改良の実際について述べる。

3.2 初期モデル構築

今回の試行は、金融系アプリケーションを扱う沖電気工業のビジネス部門で実施した。過去の16プロジェクトの実績データが用意され、12人の専門家（プロジェクトマネージャ及び品質マネージャ）がモデル構築に参加した。過去のプロジェクトデータの分析・評価と主要なコスト要因について検討した。すべて、HP-UX及びWindows環境での機能拡張プロジェクトであり、ウォーターフォールモデルで開発されたプロジェクトである（図4）⁷。

CoBRAモデル構築の最初として、企業の専門家と3種類の会合を実施した。1回目の会合では、CoBRAアプローチの基本を提示し、要因モデルの役割及び使用方法を説明した。2回目の会合では、当該組織での基本的なコ

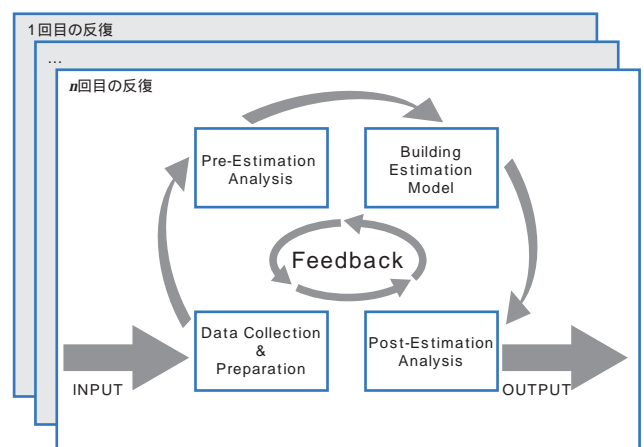


図3 見積りモデルの反復的開発

6 これは、25%以下のMMRE値のプロジェクトの比率を定義する。

7 スケールはプロジェクトデータの秘密保持のため示していない。

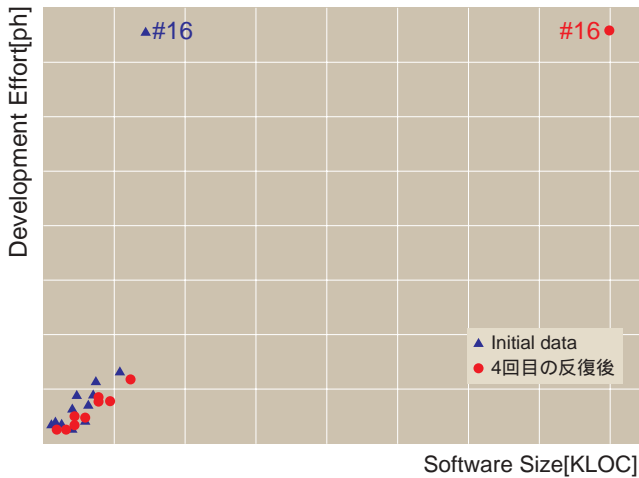


図4 規模と工数の散布図
 (規模は、コメント行抜きのLOC (Java及びC)で測定, 工数は人時で測定)

スト要因について議論し、要因モデルを組み立てた。グループ討論の作業を効果的に実行するために、参加が決まっていた専門家へ、訪問の2、3週間前にあらかじめコスト要因に関するアンケートを実施し、コスト要因の候補リストを作成した。それらのうちのいくつかは、さらにサブ要因（例：図5のPERS.2に示す適用分野の経験とプラットフォームの経験）に分割された。そして、各要因とその結果生じる工数増加の間の直接的な関係について議論した。

続いて、要因間の間接的な関係（例：統制された要求管理は、要求定義とその安定性に対して強い影響力を持つ）を議論した。2回目の会合で、すべての専門家の同意の下、図5に示す要因モデルを決定した。

決定後、ある要因の影響度合いについて最悪の場合から

最善の場合までを4段階のレベルに分け、それぞれに、4段階のLickertスケール [SPECTOR1992]としてレベルを設定した。それから3回目の会合として、各専門家へのインタビューを実施した。インタビューでは、基本的に要因のコストへの影響度合いと過去のプロジェクトの特性（各要因レベル）に関するデータを集めた。影響度合いは全専門家に、過去のプロジェクトの特性はそのプロジェクトに関係した専門家に確認した。

インタビューでは、全専門家には、コスト要因が引き起こす可能性のある最小、最大と通常起こり得る工数増加を確認した。これは、見積りに使われる確率分布（図2）を計算するためのモンテカルロシミュレーションの入力となる。

特定の専門家へは、上記のコスト要因に関して過去のプロジェクトの状況を確認した。専門家の労力を軽減するために、1つのプロジェクトに対して1人の専門家のみに確認したが、後述のとおり、結果的には失敗であった。

見積りモデル構築前の分析としてこれらのデータを分析した結果、見積りモデルの精度に影響を与える可能性のあるい

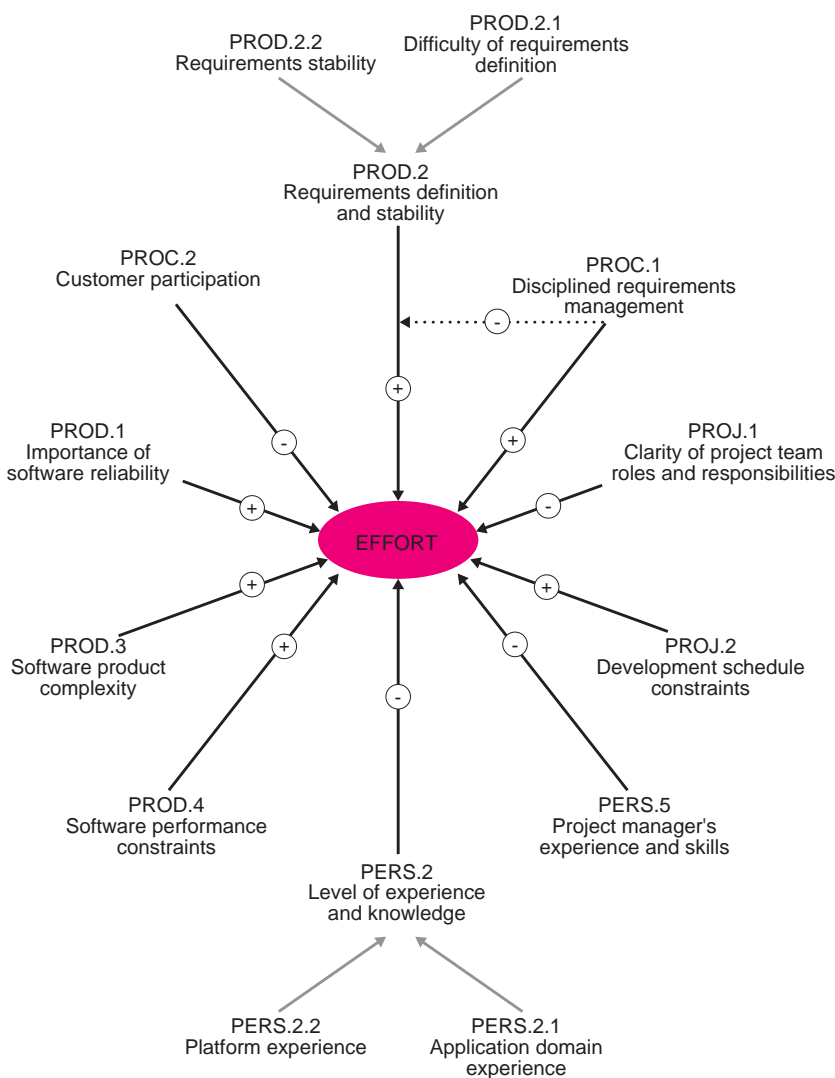


図5 最初のCoBRA要因モデル

くつかの外れ値が含まれていることがわかった。プロジェクト#16と一群の4プロジェクト(#11~#14)が、他のデータと異なった傾向を示した。生産性の分析では、規模の不経済(規模が大きくなると生産性が低くなる)については有意ではなかった。

専門家から提供されたコスト要因データにもいくつかの疑問が生じた。また、インタビューでは、何人かの専門家から、「ソフトウェアの信頼性の重要性」はコスト増加に関連しないこと、そして、要因モデルから除外することが提案された(結果的には、多くの専門家が賛成しなかったので除外されなかった)。そして、専門家から、乗数データ(各要因のためにコストオーバーヘッドの値を得ることにとって必要な影響度データ)の収集時に、各要因について他のものすべてから切り離して効果を想像するのが非常に難しいとの意見が出された。ただし、多数のプロジェクトで得られる専門家の個々の経験を反映するため、乗数データの回答における矛盾は、ある程度避けられない。

各要因のために定義されたスケールは、プロジェクトの状況に応じて一貫した理解がなされ、様々な専門家によって一貫した評価を保証されることが前提である。しかしながら、規模とコスト要因ソフトウェアの複雑さに関して、プロジェクト#16より著しく小さな規模の数プロジェクトは、等しいか、またはさらに高い複雑さを持つとして評価されていた。そのため評価の妥当性に関して疑問が生じた。もちろん、そのようなケースもあり得るが、規模における差異があまりにも有意であったので、本プロジェクトではその可能性が低いと判断された。

さらに、見積りモデル構築後の分析によっても、初期モデルの弱さがいくつか明らかになった。クロスバリデーションによる評価結果は、非常に低い精度であり、低い一貫性となった(表1の初期)。相対誤差の分布の結果は、見積り前の分析と同じ外れ値を示していたので、次

表1 CoBRAモデルの結果

反復	MMRE	MdMRE	Pred(.25)	Std.Dev.
初期	1.07	0.79	6.25%	0.70
1回目	0.32	0.28	46.7%	0.23
2回目	0.23	0.19	66.7%	0.20
3回目	0.23	0.15	73.3%	0.22
4回目	0.14	0.12	93.3%	0.07
4回目及び#16	0.12	0.10	100.0%	0.07

の反復改良にあたっての最初の検討対象とした。

3.3 データ見直しによるモデル改良

上記のとおり、最初のモデル構築の間、データの一貫性に関していくつか問題が見つかった。そこで、まずは専門家の関与なしで、データのみに基づいてモデル改良を行う反復を2回計画した。

(1) 1回目の反復

最初の反復において、入力データから外れ値プロジェクト#16を除外することとした。#16は大規模プロジェクト(規模に関しての外れ値)であることを別にしても、再開発プロジェクトであった。他は、すべて新規または機能追加の開発である。最初の要因モデルがこの要因を明確に含まないこと、そして、開発タイプが他のすべてのプロジェクトでは等しいので、残りの15のプロジェクトに基づいて見積りモデルを再構築することにした。

#16除外後のデータは、見積りモデル構築前の分析により、規模と工数の間に、さらに関係が有意であることがわかり、見積りモデル構築後の分析により、見積り精度及び標準偏差に関してモデルがかなりの改善を示した(表1の1回目)。理想の生産性も同様に規模に依存しておらず、生産性のばらつきは対象プロジェクト全般によく説明されていることが示された。一方、#11~#14は外れ値グループとなったままであった。

測定データを精査した結果、報告された工数が全プロジェクトの同じ工程をカバーしたわけではないことがわかった。情報収集プロセスでは、工数データは正しく集められたが、要件定義及びシステムテスト後半に費やされた工数が入っていないプロジェクトがあったのである。

(2) 2回目の反復

この反復により、総工数を修正し、収集データの定義を改善した(全プロジェクトで同じ工程をカバー)。なお、この問題への対処には2つの可能性があった。(i)一貫して測定されたフェーズのみの工数データを含むことにする、(ii)過去のプロジェクトの工数分布に基づき計算されたフェーズの工数を含めて全ライフサイクルの総工数を再計算することである。例えば、工数データを正規化する後者の方法は、ISBSGデータベースでも採用されている[ISBSG2004]。しかしながら、正規化は開発フェーズごとの工数の分布に関して、過去データの知識を必要

とする．今回は十分なデータがなかったため、最初の方法を採用した．図4に修正したデータでの散布図を示す．

修正されたデータの見積りモデル構築前の分析では、規模と工数間の相関に改善が見られた．一方、プロジェクト#11～#14がMMREに関して大きな外れ値ではないにしろ、視覚的に目立つかたまりを形成していた．そして、プロジェクト特性をさらに精査することにより、#11～#14が部分的にC言語で開発されているのに対し、他はすべてJavaで開発されたことが明らかになった．現在のモデルは開発言語の差異をコスト要因としていないので、次の改良ステップでの検討対象とした．

見積りモデル構築後の分析では、見積り精度と標準偏差に関して改善が見られた（表1の2回目）．見積りモデル構築前後の分析により、曖昧な情報収集プロセスが見積りモデルの品質に対して重大な影響を与えることが示された．なお、かたまりを作っているプロジェクト群と過去のプロジェクトデータに関する専門家の矛盾した評価は未解決のままであった．

3.4 専門家ベースのモデル改良

上記の反復の結果、残された問題の解決には専門家の関与が不可欠と考え、次の改良ステップで専門家と議論を実施することにした．次のステップの目的は、外れ値となっているプロジェクト（#11から#14、#16）の背景にある原因を明らかにし、最初のコスト要因（とくにソフトウェアの信頼性の重要性）の必要性を再検討し、専門家の評価の妥当性を見直すことであった．

(1) 3回目の反復

ここでは、専門家は、要因モデルにはいくつかの重要なコスト要

因が欠けていると結論した（プロジェクト#11から#14を外れ値としている要因）．専門家によれば、1つは、プロジェクト外部の技術者からの支援であり、#11～#14が他のプロジェクトと違う点とされた．一方、#11～#14を他のプロジェクトと区別するものとして意見があった第2の開発言語の利用は、決定的なコスト要因とは結論付けられなかった．#16に関しては、専門家は他のプロジェクトと著しく特性が異なることを認めた．#16はソフトウェアプラットフォームを作成する再開発プロジェクトである一方、他のプロジェクトは、#16で作られたプラットフォーム上でのプロジェクト、または機能拡張プロジェクトである．従って、専門家は、モデルからのこのプロジェクトの除外を受け入れた．

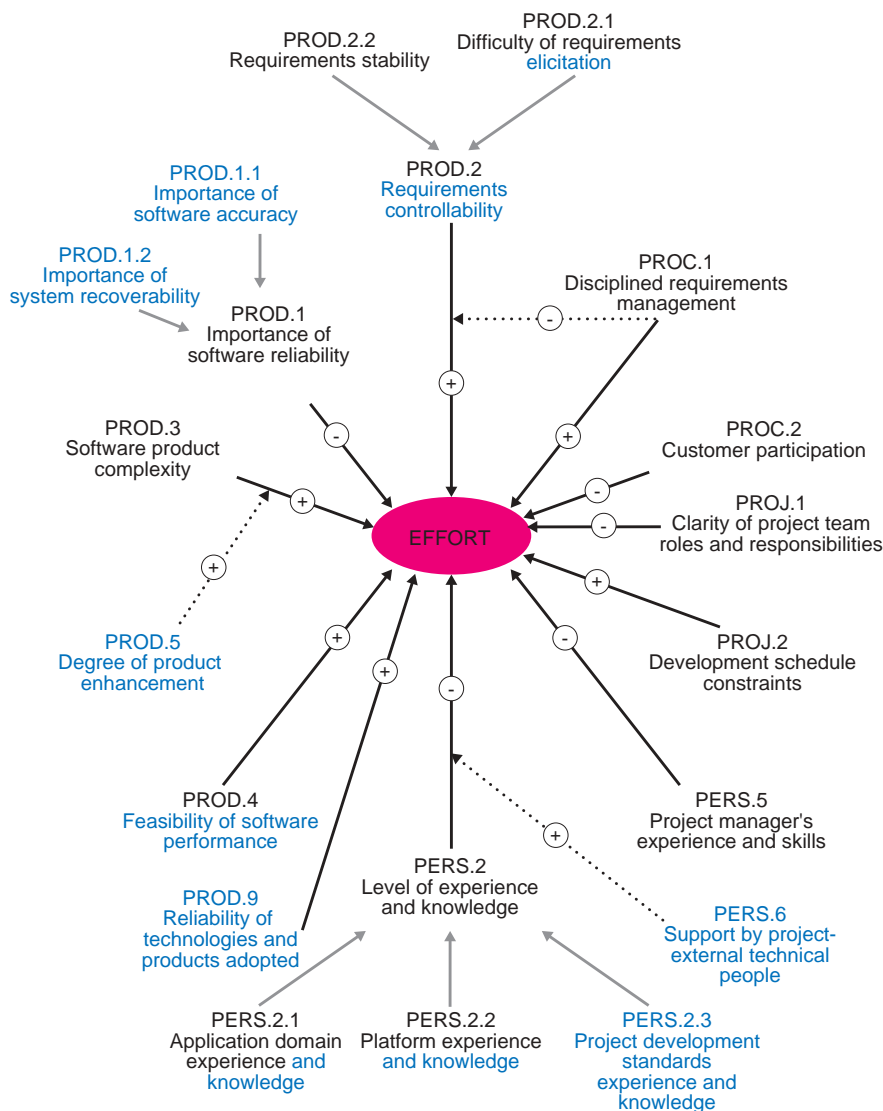


図6 改良した要因モデル

同様に、専門家は、今回除外された#16のいくつかの特性に対処する新しい要因をモデルに含めることも決定した。例えば、新規開発及び機能拡張プロジェクトを区別するために、製品機能拡張の度合いが追加された。また、ソフトウェアの信頼性の重要性が要因モデルに残された。図6に改善された要因モデルを示す。下線で示す要因が、追加されたものである。

専門家の評価の矛盾を防ぐために、1つの過去プロジェクトに対して複数の専門家からデータを集めた。また、各要因の定量化の際、主観的なスケールの理解を統一するために、要因のレベル（スケール）を詳細に定義することに注力し、すべての専門家の参加を得たグループ会合においてその議論を行った。

見積りモデル構築前の分析の結果、単独の専門家から得たプロジェクトデータの信頼性に関して改めて懸念が強まった。実際、いくつかのケースでは、専門家の違いによって同じプロジェクトの同じ要因の評価が大きく異なったものもある。これについては、関係者で評価結果の違いを議論し、共通の結果に到達するための会合を行い解決した。専門家全員で各要因のためのスケールの定義を行ったが、それでも解釈に関して違いが残ってしまったのである。測定データについては変更しなかったため、外れ値となった#11～#14の問題は、未解決のままであった。

見積りモデル構築後の分析は、モデル精度の向上が少し見られたが、見積り安定性では改善は見られなかった（表1の3回目）。さらに、理想の生産性に関する分析では、いくつかの外れ値が明らかになった（図7）。

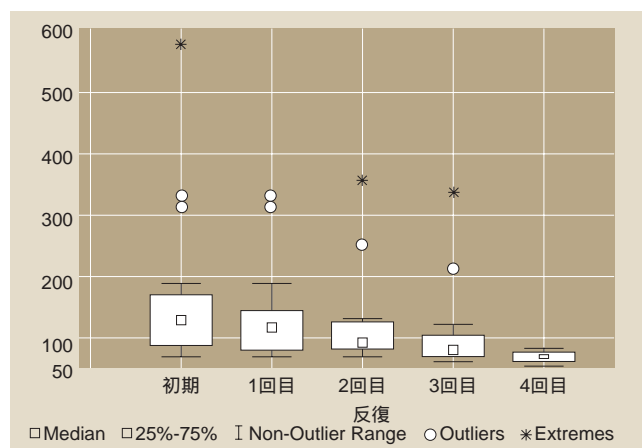


図7 理想の生産性の分布

理想の生産性は、プロジェクト規模に依らず一定であるべきである。しかし、改良されたモデルでも、いくつかのプロジェクト、とくに#1、#3、#8及び#10に関して変動を説明することができなかった。

(2) 4回目の反復

過去のプロジェクト特性をさらに分析することにより、理想の生産性に関しての外れ値であったプロジェクトが、非常に多くのGUIやバッチプログラムを持つことが明らかになった。専門家と議論した結果、モデル構築に使用した規模データは開発者によって直接書かれたコードのものであり、GUIやバッチのような生成されたコード等の他の要素を含まないことがわかった。コードが生成された場合でも、そのための工数が必要となることを専門家に確認し、次の反復的な改良で、使用する規模データを向上させ、それぞれの測定データを集め、そして、見積りモデルを再構築することにした。図4に、修正した規模データの散布図を示す。

見積りモデル構築後の分析では、見積り精度と安定性のさらなる改良を実現した（表1の4回目）。ただし、新しいモデルでも、理想の生産性はまだ一定にならなかった。実際、あるプロジェクトは実際に新しく定義した規模測定に関して外れ値であった。これは、チーム規模のような規模等に関係づけられたコスト要因が要因モデルに含まれていない可能性を意味しており、今後のモデル改善のためのスタートポイントになると考えている。

3.5 モデル改良の要約

図8と図9に4回の改良を通じたCoBRA適用をまとめる。反復が繰り返されるたびに見積り精度と安定性が一定の向上を示したことがわかる。

また、改良の過程で、見積り関連のプロセスに多数の改善が導入された。測定結果は、今回定義し利用したプロジェクトスコープに合わせた工数データに合わせられ、規模測定は、ソフトウェア開発製品（全体のプロジェクト工数に影響する）の規模を十分に定量化するように修正された。さらに、参加した専門家から、そのモデルが組織の経験ベース[BASILI2001]にできるとの意見があった。たとえば、コストに関連した知識を新しいメンバと共有するのに有効である。

3.6 線型回帰モデルとの比較

研究の一環として、最小2乗法（OLS：Ordinary Least Square Regression）の見積り精度とCoBRA法の結果を比較した（表2）。OLS選択の理由はシンプルかつ頻繁に使用されるデータ駆動型見積り技術だからである。

2つの方法は、全改良サイクルで著しく向上した。そして、最終的に非常に似た結果を達成している（表1、表2）。予想していたが、重要なコスト要因において同種のプロジェクトデータであればあるほど（例えば、#16を対象からはずした後のデータ群）、2つの方法の間に大きな相違はないという興味深い結果が得られた。

一方、CoBRA[BRIAND1998]の強みの1つは、コスト及び生産性に関してモデル化された因果関係に基づき、プロジェクトの特性に応じて生産性の変化を説明できることである。対象とするプロジェクトがコスト要因に関して非常に類似しており、同様の生産性を持つとき、

CoBRAの強みは、OLSと比べてあまり優位とならない。この関係を調査するために、外れ値であった#16を過去のプロジェクトデータベースに再び入れて実験を行い、両方法を再実行した（#16のデータは他のデータと違う特性を持っているが、その特性は要因モデルにすでに組み込み済み）。そこでの仮定は、生産性の変化を引き起こすコスト要因が要因モデルに含まれていれば、#16を含めることによる生産性の変化を説明することができるので、CoBRAはOLSより著しくよい結果になるというものである。表1と表2の最後の行の2つの結果を見てみると、わずかに1つの外れ値のプロジェクトを入れることで、2つの方法間の差異は増加した。CoBRAの利点が明らかである。2つの比較を図10に示す。

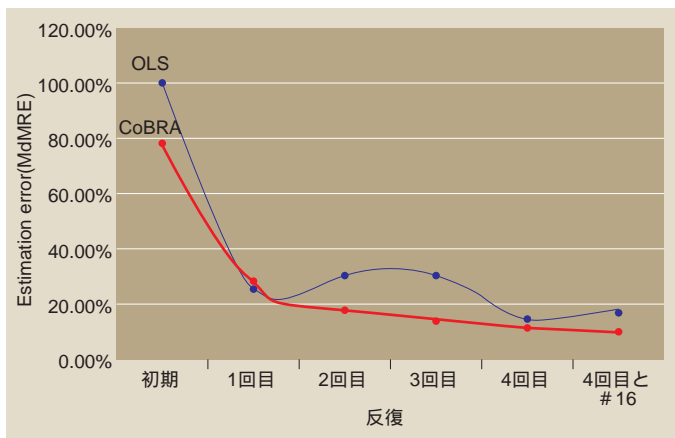


図8 見積り精度の改善

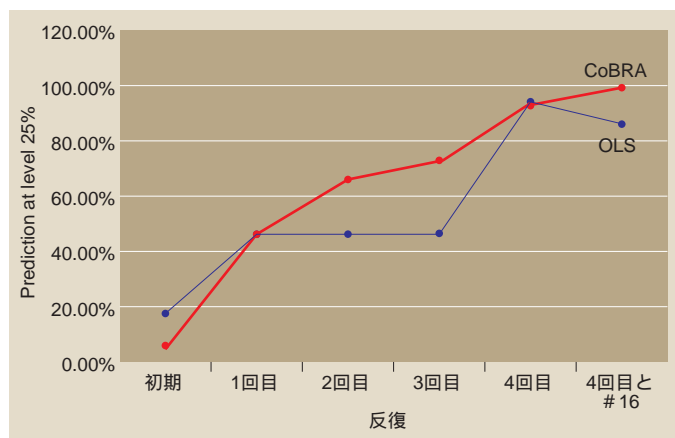


図9 見積り一貫性の改善

表2 OLSによる結果

反復	MMRE	MdMRE	Pred(.25)	Std.Dev.
初期	1.21	1.06	18.75%	0.91
1回目	0.37	0.26	46.67%	0.30
2回目	0.25	0.31	46.67%	0.20
3回目	0.25	0.31	46.67%	0.20
4回目	0.14	0.15	93.3%	0.07
4回目及び#16	0.17	0.18	87.5%	0.09

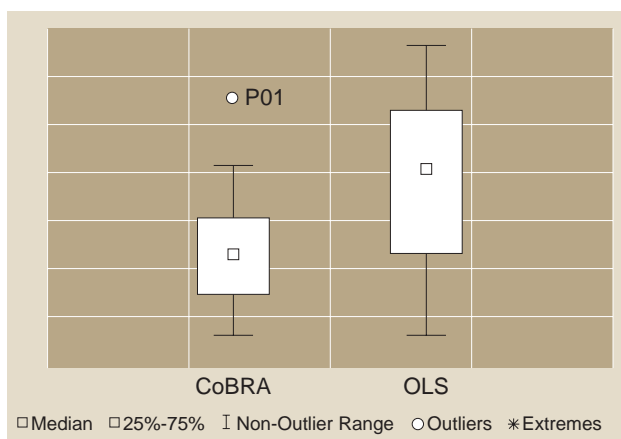


図10 異質なデータがある場合のCoBRAとOLSの結果の比較（t検定を適用し、2つの方法の見積り誤差の間の差異が統計上有意である($P=0.017$)ことを確認している)

4 得られた知見

以下に、CoBRAを反復的に適用することから学んだいくつかの教訓をまとめる。

- (L1) 専門家は、一度にはすべての重要なコスト要因を発見することができない。この場合、最初にパイロットモデルを作り、欠けているコスト要因や冗長なコスト要因の発見支援のために、モデルにプロジェクト特性を追加しながら、対話的に専門家と議論するのがよい。
- (L2) 通常、専門家ごとに、過去のプロジェクトに関して異なる見方をしており、コスト要因の定義は少しずつ違っている。従って、過去のプロジェクトを評価する際には、少なくとも2人の専門家からデータを得ることが賢明である。
- (L3) 妥当な測定データの収集は、見積りモデルとその結果の信頼性に影響を与える最も重要な要因の1つである。CoBRAでは、規模測定でとくに重要である。プロジェクトにおいて開発された製品の規模は、すべてが考慮されなければならない。本研究において、最初の規模測定では、JavaとCコードのみを反映したが、バッチ、GUI及び帳票のような要素は含まれていなかった。また、アクティビティによって工数データがない場合は、すべてのプロジェクトでデータが利用可能なアクティビティのみが、対象とされるべきである。あるいは、工数データは、見積り目的のために適用前に正規化されるとよい。
- (L4) 企業の受け入れにとって、とくに反復的に適用されるためには、必要な労力の低減は重要である。今回、専門家の関与や改良のための労力を最小限にするため、測定データ見直しによる改善から始めた。
- (L5) 最初から妥当なコストモデルを作成することは難しい。今回の反復的な構築からわかるとおり、前の反復の結果に基づいて専門家が議論することによって、コスト要因への新しい洞察、規模測定の定義や現存するプロジェクトの特性の理解が深まる。反復はモデル改良のために重要な作業である。

5 関連研究

これまで様々な見積り方法が開発されているが、それぞれ必要とするデータの種類と見積りモデルの形式が異なる。入力データに関しては、3つのグループに分けることができる。1つは実績データに基づいたもの、もう1つは専門家の知識ベース、そしてもう1つはハイブリッドな方法である。それらの方法の中で、いくつかは、過去のプロジェクトデータに基づいて、カスタマイズされたモデル(自組織用モデル)を構築し、また、他の手法では既に定義されたモデル、即ちコスト要因とその関係を固定したものを提供する(固定モデル)。固定モデルの主な利点は、実績プロジェクトデータを必要としないことである。ただし、固定モデルは特定の環境のために開発され、同じタイプの見積りに適しているが、異なる環境での当該モデルの適用はかなり制限される。見積り結果の向上には、組織のプロジェクトデータに基づいて、特定のアプリケーションを前提として一般的モデルを適合・修正させる必要がある。一方、自組織用モデル構築では、組織のプロジェクトデータを元にモデルを作るが、必要とされる量のデータが、常に利用可能であるとは限らない。ハイブリッドな方法は、データに対する要求と専門家の関与のバランスがよいので、最も実用的なものといえる。

現在、多くのソフトウェア開発組織が、Capability Maturity Model (CMM) のレベル3を達成しようとしている[CMMI2001]。そこでは、測定データの収集が重要な役割を果たす。しかし、実際にデータ収集を開始していても、専門家の知識に頼らざるを得ないことも多い。そのため、大規模なデータベースを必要とすることなく、また、完全に主観的で属人的な見積り方法(例:デルファイ法[FARQUHAR1970])に頼らずに済むハイブリッドな方法が必要とされ、知識の双方を合わせたCOCOMO II[BOEHM2000]、Estor [MUKHOPADHYAY1992]等様々なアプローチが開発されている。

ハイブリッドな方法の実験的な適用[RUHE2003, CHULANI1999]では、データまたは専門家ベースの方法と比較して、高い見積り精度及び安定性が報告されている。そして、専門家の知識ベースとデータベースの両者を活

用した見積り結果が妥当であることを示す理論的な根拠を示している[BOEHM2000]。さらに、ハイブリッドな方法には、間接的ではあるが多くの利点がある。例えば、CoBRAは、シンプルな点推定値に加えて、コスト関連のリスク評価及びプロジェクトベンチマーキングの方法を提供する[BRIAND1998]。自組織用モデル構築アプローチの例として、CoBRAは、現状の測定過程に対してゴール指向型な改善方法を提供する。透明性の高いモデル構造及び感応分析により、生産性に最も大きい影響を及ぼす要因がわかり、その結果、関連プロセスに焦点を当てる改善アクティビティにつなげられる[BRIAND1998]。従って、モデルそのものの連続的な改善の支援、及びコスト関連のプロセスの向上につながる。

品質改善パラダイム (Quality Improvement Paradigm : QIP) [BASILI2001]やCMM(I) [CMMI2001]によって実現されるソフトウェアプロセス改善では、プロセス及び技術の反復的な改善が、特に新たな導入のときに強調されている。しかし、一般にコスト見積りについては、公表された研究[BRIAND2002]から知られているように、反復が強調されることは少ない。関連の研究において、固定モデルを組織に適合させるアプローチとして、コストモデルの改良の例がある[FERENS1998, BOEHM2000, PUTNAM1992, GENUCHTEN1991]。それらでは、原則として、過去のプロジェクトデータに基づいてモデルを実行し、見積り値と実績値が比較され、平均偏差は、モデルの補正のための指標になっており、モデルの適合及び反復的なモデル改良の必要性が示されている。しかし、最近では、企業環境における適用は提示されていない[BRIAND2002]。

実際に産業のデータに基づきモデルを適合させた実証的な研究では、実績と見積りの誤差分析とそれに基づく妥当な要因の選択に限られている[PUTNAM1992]。また、そのような方法で獲得されたコスト要因は、満足な改良に至っていない[FERENS1998, KEMERER1987]。他の方法で、さらに包括的方法でモデルを洗練するものがある。例えば、[BOEHM2000]に示されるCOCOMO.81モデル[BOEHM1981]では、アップデートされた測定データと専門家の知識に基づいてモデルを組織に適合させる。モデルを適合させるプロセスでは、重み(いわゆる回帰係数)とともに、コスト要因の種類がモデルの中で調節される。

ただし、そのような方法は、より一層のプロジェクトデータを必要とする[GENUCHTEN1991, MENZIES2005]が、多くの場合必要とされる量のデータを集めることが難しい。本研究では、測定データと専門家の知識を使って、特定の組織の環境で、CoBRAモデルを何回か繰り返して改良した。モデルの改良は、モデルのパラメータに影響を与えるだけでなく、コスト要因や要因間の関係を含む、モデルの構造全体に影響を及ぼした。

文献に示されるように、一般的なモデルの特定の組織への適合・修正には、非常に多くのデータが必要とされるため、ほとんどの場合不可能である[KITCHENHAM1984]。そのため、モデルの適合・修正は、通常、様々な組織から提供されるデータに基づく部分的な改善にとどまる。結果的に、見積りモデルの改善は、プロセス改善や組織全体にわたる学習のような組織的な側面での取り組みが通常考慮されない。しかし、例えば、[JEFFERY1990]に述べられているように、見積りモデルの結果が思わしくない場合は、一貫性のない規模測定(例えば、プランク、コメント及び再使用された行の扱い)と関係があることがあり、情報収集プロセスの改善が必要となる。本研究のような、見積りモデルの反復的な構築は、組織に見積り手法を導入する際に、組織の課題の特定を可能にし、関連の測定過程とともにデータの完全性を向上させることができる。

6 結論

研究の主要な結果は、次のとおりである。第1に、反復的な見積りモデル構築が著しく見積り精度を向上させることを示した。図8及び図9に示すとおり、CoBRAモデルは、反復的なモデル構築を通じて、120%のMMREから14%まで継続的に改良した。第2に、CoBRAモデルは、典型的な日本のソフトウェア開発組織でも良好な結果を示した。3.5項で述べたとおり、日本企業という新しい条件の下で、非常によい結果を示した(14%のMMRE)。さらに、主なコスト要因を発見することができ、企業が適切な測定データを収集することを容易にし、特定されたコスト要因に関してプロセスの改善につなげられることを示した。CoBRAモデルの各反復的構築の間、要因モデルとともにデータの妥当性及び一貫性の議論を通じて、関連する情報収集プロセスも改善され、モデルが改良さ

れた。最後に、企業の専門家とともに、コスト及び生産性に影響を与える主要なコストを反映した要因モデルの構築に成功した。そして、CoBRA感応分析[BRIAND1998]の利用による生産性に対する最も大きい影響力を持つコスト要因分析の手段を得ることができた。

総括すると、今回の試行で、見積りモデルの漸次的な構築は効果を示し、見積り結果を著しく向上させることがわかった。さらに、CoBRAは、OLSのような単純なデータに基づいた方法と比べると、プロジェクトの特性が違つものを見積らなければならないときに有利である。

将来の研究として、本研究実施企業において見積りモデルのさらなる改良を行うことと、当該企業の新しいドメインや事業体に対して導入することが考えられる。方法論の見地からは、モデルの適用及びモデルの信頼性を向上させるために、専門家の知識とデータ駆動型の分析法を結合することは意味があると考えている（例えば、要因を決定するためにデータ駆動型を利用する）。見積りモデルの精度向上以外にも、今回実施した漸次的なアプローチは、企業の組織上のプロセスに良い効果があり、開発チームの知識及び経験を向上させる。他の見積りが計算精度のみを向上させる点と本質的な違いである（例えば、T. Menzies et al[MENZIES2005]）。

最後に、過去のプロジェクトデータへの適用時に改良されたモデルがよい見積り結果を出した場合、新しいプロジェクトの見積りに利用すべきと考える。そして、CoBRAモデルがプロジェクトを計画したりマネジメントするメンバから信頼が得られれば、意思決定支援システムとして利用可能である。ただし、モデルは、あくまでも見積りを支援するものであり、権威ある神託（オラクル）として使われるべきでない。本研究で示したとおり、妥当なコスト要因（モデルの精度にとって第一の重要なもの）を発見することは非常に重要である。さらに、コスト要因は時間が経つにつれ変化するので、モデルを見直す保守プロセスを確立することが必要である。本論文で示した反復的な方法は、見積りモデルの保守をサポートし、改善していくためにも有用と考える。

謝辞

CoBRA法の試行に積極的に協力していただいた沖電気工業株式会社ならびに同社の専門家・プロジェクトマネ

ージャに感謝します。そして、本事例研究に関して、積極的にバックアップしたIPAに感謝します。

参考文献

- [BASILI2001] Basili, V.R., Caldiera, G., Rombach, H.D. : Experience Factory; in: Marciniak J.J. (ed.), Encyclopedia of Software Engineering, vol. 1, John Wiley & Sons, 2001, pp. 511-519
- [BOEHM2000] Boehm, B.W., Abts, C., Brown, A.W., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Refer, D., Steece B.: Software Cost Estimation with COCOMO II, Prentice Hall, 2000
- [BOEHM1981] Boehm, B.W. : Software Engineering Economics, Prentice-Hall, 1981
- [BRIAND2002] Briand, L.C., Wieczorek, I. : Software Resource Estimation; in: Marciniak J.J. (ed.), Encyclopedia of Software Engineering, vol. 2, John Wiley & Sons, 2002, pp. 1160-1196
- [BRIAND1998] Briand, L.C., El Emam, K., Bomarius, F. : COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment, Proceedings of the 20th International Conference on Software Engineering, 1998, pp. 390-399
- [CHULANI1999] Chulani, S., Boehm, B., Steece, B. : Bayesian analysis for the Empirical Software Engineering Cost Models. IEEE Transactions on Software Engineering, vol. 25, no. 4, 1999, pp. 573-583
- [CMMI2001] CMMI Product Team. : CMMISM for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.1 Staged Representation. (CMU/SEI-2002-TR-004). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001
- [CONTE1986] Conte, S.D., Dunsmore, H.E., Shen, V.Y. : Software engineering metrics and models. The Benjamin/Cummings Publishing company, Inc., 1986
- [FARQUHAR1970] Farquhar, J.A. : A Preliminary Inquiry into the Software Estimation Process, RM-6271-PR, The Rand Corporation, August, 1970
- [FERENS1998] Ferens, D., Christensen, D. : Calibrating software cost models to Department of Defense Database: A review of ten studies. Journal of Parametrics, vol. 18, no. 1, 1998, pp. 55-74
- [GARTNER2005-1] Gartner Inc. : press releases, Gartner Survey of 1,300 CIOs Shows IT Budgets to Increase by 2.5 Percent in 2005, 14th January 2005, http://www.gartner.com/press_releases/pr2005.html
- [GARTNER2005-2] Gartner Inc. : press releases, Gartner Says Worldwide IT Services Revenue Grew 6.7 Percent in 2004, 8th February 2005, http://www.gartner.com/press_releases/pr2005.html
- [GENUCHTEN1991] Van Genuchten, M., Koolen, H. : On the use of software cost models, Information and Management, vol. 21, 1991, pp. 37-44
- [ISBSG2004] International Software Benchmarking Standards Group : The Benchmark Release 8, 2004, <http://www.isbsg.org>
- [JEFFERY1990] Jeffery, D.R., Low, G.C. : Calibrating estimation tools for software development, Software Engineering Journal, vol. 5, no. 4, 1990, pp. 215-222
- [KEMERER1987] Kemerer, C.F. : An empirical validation of software cost estimation models, Communications of ACM, vol. 30, May 1987, pp. 416-429
- [KITCHENHAM1984] Kitchenham, B.A., Taylor, N.R. : Software Cost Models, ICL Technical Journal, May 1984
- [LOTHER2001] Lother, M., Dumke, R. : Point Metrics. Comparison and Analysis. In: Dumke/Abran: Current Trends in Software Measurement, Shaker Publ., Aachen, Germany, 2001, pp. 228-267
- [MENZIES2005] Menzies, T., Port, D., Chen, Z., Hihn, J., Stukes, S. : Validation methods for calibrating software effort models, Proceedings of the 27th international conference on Software engineering, 2005, pp. 587-595
- [MUKHOPADHYAY1992] Mukhopadhyay, T., Vicinanza, S.S., Prietula, M.J. : Examining the feasibility of a case-based reasoning model for software effort estimation, The Management Information Systems Quarterly, vol.16, no. 2, 1992, pp. 155-171
- [PUTNAM1992] Putnam, L.H., Myers, W. : Measures for Excellence, Reliable Software on Time, Within Budget, Yourdon Press, Englewood Cliffs N.J., 1992
- [RUHE2003] Ruhe, M., Jeffery, R., Wieczorek, I. : Cost estimation for web applications. Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon, 2003, pp. 285-294
- [SPECTOR1992] Spector, P. : Summated Rating Scale Construction, Sage Publications, 1992
- [TSG2003] The Standish Group. : CHAOS Chronicles, West Yarmouth, MA, The Standish Group International, Inc., 2003

ソフトウェア工学国際会議 ICSE 2006 上海に参加して

SEC エンタプライズ系プロジェクト 研究員 神谷 芳樹

2006年5月に上海で開催されたソフトウェア工学の国際会議ICSE (International Conference on Software Engineering) 2006に参加した。ICSEはこの領域で毎年開催される最高水準の会議で、場所は米国をベースに隔年世界各国を持ち回る。世界各国から1,400人が参加し、アジア各国からの活発な参加もあり大盛況だった。開催場所は「上海国際会議中心」である。

論文発表には様々な部門があり、複数の部門にわたってSECと大学や海外研究機関との共同研究に関する発表が4件あった [TRENDOWICZ 2006][MITANI 2006][ABE2006][TSUNODA2006]。このうち1つは、ドイツ・フラウンホーファ実験的ソフトウェア工学研究所 (IESE : Institute of Experimental Software Engineering) との共同研究による発表だった [TRENDOWICZ2006]。筆者は今回ICSEで特別に設定された極東経験論文部門 (Far East Experience Track) のポスターセッションで発表した [MITANI2006]。

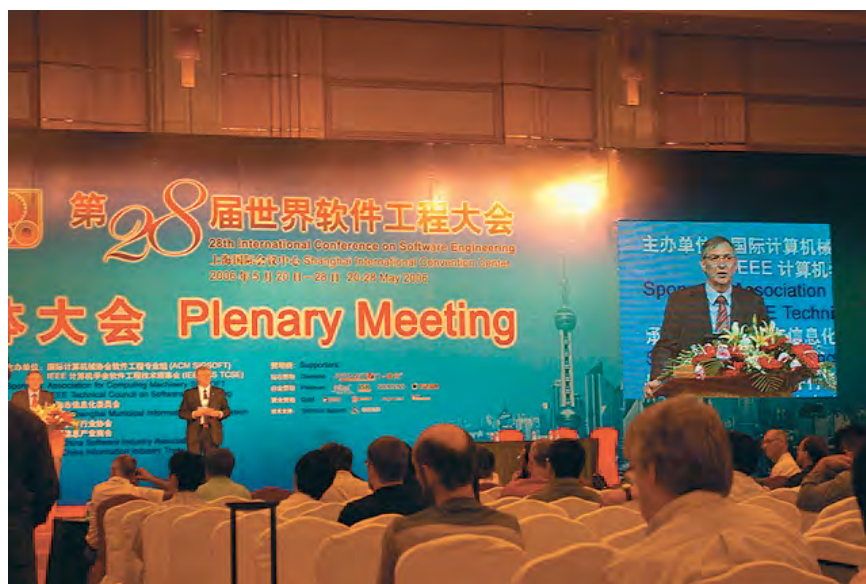
本会議での経験を通して、下記のメッセージを強く主張できるようになった。

「我が国の学界各位、SECのような産学官連携組織に集い、産業界のフィールドデータを産業界と共有しながら産学連携活動を進めてください。産学連携を進めれば、国際的にも評価される知見と論文を沢山生み出していくことができます。産学連携で共有したフィールドデータを背景とした研究は国際的に認められます。」

「我が国の産業界各位、これからのビジネスの展開には、日本がICSEのような国際イベントで国力に見合ったプレゼンスを示していく必要があります。そのために日頃の企業活動とかけ離れた『海外投稿』、『査読合戦』という審査競技への参加は必要ありません。そうではなく、SECのような産学官連携組織に集い、企業内で死蔵されているフィールドデータを提供してこれを学界と共有し、実践的なディスカッションに参加して

ください。このような産学連携活動が、学界での国際競争力のある学术论文の源となり、我が国の国際プレゼンス向上に貢献し、ビジネスのグローバルな展開に必要な土壌を育てます。」

会議については、会議予稿集や公式Webサイトに大量の情報があり、また参加者による様々な媒体への報告もあることから、以下に報告と主観も交えた率直な感想を示す。



D.Rombach教授の司会で基調講演するシーメンス社副社長のR.Achatz氏。

全般状況

産業界出身で、国際的な論文発表会議には、ごく最近になって参加するようになった筆者にとって、今回の会議は多くの意味で驚愕の内容だった。EASEプロジェクトやSECに参加して以来、これまで国際会議に数回発表者として参加し、いわゆるアカデミアの世界とプロフィットを迫る企業とは活動フィールドを共有しながらも、その振る舞いは、まったく別世界のものとの認識を深めていた。実際、国際会議での発表には多大の労力をかけて英文で投稿し、厳しい査読があり、首尾よく採択されても、カメラレディ原稿作成、ポスターあるいは発表用スライド作成、発表（口述）原稿作成、発表練習、そして会議のあるリゾート地等への数日間の出張と、一般企業では容認されにくい活動が必要となる。

しかしながら、今回のICSEのように各国から1,400人もが参加する国際会議となると、そこはオリンピックと大差ない国益のぶつかり合う社交・外交の場となり、アカデミアと産業界、そしてその背後にある各国政府・公的機関が密接につながった活動領域となる。今回のICSEは実質的に、組織委員会共同議長（Conference Coordination Committee Co-Chair）、現地共同議長（Local Chair）を勤めた岸田孝一氏（SRA先端技術研究所）の労によって誘致、実現されたものである。開会冒頭、総合議長（General Chair）のL.Osterweil教授が開会挨拶の中で岸田氏の名前を挙げてその功績を紹介し、謝辞を述べていた。

この会議は、論文の採択率が低く発表参加が容易でないため、岸田氏は特別にFar East Experience Trackを設け、ここにアジアの論文を誘致した。32件の投稿があり、このうち9編がフルペーパーとして採録された。さらにショートペーパーとして論文10編が選ばれ、ポス

ターセッション発表となった。これにより、筆者を含め日本のほか、アジアの若い人々も参加機会を得ることができた。Far East部門の発表論文の共著者数は117名、出身国は、日本、中国、米国、韓国、タイ、オーストラリア、シンガポール、ドイツ、香港、フィリピン、マレーシア、ベトナムの12カ国に及ぶ。この部門でSEC関連では、筆者らの発表のほかにSECと大阪大学との共同研究による発表が1件あった[ABE2006]。またSECから新谷勝利研究員がこの部門のプログラム委員として準備段階から参加し、会議では1セッションの議長を務めた。

ICSEは、全体で8つの発表部門と、併設ワークショップ19会議が設定されていた。このうち研究論文（Research Paper）部門が最も華やかな舞台となる。今回は370件の投稿のうち36編が採録された。ここに、先に述べたSECとIESEの共同研究成果が日本人の参加した唯一の論文として採録され、SECはそのプレゼンスを発揮できた。SECの論文共著者は、石谷靖、横山健次、菊地奈穂美の各研究員である。

今回の会議は、プログラム委員長（Program Co-Chair）を勤めたIESE所長兼カイザースラウテルン大学のD. Rombach教授がリーダーシップをとり、さながらドイツ会議の様相で、そうした国家レベルでは、残念ながら日本の存在感は全く感じられなかった。

中心となる本会議では、水曜から金曜の毎朝、90分程度の基調講演が行われた。初日は開会イベントに続いて北京大学F.Yang教授による「ソフトウェア工学の発展：産学官連携の効果」、2日目は米国南カリフォルニア大学B. Boehm教授による「20世紀と21世紀のソフトウェア工学の展望」、そして3日目はD. Rombach教授が司会を担当し、シーメンス社のソフトウェア・エンジニアリング担当副社長のR. Achatz氏による「ソフトウェア開発の最適化」だった。氏の講演は、「シーム

ンスは世界最大のソフトウェア企業である」という内容から始まり、この会議のプラチナスponsorであるシーメンス社のPRという印象だった。そして次々回の会議はドイツ開催となっている。

このほか、いくつかの招待講演があった。IEEEフェローV. Basili教授（メリーランド大学）のエンピリカルソフトウェア工学論文の書き方に関する講演等、興味深く、熱気があった。

会議を通して、基調講演、招待講演等で日本からは存在を強く示すものはなかった。全体に、岸田氏らの努力、地元中国のコミュニティの働きの上に、米国・オーストラリア・欧州各国が会議の中味を構成した構図である。米国を始めとして世界で活躍する中国人研究者の里帰り発表、同窓会の要素も強かった。日本は隣国での大規模な国際会議でありながら、国力に見合った存在を示すことができていなかった。この姿は、人口やGNPの規模、日常生活の中で体験するハイテク国家日本の姿とかけ離れて見えた。ICSEでは、かつて鳥居宏次教授（奈良先端科学技術大学院大学）がプログラム委員長を務め、日本企業から基調講演された会議もあったが、今では昔語りである。このような現象は今会議で初めて露見したのではなく、ICSEでの論文採択率が30%を切るようになってから顕著になったということである。主要大学を中心にその研究テーマをソフトウェア工学から他へ転換し、予算配分もこの現象を反映したものとなったようだ。また、これらと軌を一にして日本の産業界もICSEやソフトウェア工学そのものへの興味が薄れたということを知った。背景の1つに、この領域での産学連携が円滑でなく、国際競争力のある論文を生み出す研究環境の確保が極めて困難だったという現実がある。

今会議の位置づけ

今会議を実質的に運営したスポンサーは上海市情報化委員会（Shanghai Municipal Informatization Commission）及び上海ソフトウェア産業協会（Shanghai Software Industry Association）である。両組織は4年前にICSE誘致に成功して以来、数々のソフトウェア開発関連イベントを、今回の開催場所である国際会議中心等で実施し、会議運営の実績を積んできた。

今回1,400人が参加し、会議成否のボーダーラインで

ある1,000人を超えて、数の上では大成功であり、また海外で活動する中国人の参加が多数あり、あらゆるセッションで中国のプレゼンスが示されて、主催者はその目的を達することができた。

今会議のプログラムは以下のとおりで、全体でソフトウェア開発技術及びその研究に関する巨大複合会議を構成している[ICSE2006][ICSE2006CN]。

・5月20～23日、27～28日：

チュートリアル、併設ワークショップ、同時開催イベント

・5月23日夜：

ポスターセッション、レセプション

・5月24～26日：

ICSE本会議（24日夜：ポスターセッション、バンケット）

基調講演：3件、招待講演

論文発表トラック：研究論文、経験論文、極東経験論文、教育、挑戦と成果、博士課程学生のポスターセッション

会議予稿集はUSBメモリで配布され、ICSE本体の論文だけで1,082頁に達する規模である。

基調講演

3人の基調講演はそれぞれ特徴があり、興味深いものだった。以下、大略と印象に残ったところを中心に紹介する。

(1) Fuquig Yang教授（北京大学：1日目）

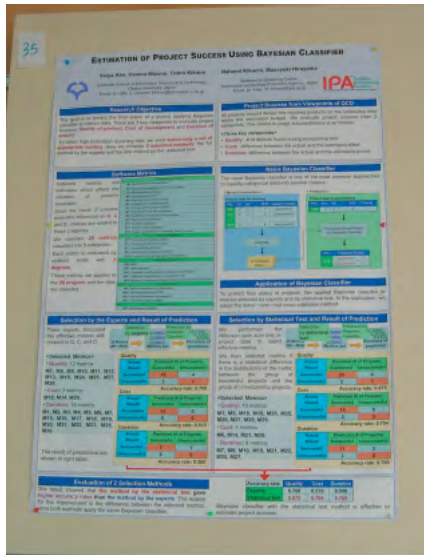
タイトル：「ソフトウェア工学の発展」

- 産学官連携の効果 -

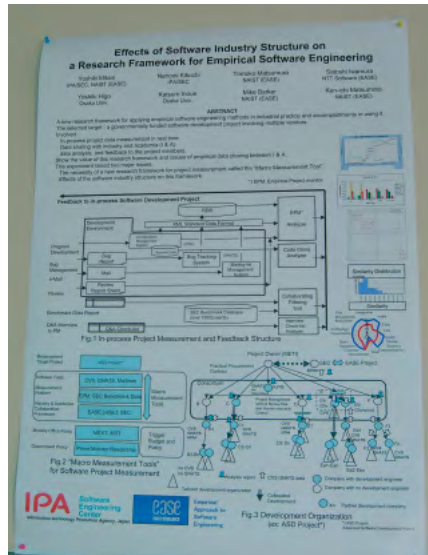
予稿集に10頁の論文が掲載され、この中に15枚の図が含まれている。全66枚のスライドが英語と中国語で用意され、会場の左右に表示された。講演は中国語で、英語への同時通訳があった。TVクルーの撮影もあり、講演後には質疑が受け付けられ、中国の新聞記者の質問等にも誠実に回答していた。スライドは事後ダウンロード可能となっている(PDF) [ICSE2006PDF]。

要旨：

過去40年間、ソフトウェア工学は計算機科学の重要なサブシステムとして発展してきた。ソフトウェア工



SECと大阪大学との共同研究のポスター。



SECとEASEプロジェクトとの共同研究のポスター。

いくつかのプロジェクトがあったが、その中にJade Bird（翡翠の鳥：神話の中の鳥）計画というソフトウェア工学環境の研究プロジェクトがある。この計画は1983年に第6次5カ年計画の中で始められた。そして2001年から2005年の第10次5カ年計画まで継続されている。北京大学を中心に20以上の研究所、300人以上の研究者と開発者が参加している。これまでに、版からJB1、JB2、JB3、そして現在開発中の版と5段階の研究が進められてきた。第10次5カ年計画では、

学の歴史について、ソフトウェアの推進力とソフトウェア工学のマイルストーンに主眼を置いて概観する。中国のソフトウェア工学の歴史について、とくにソフトウェア工学と産業界の関係に着目して振り返る。これらの概観に基づいて、ソフトウェア工学は計算機科学に沿って独立した学問になるべきだということと、ソフトウェア工学の調和ある発展には産学官連携が必要であるということを主張したい。

863の省レベルのハイテク計画、973の国家の基礎及び応用研究計画から資金提供されている。

ソフトウェア工学の歴史のマイルストーン

1940年代、1950年代：

マクロアセンブラのような初期の原始的ツール

1960年代：

FRTRANのような高水準言語の幅広い使用

1970年代：

初期のCASEツールのような高度開発ツール

1980年代：

PC時代の始まり

1990年代と21世紀の始まり：

ネットワーク時代

中国のソフトウェア工学の歴史

中国のソフトウェア工学は1980年に始まった。中国のソフトウェア産業の売り上げの世界シェアは2000年の1.20%（717万ドル）から2004年の3.55%（2,781万ドル）に拡大している。

中国における産学官連携

ソフトウェア工学は典型的な分野横断の学問である。多くの国でソフトウェア工学は、とくにIT産業を含む経済・社会からの影響をいろいろな形で受けている。

中国においては、ソフトウェア産業は国家経済の中で非常に重要な役割を果たし、経済成長に大きく貢献している。そこでソフトウェア工学から強くサポートしてほしいという要求がある。これに応えるために、中国の実情に合った調和のあるソフトウェア工学の発展モデルが実行されている。この発展モデルには主要な3つの力がある。中国政府、産業界、そして学界である。この3者がそれぞれの役割を果たしながら連携する中国のモデルは、エコシステムとして非常に効率的に運営され、産業界も学界もそれぞれの分野において発展していることを認めることができる。

(2) Barry Boehm教授（南カリフォルニア大学：2日目） タイトル：20世紀と21世紀の

ソフトウェア工学の一視点

予稿集に19頁の論文が掲載され、この中に11枚の図表、155件の参考文献が示されている。全54枚の講演のスライドは、事後ダウンロード可能となっている

(Power Point) [BOEHM2006].

要旨：

「ソフトウェア工学トレンドの全域」という図が示され、下記の時代区分と主要な時代の概念の関係、変遷が示された。

1950年代：(ハードウェア工学)

ハードウェア工学手法、SAGE、ハードウェアの効率

1960年代：(手工業)

手工業ソフトウェア、コードと修正、英雄的デバッグ

1970年代：(形式性、ウォーターフォール)

構造化手法、ウォーターフォール・プロセス、形式手法、領域理解

1980年代：(生産性)

オブジェクト手法、標準化、成熟モデル、ソフトウェア工場、4GL、CAD/CAM、ユーザープログラミング

1990年代：(並行プロセス)

並行プロセス、リスクドライブプロセス、領域特化アーキテクチャ、プロダクトライン再利用

2000年代：(アジリティ、バリュー)

アジリティ手法、ラピッド構成、進化型環境、ソフトウェア統合システム、ハイブリッドアジリティ、計画ドライブ手法、サービス指向アーキテクチャ、モデルドライブ開発

2010年代：(グローバル統合)

協調手法基盤環境、バリューベース手法、エンタープライズ・アーキテクチャ、ユーザによるシステム



ポスターセッションの様相。V.Basili教授の姿も。

構築

ソフトウェア工学の革新に対するヘーゲル的視点
弁証法的な流れが示された。主な項目を追うと、下記のようなになる。

1950年代：(テーゼ)

ハードウェアのようなソフトエンジニア

1960年代：(アンチテーゼ)

手工業としてのソフトウェア

1970年代：(テーゼ)

形式性、ウォーターフォール

1980年代：(シンテーゼ(統合命題))

生産性、再利用、オブジェクト、ピープルウェア

1990年代：(テーゼ)

計画ドライブソフトウェア、成熟度モデル

1990年代～2000年代の間：(アンチテーゼ)

アジリティ手法

2000年代：(シンテーゼ(統合命題))

リスクベース、アジリティ/計画ドライブハイブリッド、
モデルドライブ開発

2000年代～2010年代の間：(テーゼ)

統合ソフトウェアシステム工学

2010年代：(シンテーゼ(統合命題))

バリューベース手法、協調、グローバル開発、エンタープライズ・アーキテクチャ

その後：(シンテーゼ(統合命題))

自律、バイオコンピュータ

後段で、「システムとソフトウェアの未来」として
次の項目を挙げている。

8つの驚かないトレンド

1. システム工学とソフトウェア工学の統合の進展
2. ユーザあるいはバリューへの焦点合わせ
3. ソフトウェアの重大性と信頼性
4. ラピッド、加速する変化
5. 分散、移動性、相互接続性、グローバル化
6. システムにおける複雑システム
7. COTS(商用パッケージソフト)、オープンソース、再利用、レガシー統合
8. 多量計算量

2つのワイルドカード・トレンド

1. 自律ソフトウェア
2. 生物学とコンピューティングの組み合わせ

最後に、「永遠の原理」^⑧と「熟成する慣行」^⑨として、年代別に次の項目について述べられた。

1950年代から

- ⑧ 科学をおろそかにするな
- ⑧ 跳ぶ前に見よ（時期尚早な約束の回避）
- ⑨ 硬直した直列プロセスを回避せよ

1960年代から

- ⑧ 箱の外を考えよ
- ⑧ ソフトウェアそれぞれの相違の尊重せよ
- ⑨ カウボーイ・プログラミングを回避せよ

1970年代から

- ⑧ エラーを早期に摘出せよ
- ⑧ システムの目的を定義せよ
- ⑨ トップダウン開発と修正主義を避けよ

1980年代から

- ⑧ 生産性向上には沢山の道がある
- ⑧ プロダクトによいことは、プロセスにもよいことだ
- ⑨ 銀の弾丸に懐疑的であれ

1990年代から

- ⑧ 時は人間にとって金と価値なり
- ⑧ 人に役に立つソフトウェアを作れ
- ⑨ 機敏に、しかし慌てずに

2000年代から

- ⑧ 変化が早いときは、適合性が再利用性の切り札だ
- ⑧ あなたのすべての利害関係者の価値条件を考慮し、満足させよ
- ⑨ 君のスローガンに恋に落ちるな（例：YAGNI）

2010年代に向けて

- ⑧ 手を伸ばすのは制御できる範囲で
- ⑧ 出口戦略（脱出戦略）を持って
- ⑨ 読んだものを全部は信じるな
「これは真実だ、なぜならそれをインターネットで読んだから」

(3) Reinhold Achatz氏(シーメンス社副社長:3日目)
タイトル: ソフトウェア開発の最適化

予稿集に概要が示され、スライドは47枚で、事後ダウンロード可能となっている(PDF) [ICSE 2006PDF]。
要旨:

ソフトウェア開発リーダを抱えている多くの企業において、ソフトウェアは競争力決定要因として重要性が増している。シーメンスにおいても、そのビジネスの60%はソフトウェアの強い影響を受け、その特許の約50%がソフトウェアに関係している。シーメンスは世界中の30,000人以上のソフトウェア開発者と、300万ユーロ以上の費用をかけて、世界のリーディングソフトウェアカンパニーのチャンピオンリーグでプレーをしている。しかしこの事実はシーメンスで開発されるソフトウェアの多くが組込みのため、あまり知られていない。

シーメンスでは、ソフトウェア開発のコストが高く、事業上のインパクトが非常に高いために、ソフトウェア開発プロセスの最適化が、最優先の課題となった。シーメンスは世界で最も大きなソフトウェアカンパニーの1つであるという導入部に続いて、シーメンスのソフトウェアの特徴、ソフトウェアプラットフォーム戦略、標準化施策、プロセス計測の状況等がカラフルなスライドとともに力強く紹介された。

招待講演

いくつかの招待講演の中で聴講した論文査読に関する講演を紹介する。

(1) Victor Basili教授(メリーランド大学)と

Sebastian Elbaum教授(ネブラスカ大学)

タイトル: ソフトウェア工学のためのよりよいエンピリカル・サイエンス あなたのエンピリカルな研究がリジェクトされないようアドバイスしたい

予稿集は簡単な概要のみ、スライドは全59枚で、事後ダウンロード可能となっている(PDF) [BASILI2006]。

長らくIEEEの論文誌の査読委員長をしていたBasili教授の講演は、その人柄ゆえか多くの注目が集まった。

要旨：

最高のソフトウェア工学コンファレンスの開催地で十分なエンピリカルな仕事が見られない。最高のソフトウェア工学の現場において、著者と査読者をこの状況から助けることを目的に、ソフトウェア工学におけるエンピリカルな研究とは何か論じる。

また、エンピリカルな素材による論文の問題点と期待について討論する。

ソフトウェア工学におけるエンピリカルな研究とは定量的、定性的データをソフトウェア生産物とソフトウェア開発プロセスを理解し、発展させるために使うことであるという教授の熱弁が続き、実際の論文査読例が多数紹介され、熱気ある講演となった。

併設ワークショップの例

MSR2006 : International Workshop on Mining
Software Repositories

聴講したMSRというワークショップについて紹介する。MSRはICSEに併設された19のワークショップの1つ。22～23日の2日間で28編の論文発表とパネルディスカッション、それにMSR Mining Challenge Reportという話題を絞った類似テーマによる発表12編があった。これだけで予稿集の論文は186頁になる十分な規模の会議である。

15カ国から45件の投稿があり、このうち16編がフルペーパー、12編がショートペーパーとして採録されたということである。

比較的若い人の発表が多く、発表後、2～3の質問が出て答えて終わりという形式と異なり質疑・討論も活発で、そのための時間もかなり用意され、実質的なワークショップとしての運営がなされていた。なお、参加するには、相応の英語力が求められる。

本ワークショップの趣旨は、次のとおりである。

「ソースコードの制御システム、障害追跡システム、あるいはプロジェクト参加者間のコミュニケーションに関する蓄積情報のような、いわゆるソフトウェア・レポジトリが、ソフトウェアプロジェクトの進捗管理の支援に使われている。

ソフトウェアの開発者・研究者は、これらの情報の詳しい分析が、ソフトウェアシステムの保守支援、ソフトウェアの設計と再利用の促進、新しい奇抜なアイ

ディアや手法のエンピリカルな実証に対して潜在的な利点を持っていることに気が付き始めた。」

概略を見ただけでも、CVSあるいはSubversionのような構成管理システムのレポジトリを解析するものが9編、この中にはCVSと障害追跡システムのBagzilla、あるいはCVSとmailシステムのログを組み合わせで分析するものがあり、多くはSourceForge等で参照できるオープンソースプログラムとその開発レポジトリを分析の対象としている。

また、オープンソースプログラムの開発者の地理的な関係を分析した報告もあった。

このほか目にとまった発表として次のようなワークショップがあった。

- ・LinuxのDebianデストリビューションのコンパイル状況を7年間以上にわたって追跡し、全体のサイズ、パッケージのサイズと保守状況、プログラミング言語、ファイルサイズの推移を観測した研究。
- ・Nokia社内部のレポジトリを解析しアーキテクチャの進化を分析する可能性を追った研究。
- ・奈良先端科学技術大学院大学、同志社大学、SECの共同研究で、SECの収集した定量データベースの分析にかかわる報告

とくに生産性に関して、プログラムサイズ、外注率、上流工程の工数比率の影響についての分析に関する研究[TSUNODA2006]。

本ワークショップの中のMining Challengeというセッションはmining toolと手法に関する挑戦を興味の対象とし、今回は具体的にオープンソース・プロジェクトのPostgreSQLとArgoUMLを対象とした。12編のうち、4編がArgoUML、5編がPostgreSQL、そして3編が双方の分析であった。やはりCVSレポジトリの分析が大きな比重を占めている。日本からは奈良先端科学技術大学院大学の発表があった。本ワークショップで見られたCVSのような構成管理システム、そして障害管理システム、メーリングリスト管理システムからソフトウェア開発のプロセスとプロダクトの情報を収集して分析する手法は、2003年の日本のEASEプロジェクトが最初であったが、今ではすっかり普及したようにみえる。しかし、その研究の方向は、EASEプロジェクトとは大きく異なる。今回のワークショップにみられる研究対象は、もっぱらオープンソース製品のように、

すでに提供され、長期にわたって維持管理されている製品である。分析の横軸は年単位で長期に及ぶ。これに対してEASEプロジェクトやSECの先進プロジェクトでの計測対象は進行中の開発プロジェクトで、分析の横軸は日時、週次となり、分析結果を対象の開発プロジェクト内へ直接フィードバックすることを目指している。この違いは、諸外国においても進行中の実開発プロジェクトを対象とする研究環境の確保が容易でないことによると思われる。ここに、現在SECやEASEプロジェクトが確保しつつある、研究環境上のわずかな先進性を垣間見ることができた。

まとめ、大会の潜在テーマ「産学官連携」

今回の会議の大きな流れに着目すれば、「実証データを背景にものを論ずる」というエンピリカルソフトウェア工学の大きな底流をみることができる。会議をリードしたB. Boehm、D. Rombach、V. Basili、R. Jefferyの各教授、あるいは、あちこちで顔を見たドイツやメリーランドのIESEのメンバは皆エンピリカルソフトウェア工学の提唱者とその弟子たちということが出来る。3日目の基調講演者、シーメンス社副社長のR. Adchetz氏もIESEのメンバである。

そしてもう1つの大きな流れが、ソフトウェア工学研究における「産学官連携」である[MITANI2006-3]。ホスト国を代表した北京大学F. Yang教授の冒頭の基調講演のテーマは「産学官連携」だった。中国ならではの国家5カ年計画の中にソフトウェア開発環境に関する研究をきちんと埋め込み、20年以上にわたってその成果を積み上げてきたことが力強く報告された。そして今後の方向として、産、学、官それぞれの役割を果たしながら連携してそれぞれの成長を目指すことが述べられた。

また、プログラム委員長のD. Rombach教授のリードで強い印象を残したドイツは、もともと産官の関係が密であるという気風、州ごとの大学の地域密着性を背景として、連携を超えて産学官結託ともいえる振る舞いであった。

会議を通して、研究のトレンドだけでなく日本や各国のこの領域での国際的プレゼンスの度合いを肌で感じ、また街ではこの都市で進行中の驚愕の発展を目の当たりにし、同時に旧租界や革命の重い歴史に思いを



Research Paper部門でSECとの共同研究を発表するIESEのA.Trendowicz氏。

はせ、密度の濃い体験となった。そして現在推進中の、産学連携による産学での現場データの共有を基盤とした実践的（エンピリカル）ソフトウェア工学の方向の尊さを今さらながら認識させられた。

謝辞

ともにICSE 2006に参加し、本報告の作成を支援していただいた、石谷靖、菊地奈穂美、新谷勝利の各研究員に謝意を表します。

参考文献

- [ABE2006] Seiya Abe, Osamu Mizuno, Tohru Kikuno, Nahomi Kikuchi, Masayuki Hirayama: Estimation of Project Success Using Bayesian Classifier, ICSE2006 Far East Experience Track, pp.600-603 (大阪大学との共同研究)
- [BASILI2006] <http://www.cs.umd.edu/~basili/presentations.htm> (V.Basili教授の招待講演 (PDF))
- [BOEHM2006] http://www.icse2006.org.cn/index_e.asp?menu=4 (B. Boehm教授基調講演 (PowerPoint))
<http://www.isr.uci.edu/icse-06/program/keynotes/boehm.html#slides> (PowerPoint, 大会公式サイト)
- [ICSE2006] <http://www.isr.uci.edu/icse-06/>
- [ICSE2006CN] http://www.icse2006.org.cn/index_e.asp
- [ICSE2006PDF] <http://www.icse2006.org.cn/index.asp?menu=8>
- [MITANI2006] Yoshiki Mitani, Nahomi Kikuchi, Satoshi Iwamura, Yoshiki Higo, Ken-ichi Matsumoto, Tomoko Matsumura, Mike Barker: Effects of Software Industry Structure on a Research Framework for Empirical Software Engineering, ICSE2006 Far East Experience Track, pp.616-619 (EASEプロジェクトとの共同研究)
- [MITANI2006-3] 神谷芳樹, マイク・バーカー, 松本健一, 鳥居宏次, 井上克郎, 鶴保征城: 現場データを産学で共有するソフトウェア工学研究のための枠組み: 産学連携学, Vol.2.No.2 2006-3, 産学連携学会, pp.26-37
- [TRENDOWICZ2006] Adam Trendowicz, Jens Heidrich, Jurgen Munch, Yasushi Ishigai, Kenji Yokoyama, Nahomi Kikuchi: Iterative Development of a Hybrid Cost Estimation Model in an Iterative Manner, ICSE2006 Research Paper, pp.331-340 (ドイツIESEとの共同研究)
(Far East Experience Track (Poster Session))
- [TSUNODA2006] Masateru Tsunoda, Akito Monden, Hiroshi Yadohisa, Nahomi Kikuchi, Ken-ichi Matsumoto: Productivity Analysis of Japanese Enterprise Software Development Projects, MSR 2006 (International Workshop on Mining Software Repositories), pp.14-17 (奈良先端科学技術大学院大学, 同志社大学との共同研究)

* ()内の「共同研究」はSECとの共同研究を指す。

ソフトウェアプロダクトライン開発の概要

北陸先端科学技術大学院大学 情報科学研究科 特任教授 博士（情報科学）

岸 知二

製品系列としてのソフトウェアを効率的、効果的に開発する手法としてソフトウェアプロダクトライン開発（以下、プロダクトライン開発）が注目されている[CLEMENTS2001]。プロダクトライン開発は、製品系列全体の共通性と差異を明示的に捉え、再利用技術に基づき、全体最適の考え方をもって開発全体を目的指向で体系付ける先進的な形態である。こうしたプロダクトライン開発に関しては、欧米を中心に、携帯電話やデジタルテレビ等組込みソフトウェア開発での成功事例が多数報告されている[SEI]。本稿では、プロダクトライン開発の概要を紹介するとともに、2007年に日本での開催が予定されているソフトウェアプロダクトライン国際会議（SPLC : Software Product Line Conference）について触れる。

1. プロダクトライン開発とは

プロダクトライン開発の特徴として、以下が挙げられる。

第一に、開発される製品群の共通性や差異が、明示的に認識され、定義されていることである。共通性や差異は、機能だけではなく、性能や信頼性といった品質特性、OSやミドルウェアといった利用技術等、様々な側面に現れる。プロダクトライン開発ではこうした共通性や差異の明示的な分析や管理が重要となる。

第二に、製品群が、1つのビジネス上のコンテキストの下で開発されていることである。すなわち個々の製品が独立して開発されるのではなく、全体として最も効果的な開発をすることを目指すということである。例えば共通部品の利用は、個々の製品にとっては必ずしも最適化にはつながらないかもしれないが、共通化によって全体として開発コストが低減し、アドホックに開発することに比べて品質も安定するかもしれない。プロダクトラ

イン開発はこうした全体最適の視点に基づいてなされるものである。

第三は、それらが共通のアーキテクチャに基づいた再利用資産を活用して開発されるという点である[KISHI2005]。

再利用技術はソフトウェア工学における古くからの課題であり、様々な研究や実務上の取り組みがなされてきた。例えば再利用を行うためのメカニズムとしてオブジェクト指向技術におけるクラス継承等のしくみ、コンポーネントウェア等のミドルウェア等の技術が提案され、それらに基づいてクラスライブラリやフレームワーク等が開発され実際に使われている。

しかしながら現実には、大規模な再利用資産を活用することには様々な難しさが伴う。その大きな原因の1つがアーキテクチャ不整合である。再利用資産はそれがどのようなアーキテクチャの中で使われるかということ想定して作られている。アーキテクチャ不整合とは、再利用資産を使って作られるソフトウェアのアーキテクチャと、使われる再利用資産が想定するアーキテクチャとの不整合を指す。例えばイベントドリブンで使われることを想定している再利用資産は、データフローのアーキテクチャ中では使いづらい。再利用資産を活用して製品群を開発する際には、このアーキテクチャ不整合を起さないようにすることが重要となる。プロダクトライン開発は、製品群全体の開発や、それらに使われる再利用資産の開発を、全体最適の視点から総合的に捉えて進められるが、技術的に考えると、これはアーキテクチャ不整合を避けるための効果的、効率的な形態であるといえる。なお、プロダクトライン開発において、再利用の基盤となるアーキテクチャのことをプロダクトラインアーキテクチャと呼ぶ。プロダクトライン開発の成功事例の多くで、アーキテクチャの設計が重要なキーポイントだったと報告されている。

2. プロダクトライン開発の全体像

プロダクトライン開発は2つのフェーズから構成される(図1)。1つは再利用資産を開発するフェーズである。このフェーズでは特定の製品への要求ではなく、開発される潜在的な製品群への要求、すなわちそのドメインへの要求を分析し、プロダクトラインアーキテクチャを設計し、それに基づいた再利用コンポーネントを構築する。このフェーズのことをドメインエンジニアリングフェーズと呼ぶこともある。もう1つは、開発された再利用資産を活用して、特定の製品を開発するフェーズである。これをアプリケーションエンジニアリングフェーズと呼ぶこともある。

ドメインエンジニアリングフェーズで作られる再利用資産は、ライブラリやフレームワークといったソフトウェア部品だけでなく、要求分析の結果や、設計の結果等も含まれる。アプリケーションエンジニアリングフェーズにおいては、再利用可能な要求分析結果を活用して特定の製品の要求を整理し、再利用可能な設計結果を活用して設計を行い、さらに再利用可能なソフトウェア部品を活用して特定の製品を構築することになる。これら再利用可能な要求、設計、ソフトウェア部品間のトレーサビリティを確保することで、個々のアプリケーション開発を手続きに沿って行うことができる。

3. SPLC : プロダクトライン国際会議

プロダクトライン開発を実現するためには、要求分析、アーキテクチャ設計、再利用技術、テスト技術等の個別技術を、プロセス面、組織面等、多様な観点からの検討を踏まえて体系化することが必要となる。こうした意味から、プロダクトライン開発は、ソフトウェア工学の最も先進的な実践形態の1つと考えられる。

こうしたプロダクトライン開発の技術や実践に関する議論や情報交換を行うために、米国SEI (Software Engineering Institute) では、2000年よりSPLCを開催している。2005年からはそれ以前より欧州で行われていた同

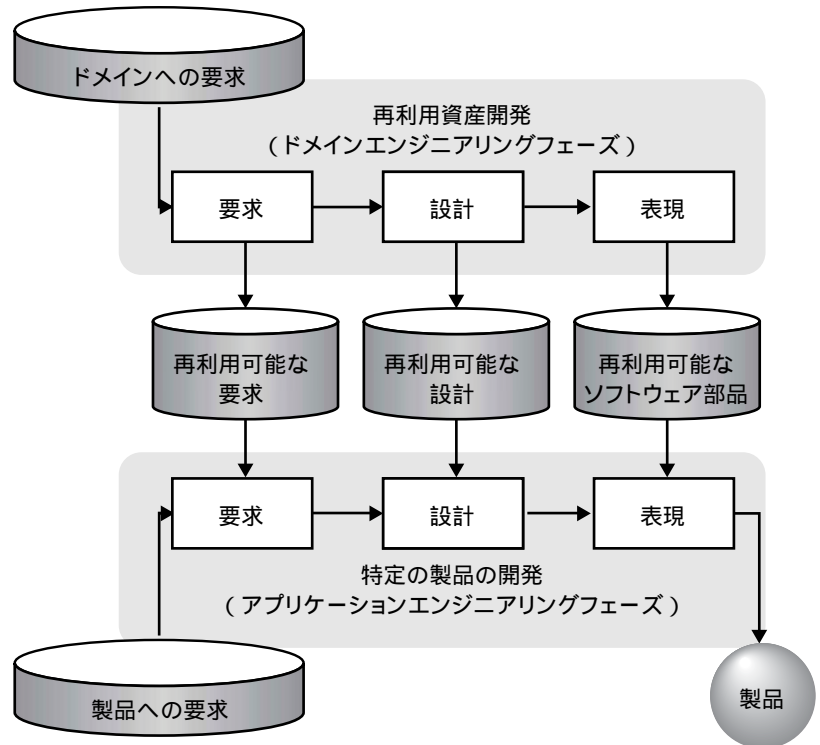


図1 プロダクトライン開発の全体像

様の会議であるPFE (Product Family Engineering) と統合され、名実ともにプロダクトライン開発に関する最も権威のある会議となっている。この会議の特徴は実務者と研究者がバランスよく参加しつつ、高い技術レベルでの議論を維持している点にあり、例年論文発表、ワークショップ、チュートリアル、パネル等多彩なプログラムが組まれている。

4. SPLC ASIAに向けて

2006年のSPLCは8月に米国で開催されたが、2007年は、初めてアジア地区で行われることになっており、韓国と日本が中心となり9月10日から14日に、京都での開催が予定されている。プロダクトライン開発のソフトウェア産業界への重要性を鑑み、IPA/SECには、本会議の開催を支援していただく。SPLC 2007では、例年通りのプログラムに加え、いくつかの業種にターゲットを絞ったワークショップ等、産業界のための企画も検討されている。積極的な参加・活用を期待するものである。

参考文献

- [CLEMENTS2001] Clements, P., Northrop, L.: Software Product Lines: Practices and Patterns, Addison-Wesley, 2001
- [KISHI2005] 岸知二, 野田夏子, 深澤良彰: ソフトウェアアーキテクチャ, 共立出版, 2005
- [SEI] <http://www.sei.cmu.edu/>

機能安全の枠組み

株式会社日本機能安全 取締役 上級コンサルタント
田邊 安雄

コンピュータが一般家庭で家電製品と同等の位置を占めるようになって久しいが、産業システムにおいては、実用性に富むことから安全目的でコンピュータ技術が使用されるようになってきている。

一方、1970年代のイタリア・セベソの化学工場や米・スリーマイル原子力発電所の事故に始まり、1980年代にはインド・ボパール¹の農薬工場やウクライナ・チェルノブイリ原子力発電所の事故のような大規模な産業事故を経験したことから、次第にリスクを前提とした安全確保の考え方に移行するようになった。

このように、これまでなかった環境や技術の台頭に呼応し、国際電気標準会議（IEC）は、コンピュータ技術による安全確保を実現するために、2000年に機能安全国際規格 IEC 61508[IEC61508]（日本では JIS C 0508[JISC0508]）を制定し、リスクを規範とした産業の安全確保の枠組みを与えた。規格は電気・電子技術やコンピュータ技術による安全確保を行う産業に分野規格の制定を促すことが制定目的の1つになっており、対象とする産業を特定していない。

コンピュータ技術は電子部品からなるハードウェア技術と、これを動作させるソフトウェア技術とからなる。規格は、安全関連系のハードウェア・ソフトウェアの設計指針であると同時に、業務遂行に関するマネジメントの規格でもある。規格は400頁もあり、膨大である。また、リスク評価、安全度水準、全安全ライフサイクル、機能安全評価、多くのハードウェア・ソフトウェア技法の導入等、これまでにない規格構造を有している。

一方、産業機械や自動車分野等、機能安全の考え方が様々な産業へと浸透しており、予測を超えた広がり方を見せている。また、英国、ドイツ等ではIEC 61508に対する認証を実施している。次第に、国内企業へも影響が現

れてきていることから、本稿では、機能安全規格の概要と認証動向を解説する。

1. IEC 61508

1.1 概要

IEC 61508はリスクに基づいた規格であり、ISO/IECガイド51[ISO GUIDE51]にその考え方が由来している。ISO/IECガイド51では絶対安全は存在しないことが述べられており、リスクは、「危害発生のがい然性と危害の過酷さの組み合わせ」と定義されている。すなわち、システムやプラント等に潜む潜在危険が何らかの要因によって発現した場合のリスクを想定している。リスクを軽減するための様々な安全関連系があるが、この中で、E/E/PES¹、すなわち、コンピュータ技術による安全関連系の設計・管理を対象としている。

一般にコンピュータの部品は故障する。また、使用するソフトウェアにはバグといった類のソフトウェアエラーがある。安全関連系を作動させる必要がある事態に、構成するコンピュータのハードウェア部品の故障やソフトウェアのエラーによって正しく動作しないときには、事故に至る可能性がある。IEC 61508では、このような安全関連系のシステム・サブシステムの信頼性をSIL²と呼ぶ指標に照らし合わせて評価することを要求している。

また、ISO/IECガイド51は、品質と安全は同義語ではなく、品質規格と安全規格の役割を混同すべきではないことを述べている。必ずしもすべての産業システムで品質が良ければ安全であるとは限らない。IEC 61508は、例えば品質問題等で事故に至るような事象が発生した場合に、安全関連系が作動して、リスクが大きくなるようにするための安全規格であり、品質規格を補完するも

1 E/E/PES：電気・電子・プログラマブル電子系（Electrical/Electronic/Programmable Electronic System）

2 SIL：安全度水準（Safety Integrity Level）

のであるといえる。

規格は、部品メーカー、サブシステム統合会社、エンジニアリング会社はもちろんのこと、システム運用会社等に広く関係する。このため、全安全ライフサイクルという考え方を採用し、概念・設計・保守・改修・廃却に至るライフサイクルを通じた業務管理手順を規定している。

1.2 構成

IEC 61508は全7部から構成され、翻訳規格であるJIS C 0508では全350頁からなる大きな規格である。

JIS C 0508：電気・電子・プログラマブル電子安全関連系の機能安全

・第1部 一般要求事項

全体の考え方がまとめられ、機能安全を達成するための管理及び技術上の要求事項をまとめた機能安全管理、概念・設計・改修・廃却にわたる機能安全の維持を目的とした全安全ライフサイクル、潜在危険及びリスク解析、E/E/PESの安全度水準、適合性確認、評価者の独立性を示した機能安全評価等が規定されている。

・第2部 電気・電子・プログラマブル電子安全関連系に対する要求事項

E/E/PESのハードウェアに対する要求事項であり、E/E/PESの設計に関わる要求事項を中心に機能失敗確率の推定法や安全度水準決定法等が述べられている。

・第3部 ソフトウェア要求事項

E/E/PESのソフトウェアに対する要求事項をまとめており、オペレーティングシステムからアプリケーションまで、安全度水準に応じたソフトウェアが使わなければならないことを述べている。

・第4部 用語の定義及び略語

機能安全に関わる用語の定義をまとめている。

・第5部 安全度水準決定方法の事例

安全関連系の信頼度、即ち安全度水準（SIL）を解析で求める方法をまとめている。

・第6部 第2部及び第3部の適用指針

第2部、3部の適用指針。プログラマブル電子系の信頼度を、アーキテクチャ構成をもとにして算出する事例をまとめている。

・第7部 技術及び手法の概観

第2部、3部に関係する安全技術・手法を紹介している。

1.3 機能安全

「機能安全」とは新しい用語であり、本質安全と対比する用語である。鉄道がよい例であるが、立体交差にすれば、踏み切りを渡って事故に遭遇する可能性はなくなる。このような対策によって根源からリスクを無くして達成する安全が本質安全である。しかし、一般にはすべてを立体交差にできないので、ある程度のリスクが残る。すなわち、そのままでは本質安全を達成できないような場合には、リスクが残るので、信号や列車停止装置等の周辺の安全機能により、相対的にリスクを軽減させ、許容されるリスク以下にして安全を確保することを機能安全と呼ぶ。

IECのホームページ[IECWeb]では、下記のように機能安全を定義している。

「機能安全とは入力に対して正しく動作するシステムや機器に依存する全体安全の一部である。機能安全は要求された全ての安全機能が機能するとき、達成される。」

1.4 安全関連系

プラントやシステムには制御系があるが、センサによって状態を監視し、異常があったときにプラントやシステムを停止する等の安全機能を有する場合、安全関連系とみなされることがある。なお、システムの安全はE/E/PESだけで確保される訳ではなく、電気・電子技術をまったく用いない機械系等他の技術による安全関連系や、防護壁等の外的リスク軽減施設を必要に応じて設置して確保されることもある。しかし、IEC 61508ではE/E/PESに基づいた安全関連系に関する安全管理や安全技術を規定し、他技術安全関連系や外的リスク軽減施設については対象外としている。

図1に示されるように、E/E/PESの全体システムは、ハードウェア及びソフトウェアからなる中核部分の安全コントローラと、センサやアクチュエータ等のサブシステムから構成される。従って、安全機能は個々のサブシ

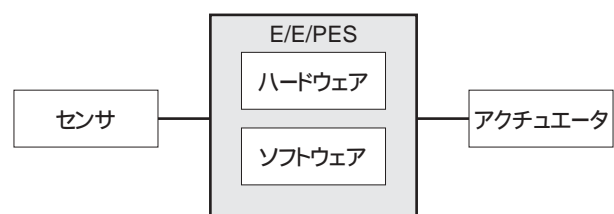


図1 典型的なE/E/PES安全関連系の構成

テムによってではなく、これらのサブシステムがすべて機能したときに正しく実現される。

1.5 全安全ライフサイクル

全安全ライフサイクルの構造概念を図2に示す。

英国では様々な産業で実際におきた安全関連系に関する事故34例を分析し、事故原因は図3に示すようにライフサイクルの様々なフェーズにあることが判明した[OOC]。従って、システムのリスク評価と、設計段階から製造・運転・保守・改修、さらには廃棄段階までの16フェーズにわたる全安全ライフサイクルの安全管理が重要との結論に至っている。こうした背景から、IEC 61508は文書での引継ぎ・引渡し事項が重視され、全安全ライフサイクルやハードウェア・ソフトウェアライフサイクルにおいて系統的な業務管理が行われる。

1.6 故障と安全度水準

安全に関わる故障は、ランダムハードウェア故障と決定論的原因故障に分類される。ランダムハードウェア故障は、主にハードウェアに関し、部品や材料の劣化等に

より起こるものであり、いつ発生するかはわからないが、その発生頻度が定量化できる類の故障である。ランダムハードウェア故障に対しては、故障を検知（診断）し、故障を隠す（多重系）技術等が採用され、決定論的原因故障に対しては故障を回避し、故障を制御する技術が採用される。

安全装置のSILを決定するためには、FTA（フォールトツリー解析）やETA（イベントツリー解析）による定量的な確率論的評価や、リスクグラフやリスクマトリクス等の定性的なリスク分析を行い、全体システムのリスクが許容リスクを下回るように必要とされるSILが決定される。この作業をSILの割り当てと呼んでいる。

IEC 61508では、表1に示す4段階の数値目標をそれぞれのSILに対して定めている。SILが大きいほど、リスクはより軽減されることを示す。

一方、決定論的原因故障は、設計の誤りや製造ミス等、発生してみて初めてわかるような定量化できない類の故障である。これは、前述した全安全ライフサイクルの16フェーズにおける安全評価・対策や文書化等によって、度重なる発生を防ぐことができる。

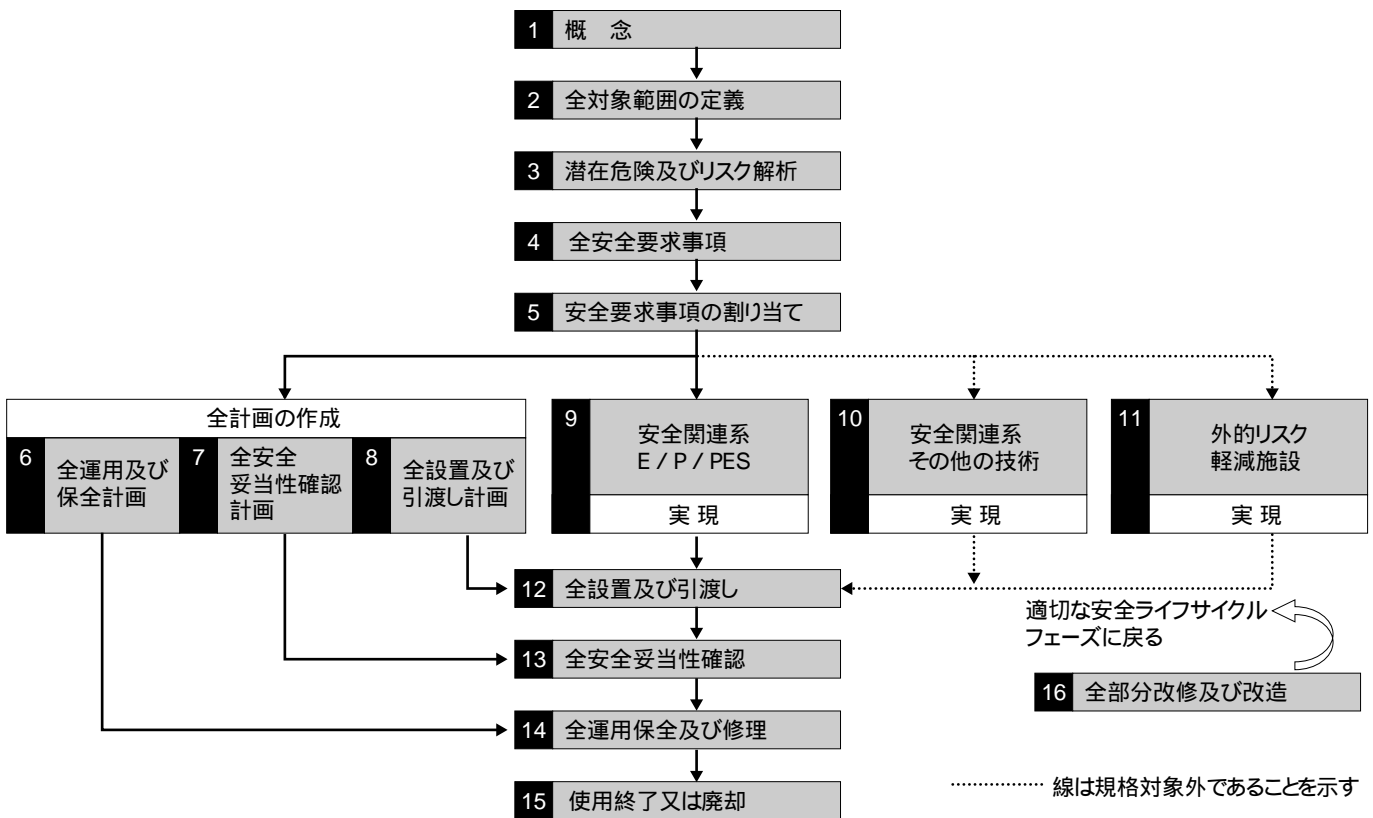


図2 全安全ライフサイクル

表1 安全度水準：E/E/PES安全関連系に割り当てられる安全機能に対する目標機能失敗尺度

SIL	低頻度作動要求モード 運用（注1）	高頻度作動要求または 連続モード運用（注2）
4	10^{-5} 以上 10^{-4} 未満	10^{-9} 以上 10^{-8} 未満
3	10^{-4} 以上 10^{-3} 未満	10^{-8} 以上 10^{-7} 未満
2	10^{-3} 以上 10^{-2} 未満	10^{-7} 以上 10^{-6} 未満
1	10^{-2} 以上 10^{-1} 未満	10^{-6} 以上 10^{-5} 未満

注1 作動要求当たりの設計上の機能失敗平均確率
注2 単位時間当たりの危険側故障確率[時間]

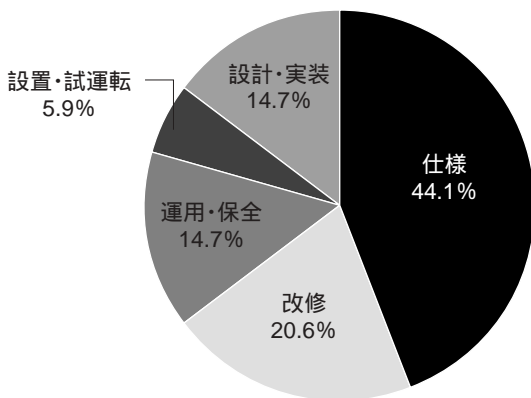


図3 制御系故障原因の分析[OOC]

2. 機能安全の実現

次に、化学プラントの反応容器の事例をもとにして、機能安全がどのようにして達成されるかについて示す。

2.1 安全度水準の割り当て

反応容器のリスク評価をもとにしたE/E/PES安全関連系の設計事例を紹介する。図4は内部で化学反応を行う反応容器を示す。内部の液位を監視し物質の流入を制御する制御弁から構成される基本プロセス制御系と、何らかの異常により内部圧力が上昇した場合に、反応容器が破壊しないよう圧力を監視する圧力センサ（ここでは、3つ）、プログラマブル電子機器、遮断弁から構成されるE/E/PE安全関連系及び機械的な安全関連系の逃し安全弁からなる防御系から構成される。

最初にリスク目標が定義される必要があるが、ここでは、定量的な実施例であるが、「1年に 10^{-3} 以上の頻度で反応容器からガスが放出されない」とした。このシステムにおいて過圧事象は1年に1回発生すると仮定し、図5に示すイベントツリーによって事故シナリオを定量化する。

事故対応として考慮できる警報や運転員対応及び機械

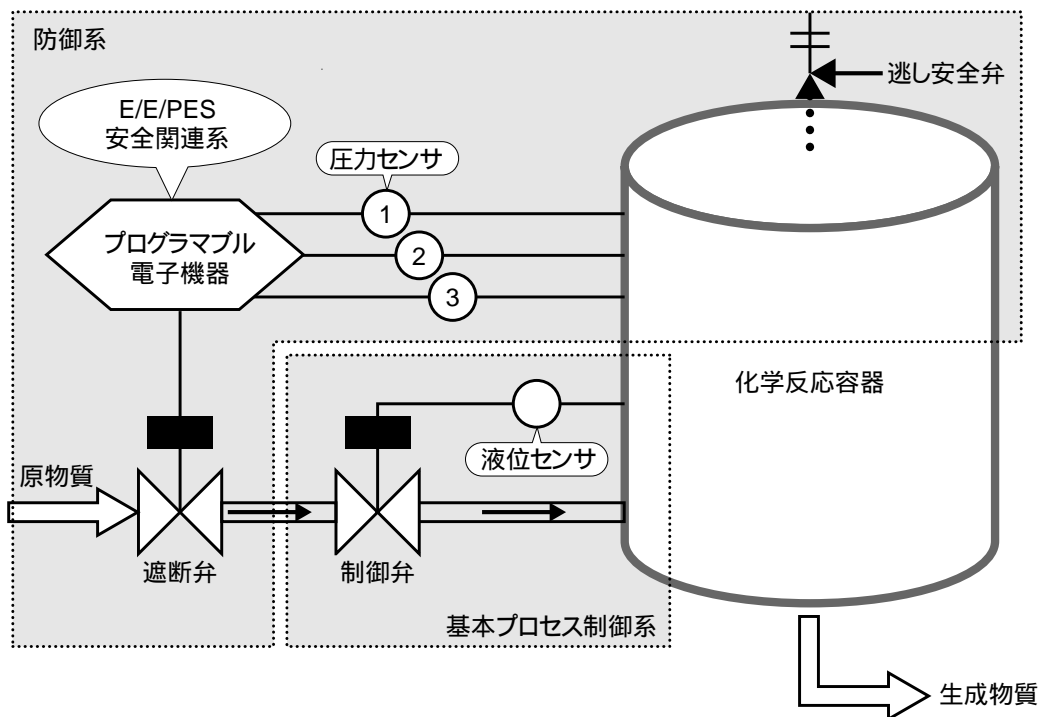


図4 E/E/PES安全関連系

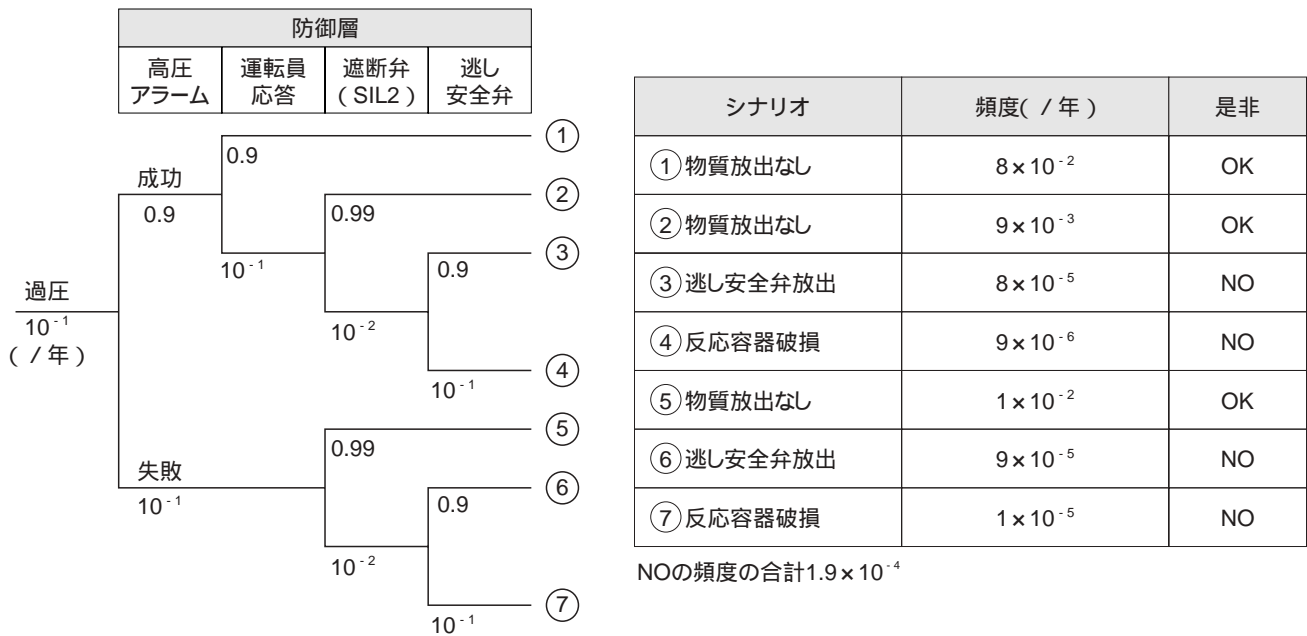


図5 イベントツリー解析

的な安全関連系である逃し安全弁の作動失敗確率をそれぞれ0.1とする。これらは反応容器の運転や設計に携わったことのある技術者の経験値でもよい。

さらに、2桁以上リスク軽減を図れる機能失敗平均確率が 10^{-3} から 10^{-2} の範囲にあるSIL2のE/E/PES安全関連系である遮断弁を設置し、圧力異常を圧力センサによって検知した場合には、逃し安全弁が作動する前に反応容器への物質の流入を遮断する。この場合には、7つのシナリオが考えられ、放出の頻度をイベントツリーで計算すると、1年に 1.9×10^{-4} となり目標値を満足する。

なお、本事例では、リスクと安全関連系のSILの関係を明示的に示すことを目的にしており、評価は簡略化しているが、実際には共通原因故障を考慮しなければならない場合があることに注意する必要がある。

2.2 安全度水準の実現

安全関連系に対する安全機能と安全度水準への要求(即ち、SILの割り当て)がリスク評価で決定されると、これを安全関連系の設計や運用で実現する。ハードウェアは、ランダムハードウェア故障に対応したハードウェア安全度水準要求事項と決定論的原因故障に対する要求事項を満足するように設計する必要がある。

2.2.1 ハードウェア安全度水準要求事項

ハードウェア安全度水準要求事項については、ハード

ウェア安全度水準に対するアーキテクチャ上の制約事項と危険側ランダムハードウェア故障確率の要求事項を同時に満足する必要がある。

(1) アーキテクチャ制約

(a) 全体システムのハードウェア制約

図1に示した安全関連系の構成では、最も低い安全度水準を有するサブシステムで全体システムの安全度水準が制約されることから、全体システムでSIL2を実現するために、センサ、安全コントローラ、最終要素(遮断弁)のいずれかはSIL2以上となる必要がある。

(b) サブシステムのハードウェア制約

安全関連系の個々のサブシステムアーキテクチャ例を図6に示す。

安全機能を達成するために要求されるサブシステムに使われる部品の信頼度等によって、使用する部品がタイプAかBの製品として分類する。部品の素性がより明確な場合にはタイプA、そうではない場合にはタイプBとなる。タイプA、Bの安全関連サブシステムのアーキテクチャ制約によるハードウェア安全度水準を表2に示す。次に、サブシステムの安全側故障割合(故障してもリスクの発現に至らない故障の全体の故障率に対する割合)とフォールトトレランスとの関係でSILが決められる。

(2) 作動要求あたりの危険側故障確率

(1)項のアーキテクチャ上の制約に加えて、作動要求あたりの危険側故障確率が目標とするSILを同時に満足

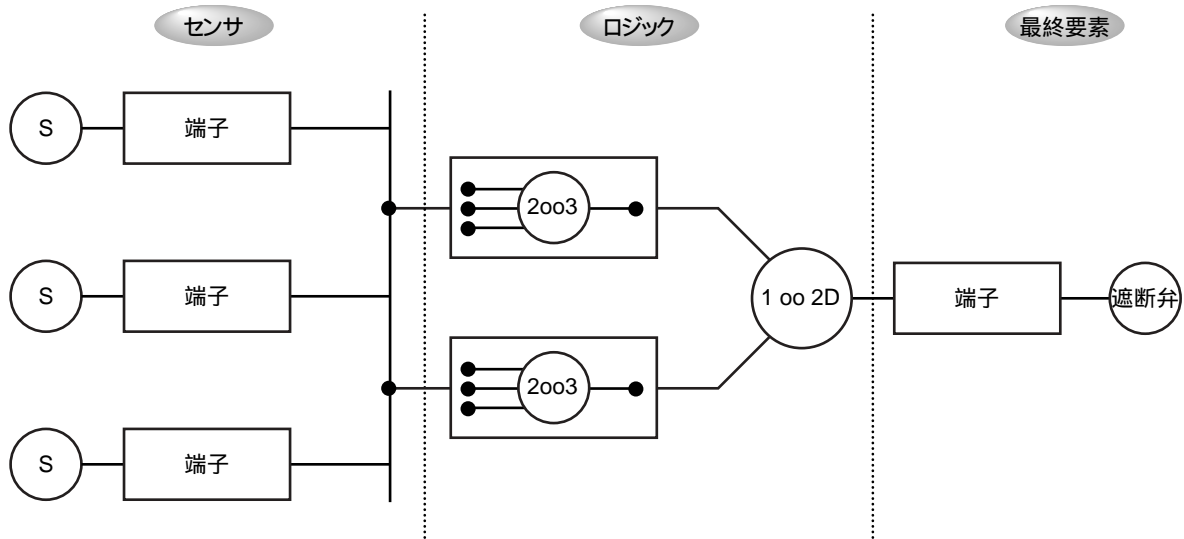


図6 安全関係のアーキテクチャ例

表2 ハードウェア安全度水準：安全関連サブシステムのアーキテクチャ制約

安全側故障割合	ハードウェアフォールトトレランス(タイプA/タイプB)		
	0	1	2
< 60%	SIL1 / 認めず	SIL2 / SIL1	SIL3 / SIL2
60% < 90%	SIL2 / SIL1	SIL3 / SIL2	SIL4 / SIL3
90% < 99%	SIL3 / SIL2	SIL4 / SIL3	SIL4 / SIL4
99%	SIL3 / SIL3	SIL4 / SIL4	SIL4 / SIL4

ハードウェアフォールトトレランス N は $N+1$ 個のフォールトが安全機能の喪失を引き起こすことを意味する。

する必要がある。

ここで、危険側故障確率は安全関連系の作動要求モード、プルーフ試験間隔、平均修復時間の条件を設定した後、個々のサブシステムの故障率、診断試験で検知されない故障の共通原因故障割合、診断試験で検知される故障の共通原因故障割合 D 及び自己診断率 DC を用いて計算される。

(1) (2) 項で求めたSILの小さい方で全体システムのSILが決定される。

2.2.2 決定論的原因故障に対する要求事項

IEC 61508は故障についてランダムハードウェア故障と決定論的原因故障の2つがあることを、1.6項で述べたが、ランダムハードウェア故障のハードウェア対策は決定論的原因故障に対してはほとんど役に立たないといわれている。また、ソフトウェアのエラーは決定論的原因故障に分類される。そこで、さらにハードウェアとソフトウェアの決定論的原因故障の要求事項を満足することが要求される。

図7に示すように、E/E/PESの設置前及び設置中に、ハードウェアには製造で組み込まれた故障や部品選定時の故障要因があり、ソフトウェアには仕様の誤りやバグ等のソフトウェア故障があり得る。一方、設置後においてはハードウェアにはランダムハードウェア故障の他に誤使用も考えられる。そこで、規格ではハードウェアについてのSILに応じた故障回避の手段が第2部付属書Bに、運転中の故障抑制手段が第2部付属書Aにおいて、合計で100以上示されている。ソフトウェアの技法及び手法についても同様に、第3部付属書AにSILに応じて100以上の手法が提示されている。これらの手法は、ハードウェ

時期	ハードウェア	ソフトウェア	
設置前	ハードウェア ・製造フォールト ・部品選定ミス	ソフトウェア ・仕様 ・バグ	技法及び手法選択 (第3部付属書A)
設置中			
設置後	ヒューマンエラー ・誤使用	故障回避手段 (第2部付属書B)	運転中故障抑制手段 (第2部付属書A)

図7 フォールトが発生した場合の故障要因と手法

ア及びソフトウェアライフサイクルのそれぞれのフェーズにおいて、要求されているSILに応じて使い分けることとされている。

このように、ランダム故障に対処するハードウェア構成とともに、ハードウェア・ソフトウェアについての技法をライフサイクルの各フェーズで採用することで、SILを実現・維持できる。

3. 妥当性確認と適合確認

適合確認 (Verification) とは、「要求事項が満足されていることを調査して客観的証拠を提示することによって確認すること」であり、解析や試験等によってそのライフサイクルフェーズへの引き渡し事項が、そのフェーズの目的と要求事項に整合していることを保証する審査のことをいう。例えば、設計レビューや確認試験や環境試験等がある。

一方、妥当性確認 (Validation) は仕様で定められた使用方法に関する特定の要求事項が満足されていることの審査と客観的な証拠の提供による確認のことをいう。

適合確認がフェーズごとに実施するのに対して、妥当性確認はいくつかのフェーズをまとめた後で実施することも可能である。

4. 機能安全評価と機能安全監査

IEC 61508では、適合することを調査するための機能安全評価を実施することが要求されている。

規格では、機能安全評価は、「根拠に基づいて、一基以上のE/E/PE安全関連系、他技術安全関連系又は外的リスク軽減施設によって機能安全が達成されることを判定するための調査」とあり、機能安全評価者を1名以上任命する必要がある。

表3は機能安全評価者の独立性の程度を示したものである。E/E/PESの安全度水準の程度、または災害の過酷度に応じた独立性が次第に厳しく要求されていることに注意する必要がある。

機能安全評価者は、IEC 61508の要求事項と照らし合わせて、受容、条件付受容、拒否のいずれかの勧告を行う役割を担う。

また、機能安全監査は、「計画された定めに基づく機能

表3 機能安全評価を実施する者にかかる独立性の最低限の水準

独立性の最低水準	被害の程度(備考参照)			
	A	B	C	D
独立した人員	HR	HR ¹	NR	NR
独立した部門		HR ²	HR ¹	NR
独立した組織			HR ²	HR

備考:典型的な障害の程度は以下による。
 (A~DをSIL1からSIL4の安全度水準としてもよい)
 A: 一時的な機能喪失
 B: 一名以上の深刻な永久障害又は一名の死亡
 C: 数名の死亡
 D: かなり多数の人命損失
 注)NR:推奨されない
 HR:推奨される(一般論としてHR¹よりHR²が推奨される)

安全要求事項に定められた手順が効果的に実施されているか、さらに定められた目的の達成に適切であるかを決定する系統的、かつ、独立した審査」とあり、評価者は監査者を兼ねることができる。従って、これらの評価者・監査者は、機能安全についての知識が必要であり、独立性が必要になる場合がある。

5. 機能安全認証

ドイツや英国ではIEC 61508に適合していることを証明する認証が開始されている。ドイツでは伝統的に製品認証が行われており、安全装置に対してもDIN規格に対する認証が行われていたが、1990年の中頃から、認証機関であるTUVがIEC 61508に対する認証を行っていた。IEC 61508は製品の規格であるとともに、マネジメントの規格でもある。英国では、1980年代から国内で機能安全に関する認証制度の枠組みを多くの産業界が参加して開発してきた。この認証制度はCASS (Conformity Assessment of Safety related System) と呼ばれており、下記と図8に示すタイプ1からタイプ5までの認証タイプが考えられている。

- タイプ1 : 汎用部品評価 (PLC用部品、スマート計器、ソフトウェア等)
- タイプ2a : 統合システム評価 (緊急時停止系、火災防護系等の特定用途向け)
- タイプ2b : サブシステム評価 (タイプ2 aのうち、センサ・アクチュエータを除く)
- タイプ3 : 運用・保守評価 (E/E/PES運用・保守・改修)

タイプ4 : 安全要求事項評価 (安全要求事項、潜在危険・リスク評価)

タイプ5 : 組織の機能安全能力評価 (FSCA: Functional Safety Capability Assessment)

ここで、タイプ5は個々の製品やシステムが対象となる審査ではなく、部品メーカ、エンジニアリング会社、運用会社等多くの分野に共通し、組織が機能安全を達成できる能力を有するかどうかについての審査である。

認証を実現するために、非営利組織であるThe CASS Scheme Ltd.を発足させ、すべての産業界からの代表が入った技術委員会がルール作りを行っている。産業界からの審査員を登録する仕組みもあり、これらの審査員は、どの認証機関も利用でき、どのような産業の認証も可能なようにしている。なお、審査員には独立性、中立性が強く求められ、事前に被審査側の同意が必要である。英国では、2001年から認証機関Sira (Sira Test & Certification Ltd.) が認定機関のUKASからIEC 61508に関する認証機関として認定され、現在、認定されたタイプ2、5の適合認証を開始している。また、継続して残りのタイプの認証の実績も積み、UKASの認定に備えている。英国には、国際規格に適合していることを国が認定した認証を実施することで、世界の市場に参入し易くするという戦略がある。このような、広い枠組みを有する認証制度は、今後、国際的に共通した認証の枠組みになり得ると考えられる。ドイツのTUVにおいても、製品に加えてマネジメントの認証も行うようになっており、適合認証の方向性を示している。

6. まとめ

機能安全は1980年代の大きな産業事故の発生とコンピュータ技術の普及と密接に関係して登場した。IEC 61508を産業システムに適用した場合、故障があっても安全機能が喪失しないように、安全関連系のハードウェアにはフォールトトレラントな技術を導入する必要性が生じる。また、安全規格の中で、初めてソフトウェアが扱われている。コンピュータ技術を安全目的で使用する場合に、多くの産業は、IEC 61508の規定の下で設計・管理が要求される。IEC 61508は任意規格である。拘束力はないが、海外では強制規格に近い捉え方をしている。英国ではHSE、DTI等政府機関・規制機関の支援により、CASSスキームという全産業を網羅する機能安全認証の仕組みがある。製品規格とともにマネジメントの規格である特徴は、これまでに存在していない。この特徴を踏まえて英国やドイツでは製品認証に加えてマネジメントの認証を行うようになってきている。IEC 61508は、ASICのような新技術に対しても、2008年制定予定の改訂版で取り扱おうとしている。さらにISOにおいても、自動車分野での機能安全規格が審議されていることや、ここ数年、海外でのIEC 61508への関心の急激な高まりにも注目すべきである。

したがって、国内でも、産・学・官が一体になり、新しい技術動向をフォローし、海外の動きに遅れないようにするべきである。

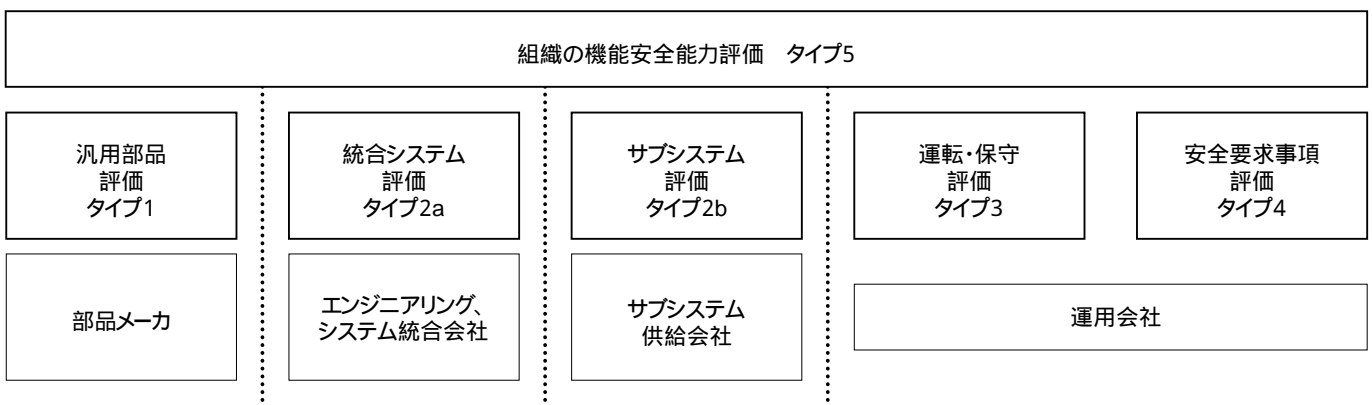


図8 機能安全認証のタイプ

参考文献

[IEC61508] IEC 61508, Functional Safety of Electrical/ Electronic/Programmable Electronic Safety Related Systems, 1998
[JISC0508] JIS C 0508, 電気・電子・プログラマブル電子安全関連系の機能安全, 日本規格協会, 2000

[ISO GUIDE51] ISO/IEC GUIDE51, 1999
[IECWeb] IECホームページ: http://www.iec.ch/zone/safety/pdf_safe/brochure.pdf
[OOC] OUT of CONTROL, HSE Books, 1995

最近のソフトウェアファクトリ論

アングル

財団法人 京都高度技術研究所 顧問
松本 吉弘

1. まえがき

1960年代以降、ソフトウェアファクトリは、事業として実施されるソフトウェア開発・保守における品質及び生産性向上のための手法、さらにそれを実践するための設備・組織体制として、企業は、それぞれなんらかの形でこれを実現してきた。ソフトウェアファクトリには、第2節で説明する歴史的背景と定義が存在する。今世紀に入ってから、ソフトウェアエンジニアリング・ビジネス論 (software engineering as business¹) や、ソフトウェア・サービス論 (例えば、SaaS: Software as a Service²) 等の刺激を受けて、ソフトウェアファクトリには新しい動きが見られるようになってきた。筆者はこの動きを、モデルを基本とするソフトウェアファクトリMBSF (Model-Based Software Factory) 及びセル方式ソフトウェアファクトリCBSF (Cell-Based Software Factory) に分けて説明する。いずれも分散配置されたソフトウェアファクトリを、ネットワークを通して統合し、顧客³に対してオンラインで開発・保守の対象となるプロダクトを提供する方向を志向している。第3節で、MBSF及びCBSFがもつ主たる性質について、比較しながら説明する。これらソフトウェアファクトリの基本となる思想は、フィーチャ指向アプローチであり、特定された領域の問題フィーチャに基づいて、ソフトウェア資産 (固定資産: fixed assets及び可変資産: variable assets) を開発し、顧客の要求に応じて可変資産をカスタマイズする。MBSF及びCBSFは、いずれもイノベーション段階及びコモディティ段階から構成されるファクトリ・ライフサイクルに沿って運用される。第4節では、ソフトウェアファク

トリを計画するにあたって前提となる経済モデルについて説明する。第5節ではソフトウェアファクトリに関して展開されている議論の主なものを紹介する。

2. ソフトウェアファクトリの歴史と定義

1968年、ドイツ・ガルミッシュにおいて、第1回ソフトウェアエンジニアリング・ワークショップが開かれ、「ソフトウェアエンジニアリング」という語が初めて铸造されたとき、McIlroyが、再利用によってソフトウェア生産性、品質を向上させるための組織を、ソフトウェアファクトリと呼ぶことを提唱した [MCILROY69]。

その後まもなく、米国にあって、AT&TやIBM等がこれを実用しようとしたが、時期尚早のため数年後に終焉を迎えた。その後、Basiliらは、Experiment Factory (EF) と称する「経験」の再利用によってソフトウェアの生産性、品質を向上するための組織を開発し、国防省傘下の一部のソフトウェア開発現場に適用して、成果を挙げた [BASILI89]。

わが国では、1960年代末から1970年代初めにかけて、電機、電子、通信メーカ各社が、それぞれのソフトウェア事業所をソフトウェアファクトリ (またはソフトウェア工場) と位置づけ、独自のマネジメント体制を確立し、現在に至った [CUSUMANO91]。

ヨーロッパは、1980年代に入ってから、Eureka Software Factory (ESF) と称する構想を発表した [AAEN97]。ここでは、ソフトウェアファクトリは、プログラミング品質及び生産性を向上させるために必要な標準的なツール、成果物及びマネジメントデータを管理す

1 IEEE Software, Vol.18, No.6 (2002) The Business of Software特集

2 http://en.wikipedia.org/wiki/Software_as_a_Service

3 この解説全体を通して、「顧客」は、特定または不特定のユーザに対してプロダクトまたはサービスを提供する事業主で、ソフトウェアファクトリからソフトウェアプロダクトまたはサービスの開発・保守を調達する組織を指す。顧客とユーザが同一の場合もある。

るためのデータベースを共有する、ソフトウェア開発・販売・工程管理・原価管理・調達管理・品質管理を統合した組織である、としている。

2000年代の初めに、Microsoft Corporationは、TSF⁴、EF、ESF、CMMI⁵を参照し、これらからの発展として「ソフトウェア・ファクトリーズ (Software Factories) [GREENFIELD04]」を提案した。これは、MBSFの基本的な性質を備えた構想である。Microsoft Corporationでは、MBSFの1つとして、ソフトウェアファクトリを次のように定義している。1つのソフトウェアファクトリとは、1つのプロダクトラインのことである。プロダクトラインは、ソフトウェアファクトリ・スキーマに基づいたソフトウェアファクトリ・テンプレートに従って、拡張可能なツール、プロセス及びコンテンツを選定し、フレームワーク部品を適用、組立て及び形成することによって、アーキタイプ・プロダクトの可変部開発及び保守を自動化するためのものである。

2005年、筆者は企業十数社の協力を得て、CBSF構想をまとめた。CBSFでは、ソフトウェアファクトリを次のように定義している。CBSFは、事業として開発・保守するソフトウェアの適用目的領域を特定し、その中で蓄積される知識、スキル、経験、固定ソフトウェア資産、可変ソフトウェア資産及びツールを利用し、ソフトウェア生産セルのモジュラな連携によって形成されるプロジェクトによって、受注したソフトウェアシステムを工業的、

または半工業的に開発及び保守するための事業組織であり、ハードウェア生産セル⁶に相似する形態である [MATSUMOTO06]。

ここで、1つのソフトウェア生産セルとは、ソフトウェアプロジェクトに与えられた問題に対する、1つの視点から見た解 (solution) を、その視点に必要な適切な知識・スキル及び経験をもった、1人または複数のメンバーから構成されるチームが、その視点に対してあらかじめ設定されたセル役割仕様書、記述形式、ツール、手順書、部品、資産、関連文書に適合するとともに、他のソフトウェア生産セルと知的に連携しながら、その視点解に相当する成果物を開発、または保守するように設定された、並行して実行可能なソフトウェア生産ライン上の区切りのことである。

3. ソフトウェアファクトリの新しい展開

3.1 顧客に対するサービスの将来形態

調達され、事業の中で運用されるソフトウェアプロダクトは、保守・改善を繰り返しながら、導入期、成長期、安定期、高齢期を経て、最後に廃棄される。この運用ライフサイクルの中で、事業者は利益を得て、調達、保守、改善に投資したコストを回収する。

多数の事業所の連携によって本業を展開する事業主の多くは、運用中のソフトウェアプロダクトの保守・改善

に頭を悩ませている。このような課題に
 応えるための手段の1つとして、ソフトウェアファクトリは、図1に示すようなモデルを実現し、開発から運用後の保守・改善・廃棄に至るまでの一貫したサービス提供を志向する。

このモデルでは、ソフトウェアファクトリ (図の右) 及びソフトウェアプロダクトをこのソフトウェアファクトリから調達し、運用する事業主 (図の左) は、共通のプロダクト統合基盤の上で相互に連携する。図の左上にある監視部署では、常に組織の戦略構想、ビジネス・ニーズ、

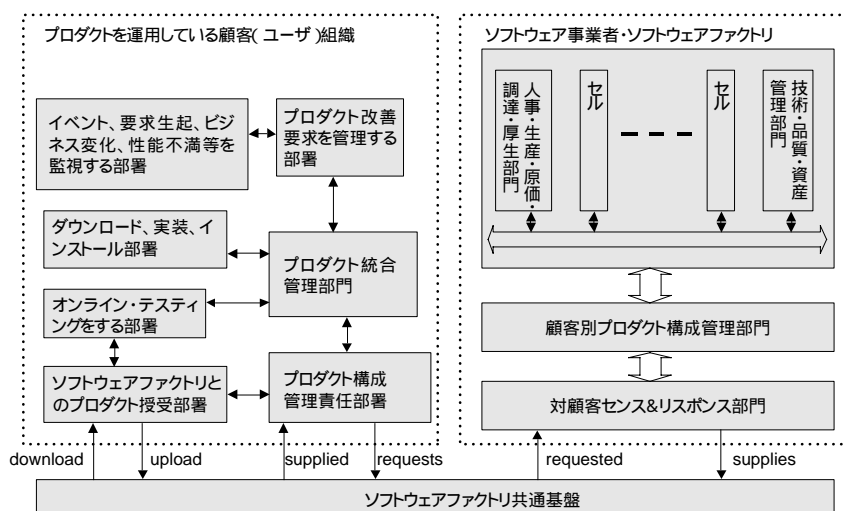


図1 次世代ソフトウェアファクトリ基盤モデル

4 東芝は、1977年に府中事業所内に、電力、鉄鋼、公共、交通システム向けの実時間制御システムを対象を絞ったソフトウェア工場 (以下、TSF : Toshiba Software Factory と略称する) を設立し、ソフトウェア再利用を徹底して実践した [MATSUMOTO81, 87, 92A, 92B, 93]。

5 Capability Maturity Model Integration : <http://www.sei.cmu.edu/cmmi/>

6 ハードウェア生産セルとは：1人から数人が、部品取り付け・組み立て加工・検査等、1つの製品を完成するまでの全工程、または比較的大きな部分工程に必要な作業を遂行する生産ライン上の区切りである。コンベヤ生産方式に代わる、少人数完結型生産方式である。

ビジネス・オペレーション、使い心地、技術レベル等とプロダクト特性との間の整合性を監視、予測、評価し、必要があればプロダクト要求管理部署に報告する。統合管理責任者は、これらの情報を受領し、評価し、意思決定して、プロダクト構成管理責任者を通してソフトウェアファクトリに要求を伝達する。図の右にあるソフトウェアファクトリ・顧客別構成管理責任者は、伝達された要求に応じてファクトリ内に必要な指示を与え、プロダクトの必要部分を改訂し、改訂部分をオンラインで、要求した事業主のプロダクト授受部署へ送達する。統合管理責任者は、評価、テスト及び実装に必要な手順を進め、運用中のプロダクトの更新を実施する。

3.2 MBSF及びCBSFに共通した発想

ソフトウェアファクトリは、顧客⁷からの受注に基づいてソフトウェア製品を開発し、納入し、保守・改善を行うことを事業とする。古典的ソフトウェアファクトリに比べて、MBSF及びCBSFでは、この項で述べるような共通した発想の転換が見られる。

3.2.1 基本思想

a. 高いコスト効果とすばやい市場参入

従来の開発・保守手法に比べ、より高いコスト効果性 (cost-effectiveness) 及びすばやい市場参入 (short-time-to-market) を狙う。

b. 無益な一般性の排除

取り組む問題 (problem) の範囲を絞り込み、問題に対して最適化された資源及び手法を用いて開発を行う。例えば、一流料理店は、懐石料理、フランス料理等、ジャンルを限定することで評価と利益を上げている。

c. 単発開発 (one-off development) の回避

新しく開発したソフトウェアは資産化して、次の受注開発で再利用する。カスタマイズ可能な資産の開発に正当な投資を惜しまず、資産の長期にわたる継続使用 (運用・改善・保守) によって得た利益によって投資を回収する政策をとる。あらかじめ適切に定義され、管理されている、フィーチャ (features) (3.2.2項で述べる) という概念の上で、販売戦略のために特定された共通のフィーチャをもつ実世界問題 (またはミッション) を解決す

るために、再利用可能資産 (固定資産と可変資産に分かれる) だけを用いて構築されたソフトウェアシステムをプロダクトラインと呼ぶ。同一のフィーチャをもちながら要求が異なる新たなソフトウェアシステムは、可変資産をカスタマイズすることによって開発・保守できるように計画する。

d. モデルの賢い利用

一般に、モデル指向開発にはまだ実用上多くの問題が残されている。MBSF及びCBSFでは、問題の領域を絞り込むことによって、モデルによって満たすことができる論理システムの範囲を限定し、フレームワーク、アーキテクチャ・パターン、デザインパターン等を活用して、モデルからコードへの変換を工業的に行えるようにする。モデルによって満たし得るとは、モデルで示された解釈に従って評価した論理システムが、真 (true) になり得ることを指す。モデルと、それによって満たし得る論理システムとの間の仲立ちをするものが、フレームワーク、パターンである。問題領域を絞り込むソフトウェアファクトリでは、絞り込まれた問題領域からマッピングされる解の意味を記述するためのモデル、すなわち「絞り込まれたモデル」を記述するDSL (domain-specific language) を提供し、ライブラリに準備された、その解領域に適したフレームワーク、パターンを利用して、コードを工業的に生成しようとする。

e. 分散した複数ソフトウェアファクトリの統合

顧客システムが、分散配置された異目的事業体から構成されているとき (例えば、SCM⁸) 扱う問題領域が異なる複数のソフトウェアファクトリを統合する必要がある。

3.2.2 フィーチャ指向アプローチ

「問題 (problem)」は、実世界に存在し、人の認知の中にある概念である。これを抽象化して表現したものが問題記述 (problem statement) と呼ばれる。要求 (requirements) は、記述された問題に対する理想解 (idealized solution) を「抽象的に」表現したもので、問題そのものではなく、問題からマッピング (mapping) された抽象解の表現である。フィーチャとは、ランダムに集められたシステムがあるとき、ユーザが理解し、表現できる (user-visible) システムの特徴 (characteristics) 、

7 例えば、組込みシステムの一部では、調達者、プロダクト・ユーザ (運用者) は同一ではない。

8 SCM : supply chain management

または側面 (aspect) を、And/Or ノードから作成される樹構造図で表し、特徴の共通性と可変性を識別し、共通性によって、システムを分類できるように仕組まれた特性のことである[CLEMENTS02]。

フィーチャ指向アプローチでは、まず同じフィーチャ集合 (通常、フィーチャ・モデル表記法[CZARNECKI04]によって記述される) によって可視化された問題領域を定義し、次にこの問題領域からマッピングされる要求領域を満たし得るようなプロダクトラインを開発する。この段階を、ソフトウェアファクトリのイノベーション段階 (innovation stage) と呼ぶ。イノベーション段階における主な作業項目は、次に示す5ステップに分かれる。

対象とする問題領域に共通するフィーチャ集合の定義、

フィーチャ集合を実現するソフトウェアシステム・ファミリに共通する論理アーキテクチャ及び物理アーキテクチャの設計 (これはプロダクトライン・アーキテクチャと呼ばれる)、ソフトウェア部品の設計、アーキテクチャ及び部品を再利用可能な固定資産及び可変資産に変換、対象とする問題領域名を付して、固定資産、可変資産、資産間の依存性定義、必要なツール、開発プロセス・ガイドラインを、適切なスキーマに従ってデータベースに記憶する。

この作業の中で作られるアーキテクチャ及び各資産は、対象とする問題領域に属する様々なアプリケーションの偏差に従ってゆれ動くようなものであってはならない。

顧客からの新しい受注があると、顧客が直面する問題を分析し、そのフィーチャを識別する。次に、識別されたフィーチャに合致するソフトウェア資産を、先に開発されたプロダクトラインから選定し、与えられた新しい問題において生起する要求差分によって、選定された可変資産をカスタマイズする。この段階を、ソフトウェアファクトリのコモディティ段階 (commodity stage) と呼ぶ。以上、3.2で述べたことを要約して、表1に示す。

3.3 MBSFとCBSFにおける互いに異なる特徴

MBSF及びCBSFは、それぞれ異なる特徴をもっているが、互いに補完関係にある。MBSFは、問題領域に特化したツール (DSL等) を強化することによって、多様な顧客要求を満たしながらも、なおかつ完全工業的に開発・保守を行うことを目標とする。他方、CBSFは、MBSFと同じ手法を採用しながらも、現状のツール能力で対応で

表1 MBSFとCBSFに共通する特徴

	MBSF	CBSF
対象とする問題領域	あらかじめ、なんらかの特定領域を選定して、定義する。	
特定された問題領域の特徴定義	フィーチャ集合によって、問題領域を定義する。	
ソフトウェア資産群の登録、管理	特定された問題領域のフィーチャ集合をパラメータとして、それを解決するために必要なソフトウェア資産群を定義し、固定部と可変部に分離し、プロダクトラインとして登録、管理する。	
ソフトウェアファクトリ・ライフサイクル	特定された問題領域のフィーチャ集合をパラメータとしてプロダクトラインを開発するイノベーション段階と、顧客の個別要求に従って可変資産をカスタマイズしてソフトウェア製品を開発するコモディティ段階に分ける。	

表2 MBSFとCBSFの特徴で、互いに異なる主な項目

	MBSF	CBSF
ファクトリを構成する単位 (ファクトリ単位)	ビューポイント (モデル指向)	セル (人智、モデル及びコード指向)
ファクトリ単位を構成する要素	成果物、ツール、ソフトウェア資産、ガイドライン	1人、または少人数チーム、セル・インタフェース、セル・コントラクト、入力成果物、出力成果物、ソフトウェア資産、ガイドライン
プロジェクトの形態	プロダクトラインをベースとし、ビューポイント及びビューポイント・マッピングによって形成されるファクトリスキーマの反復実行	プロダクト / プロセスラインをベースとし、セル及びセル間コミュニケーションによって形成されるセル・ファクトリスキーマの並行実行

きない顧客要求の変化に対する十分な対応を、人またはチームで実現し、従来の手法よりは、より工業的に顧客の要求を満たすことを目標とする。後者は、情報家電機器を生産するハードウェア工場で採用されているセル生産方式に倣ったシステムであり、人及びチームの知的能力、コミュニケーション能力を最大限生かすことによって、よりコスト効果性、対要求即応性を高めるとともに、開発・保守員の参加意識、責任感、スキルの向上も合わせて図ろうとする。主な違いを要約して、表2に示す。

3.3.1 MBSF

a. ソフトウェア開発の進め方⁹

富士山は、観察する位置によって様々に姿を変える。観察する位置は、視点とも呼ばれる。与えられた問題が複雑である場合、その解のある側面を、それぞれの視点から求めて、それを表現する。各視点では、当該プロダクト側面を記述するために用いる意味、パターン、記述すべき範囲、記述の目的、対象とする読者、成果物 (artifact) を生成するために必要な記述規約、言語及び記述法が定義される。これら定義のことをMBSFでは、ビ

9 イノベーション段階及びコモディティ段階に共通する事項である。

ューポイント (viewpoint) ¹⁰ と呼び[GREENFIELD04]、ビューは、ビューポイントに適合するインスタンスである (1つのビューポイントに対して複数のビューが存在し得る)。

図2には、ビューポイント (長方形)、ビューポイント・マッピング (矢) 及びビュー (長方形の中に記載された項目) の例を示してある。MBSFでは、ビューの中に、このビューポイントで使われるツール (例えば、DSL) やガイドラインも含める。文書化される半完成、または完成ビューを成果物と呼んでいる。

ビューポイントは、ウォータフォール・ライフサイクルモデルにおけるフェーズ、またはステップと混同される恐れがあるが、そうではない。ビューポイント・マッピングは、ビューポイント間の依存関係 (dependency) を表し、有向矢、または双方向矢で表される。

すべてのビューポイント、ビュー、ビューポイント・マッピングを網羅したグラフを、ソフトウェアファクトリ・スキーマ (software factory schema) と呼んでいる。

ソフトウェア開発・保守プロセスは、ソフトウェアファクトリ・スキーマを、最終プロダクトが完成するまで、MBSFでは反復実行しながら、他方CBSFではビューポイントごとに並行して同時実行しながら進められる。

b. プロダクトライン

MBSFでは、まず、イノベーション段階でプロダクトラインを開発する。ここで開発するプロダクトラインには、特定されたフィーチャ集合で特徴づけられた問題領域の名称、特性及びこの問題を解決するために実施するソフトウェアプロダクト・ファミリー開発に必要な固定資産、可変資産、資産間の依存性定義、必要なツール、開発プロセス・ガイドラインが格納される。資産には、ソフトウェアプロダクト・ファミリーに共通するソフトウェアアーキテクチャ、ソフトウェアコンポーネントが含まれる。

プロダクトラインは、特定された問題領域ごと、すなわち特定されるフィーチャ集合ごとに開発される。ソフトウェアファクトリは、問題ごとに設定されたプロダクトラインと一意に対応する。フィーチャ集合は、ソフトウェアファクトリを決定するパラメータとなる。

受注活動では、すでに開発されたプロダクトラインを決定づけている問題領域、すなわちフィーチャ集合に合致するような受注を狙う。受注する問題のフィーチャが完全にプロダクトラインのフィーチャと合致することはあり得ない。したがって、フィーチャの差分は、コモディティ段階に入ってから、個々のソフトウェア要求に対応し、可変資産のカスタマイゼーションによってこの差分に

適応する。以上で述べた2つの段階の関係をまとめて、図3に示す。

ソフトウェアファクトリは、しばしばレストランの厨房業務に例えられる。プロダクトラインを挟んで図3の左にある流れはイノベーション段階、右にある流れはコモディティ段階を表す。イノベーション段階はシェフモード、コモディティ段階はコックモードとも呼ばれる。シェフは、過去に蓄積された料理、調理経験から、新しいメニューを開発し、その調理プロセス、

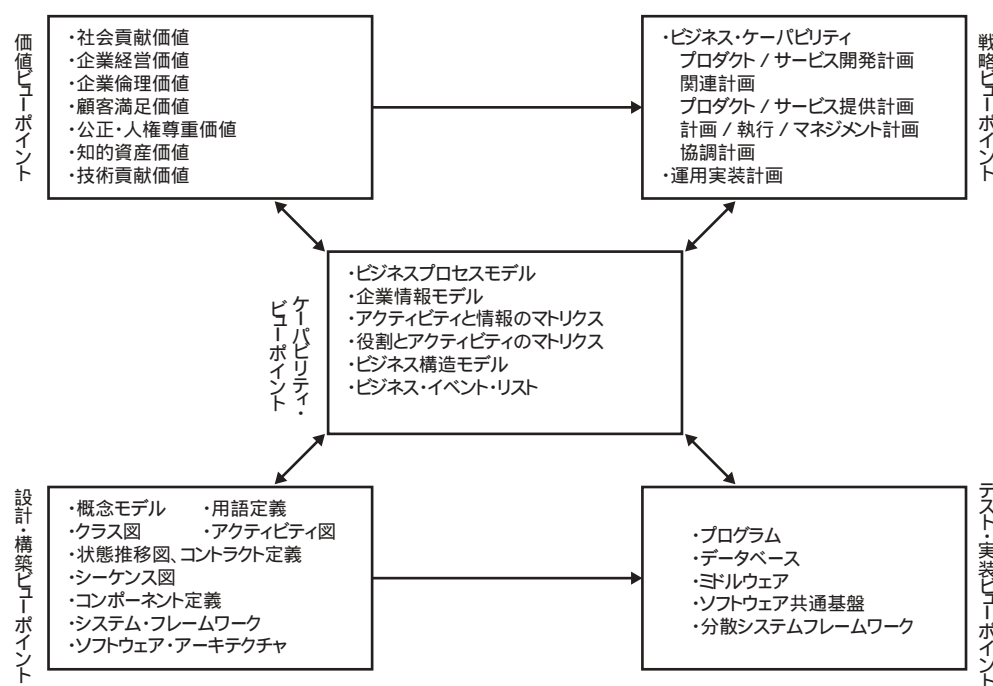


図2 ビューポイント、ビュー及びビューポイント・マッピングの例

10 TSFでは、ソフトウェア生産セルに対する作業指示仕様書、すなわちビューポイントをS-Mol、またはunit-workload order sheet (UWOS) と呼んでいた。

すなわちレシピを文書化してコックに与える。

客が訪れ、メニューに従って料理を注文する。コックモードがこれに適応する。高級レストランでは、なじみの客はしばしばメニューからの差分を示唆するため、コックはこの差分要求を適正に読み取って、レシピを修正し、調理を行う必要がある。

イノベーション段階では、ソフトウェア事業経営責任者は、事業が対象とする問題領域を、責任をもって特定する。シェフ、すなわちプロダクトライン開発者は、与えられた問題の特徴付けるフィーチャ集合を定義し、過去に出荷したソフトウェアプロダクトを記録、管理したデータベース（これをプロダクトグリッドと呼ぶ）から、このフィーチャ集合に適合する遺産を選出し、これを用いて図3の左端に記述された手順でプロダクトラインを開発し、データベースに実装する。

受注後、顧客の要求が定義されると、コック長は、これに適合するプロダクトラインから、ソフトウェアファクトリ・スキーマ、可変資産、固定資産、ソフトウェア部品、開発ツール、開発資源、プロセス記述を選定し、コックチームは可変資産をカスタマイズして、顧客の要求を満たすようなソフトウェアプロダクトを開発する。シェフは、コックチームを監督・指導し、コックからのフィードバックを得て、プロダクトラインに対して必要な改善を行う。

3.3.2 CBSF

CBSFの基礎となっている思想は、家電機器等ハードウェア工場で採用されているセル生産、システムバイオロジで研究が進められている生物体の細胞から構成されるシステムにおけるロバストネス評価[KITANO01]及び

Winogradらが進めている「language-action-perspective（以下、LAPと略称）による設計」である[WINOGRAD97]。

は、プロジェクト・ロバストネスを向上し、評価するための方法論として役に立ち、「設計に対する支援はコンピュータでなく、人と人との間で交わされるlanguage-actionに求めるべきである」とするの思想は、プロジェクト・マネジメントのための方法論の1つとしても、利用することができる。

a. ソフトウェア生産セル

ソフトウェア生産セルは、ANSI / EIA-632及びINCOSE Systems Engineering Handbook[INCOSE04]が提唱するシステムエンジニアリング・プロセス・フレームワークの中に組み込まれる。このフレームワークによれば、システムエンジニアリング・プロセスは、テクニカルマネジメント、テクニカル・エバリュエーション、調達及び供給、システム設計及びプロダクト実現という5つの基本プロセスに分かれ、互いに並行して連携を行うものとされる。

ソフトウェア生産セルは、図4で示すように、INCOSEでいう、調達及び供給、システム設計、プロダクト実現

に当てはめる。上下にあるテクニカルマネジメント及びテクニカル・エバリュエーションは、ソフトウェアファクトリを支援するソフトウェア事業組織の中で、セルとは別に組織されている必要がある。

図4の中の中核セルは、基本的構成要素、例えばシステムフレームワーク、ソフトウェアアーキテクチャ、タスクやスレッドが共有する共

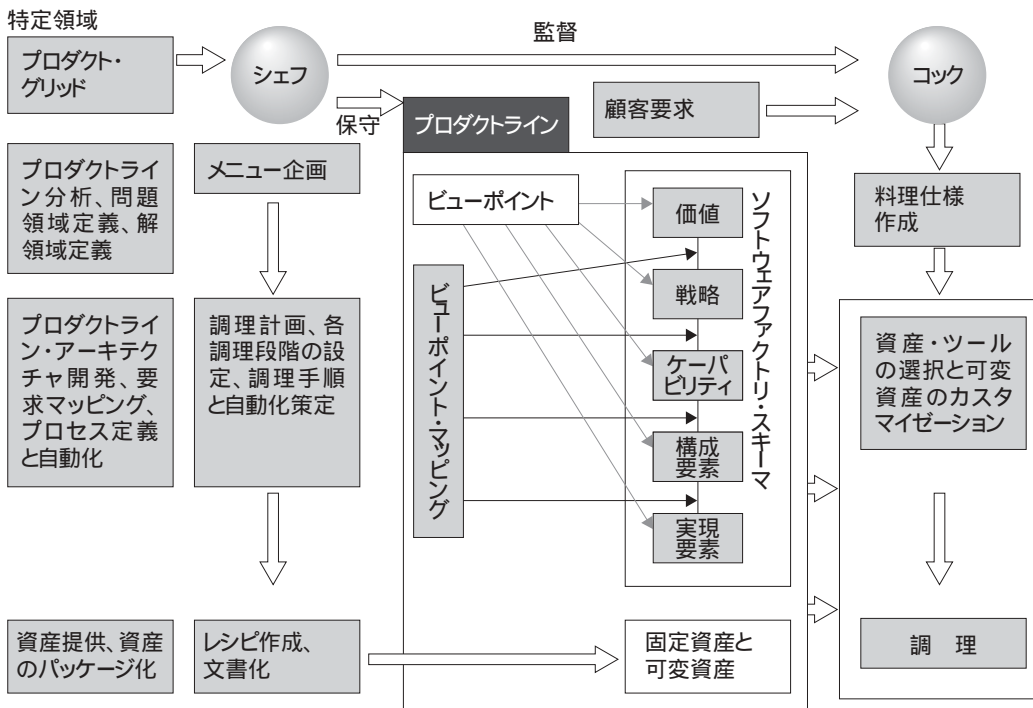


図3 イノベーション段階とコモディティ段階の関係

通資源（データ、アспект、ユティリティプログラム、アプリケーション・プログラミングインタフェース、プロトコル、チャンネル等）を設計し、その下に続く各種セル間のネゴシエーション、調整を行う等、中核的な役割を果たすPITセルであり、PITセルとPDTセルとの間のコミュニケーションは、LAPが提唱する design-by-language-action[WINOGRAD97]に沿って実施する。PIT及びPDTは、それぞれINCOSEで定義されている。

ソフトウェア生産セルの静的及び動的特性は、セル仕様¹⁾、静的インタフェース及び動的規約（コントラクト）

によって定義される。図5にこれらの関係を表すモデルを示す。

セル仕様：セルのミッション、セルが受ける制約（工程、コスト、リスク、品質、準拠標準等）、セル内作業（マイクロプロセスと呼ぶ）のための手順、入力成果物仕様、出力成果物仕様、利用する固定資産、可変資産、ツール及びソフトウェア部品を特定する。

静的インタフェース：入力成果物を受け入れるためのインタフェース、出力成果物を提供するためのインタフェース

コントラクト：

- ・トリガ条件：セル成果物仕様等をパラメータに持つ、プロジェクトリーダからの始動指示等を受け入れるための条件
- ・割り込み許可条件：パラメータで指定された外部のイベント生起によって、一時的にセルの実行を中止し、イベント処理終了後に再開することを許可するための条件
- ・トラップ条件：パラメータで指定された罠（trap）、すなわち外部イベント生起によって、セルの実行を放棄（abort）するための条件
- ・イベント通知条件：セル内部で生起したイベント（例えば、他のセルへのコミュニケーション・ニーズ）によって、パラメータで指定されたセルに、イベントの生起を通知するために利用する条件

ここで示したコントラクトは、セル相互間のコミュニケーション・シナリオをコントロールする、LAPに基づいた対話支援環境と結合される。

b. セル・スキーマ

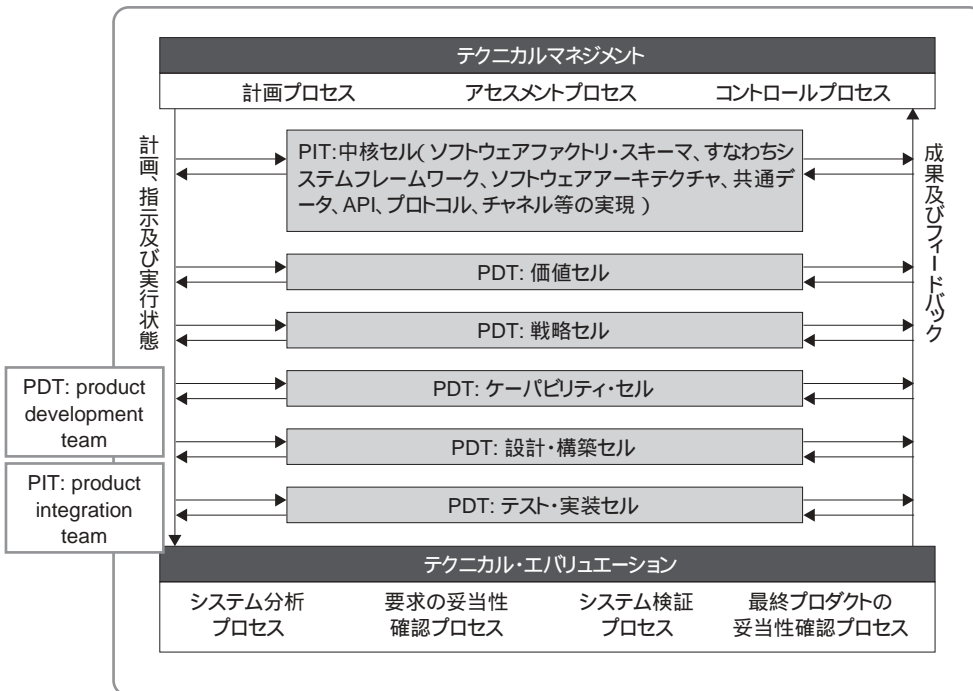


図4 ソフトウェア事業組織の中でソフトウェア生産セルの占める位置

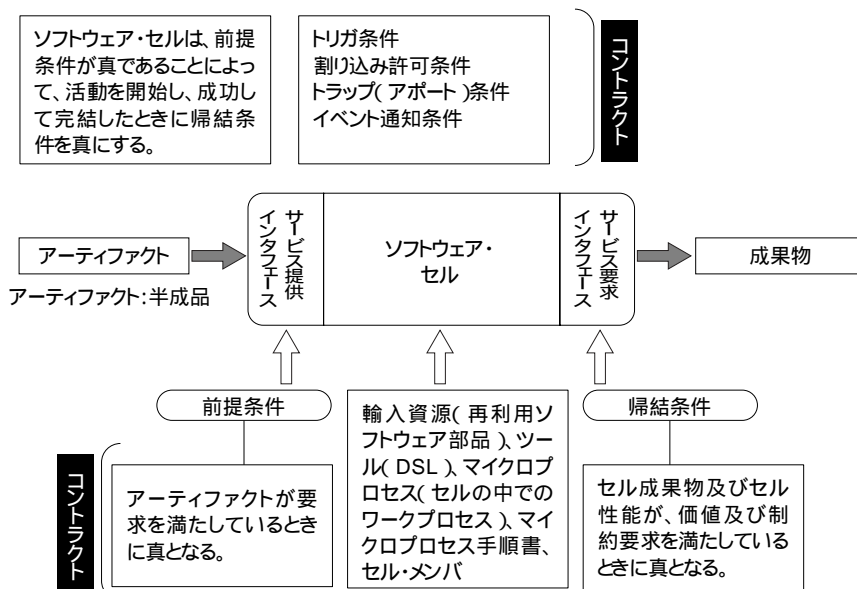


図5 ソフトウェア生産セルの特性を示すモデル

一般に、モジュール (module) とは、抽象化され、内部の詳細が隠蔽され、公開されたインタフェースをもつ、1つの目的をもった実体 (object) を指す [PARNAS83]。互いに連携するモジュール間には、依存 (dependency) が存在し、ソフトウェア・モジュール間の依存には、次のようなものがある。

一般依存性

- ・プロダクト依存性

設計依存性 (design dependency)

- ・設計過程における、プロダクト依存性とプロセス依存性を統合した「デザイン」と称する抽象実体間の依存 [BALDWIN00]

複数のモジュールから構成されるシステムにおいて、依存性ができるだけ小さく、かつ一方方向になるように構成されたシステムを、モジュラシステム (modular systems) と呼び、システムにおいてモジュール間の依存をできるだけ小さくすることをモジュラ化 (modularization) ¹² と呼ぶ。モジュラ化のためのツールとして、一般には DSM (design structure matrix) [EPPINGER91] 等が用いられるが、CBSF では、DSM の代わりに、CDM (cell dependency matrix) を用いる。CDM

の例を図6に示す。

CDMでは、DSMとは違って、行にはセルに対する入力、列には行からの出力成果物を配置する。対角線上にはセルの識別記号を記入し、×印によってセル間の依存関係が存在することを示す。DSMを使う場合と同じようにこの図を用いて、セル間の関係をモジュラ化する。モジュラ化されたセルの構成を、セル・スキーマと呼ぶ。3.2.2項で述べたフィーチャをキーにして分類し、記憶し、管理されているセル・スキーマを、プロセスラインと呼ぶ。CDMの右上の空間にある×印を完全に取り除くことは不可能であり、相互に依存関係のあるセル間の調整は、各PDTの代表者が参加するPITが行う。

c. LAP

これまで設計は、コンピュータへ向かって行う記号操作、コンピュータによる編集・検索支援及び表示によって行われるものと考えられてきた。人とコンピュータの対話は、ある狭いセマンティックによって満たされるプロトコルの連鎖によって行われるものされ、セマンティックをできるだけ限定することによって、より正確な意志伝達が行われるものされてきた。

自然言語は、人と人との間のコミュニケーションを行

行方向にはセル入力、列方向にはセル出力成果物	社会貢献価値 企業経営価値 企業倫理価値 顧客満足価値 公正・人権尊重価値 知的資産価値 技術貢献価値	ビジネス・ケーパリティ計画 運用実装計画	ビジネスプロセスモデル 企業情報モデル アクティビティと情報のマトリクス 役割とアクティビティのマトリクス ビジネス構造モデル ビジネス・イベント・リスト	概念モデル 用語定義 クラス図 アクティビティ図 状態遷移図、コントラクト定義 シーケンス図 コンポーネント定義 フレームワーク アーキテクチャ	プログラム、ミドルウェア、データベース、ソフトウェア共通基盤
企業定款、決算報告書、事業計画	価値セル		×		
組織構造及び機能、経営資源、市場分析	×	戦略セル	×		
伝票、事業報告		×	ケーパリティ・セル		×
ソフトウェア要求定義書			×	設計・構築セル	
				×	テスト・実装セル

図6 CDMの図法を説明するための図

11 MBSFのビューポイントに相当する。

12 「モジュラ化」は、モジュール化することではなく、モジュール間の依存性を識別し、設計・実装の後戻りを少なくするために行う設計の合理化 (rationalization) を指す。

うために利用される最も強力な仕組みであり、なんらかの文法による制約を受けるにせよ、その国の文化で培われた広いセマンティックを伝達する力をもつ。これまであまり議論されてこなかったが、人は自然言語を交わすことによって、設計という行為を行うことができる。

PITが、専門領域（例えば、ビジネス専門とIT専門）の異なるPDTとの間で行う調整では、従来のコミュニケーションにlanguage-actionを加えることによって、さらにコミュニケーションの品質改善を試みることができる。このコミュニケーションは、すでに実施されている「見える化」（たとえば、[SEC06][NTTDATA06]）の実践を前提とするものである。CBSFでは、次に示すような規則に基づいた自然言語によるコミュニケーションを交えた、「わかる化」¹³を目指す。

発話規則（theory of speech acts）[SEARLE69]

- ・命題（propositional contents）を明示すること
- ・背景となる情報を正確に伝えること
- ・発話者の意図を完全、かつ十分に伝えること

対話規則（theory of communicative action）

[HABERMAS84]

- ・対話で創造する目標（goal）を明示して始めること
- ・対話の結果に対する価値評価基準を明示して始めること
- ・対話が辿る模範的道筋（norm）を明示して始めること
- ・対話に対して責任をとる職位、責任者を決めて始めること
- ・対話に必要な資源の妥当性が確認されていること
- ・対話の過程を記録し、評価すること
- ・対話の結果を、あらかじめ決めた価値評価基準に基づいて評価し、取舍選択について意思決定する手続きを定めること

具体的な試みとしては、次のような項目がある。

- セル間折衝のための対話パスを構造化し、各レベル及びレベル間で交わされる対話シナリオを明示、対話履歴の記憶・管理して、検索できるようにする。
- 文書、または図式の難解な部分にアンカーを設けて、それを通して自然言語による音声解説を聞き取り、または関係者との直接対話、あるいは複数関係者との遠隔会議に入ることができるようにする。

4. ソフトウェアファクトリの経済モデル

ソフトウェア事業組織が、ソフトウェアファクトリを事業として採用することの意思決定は、当該組織トップマネジメントの責任で行わねばならない。ソフトウェア事業組織は、顧客の問題領域に関して、十分なソフトウェア製品開発・納入・運用・保守の実績を保有することが必要である。この意思決定を行うためには、対象とする製品系列、または部分製品系列ソフトウェアファクトリのイノベーション段階への投資額（I）が、コモディティ段階を通して得られるキャッシュフローの現在価値（PV）を越えてはならない。次に試算例を示す。

試算例

あるソフトウェア製品系列に関して、資産寿命5年のソフトウェアファクトリを利用して、毎年平均10システムを受注して、開発・出荷・据付け・運用・保守し、ソフトウェアファクトリのイノベーション及び保守・改善を勘案し、毎年のキャッシュフロー値が3,000万円になるとする。

資本コストが、各種リスクを勘案して5%と見積もる。これを現在価値（PV）に換算すると、12,988万円になる。

したがって、ソフトウェアファクトリ開発投資額は、12,988万円以下であるとき（またはIRR、すなわち内部収益率が資本コスト（利子率+リスクプレミアム）を上回るので）、当該ソフトウェア製品系列のためのソフトウェアファクトリ開発を決定してよい。

5. 議論

半導体や通信の分野では、イノベーションの結果をコモディティ化し、コモディティを国際標準化することによって市場占有率を高めようとする。ソフトウェアの分野でも、SCM、CRM¹⁴、自動車搭載用電子システム等、いくつかの問題領域では、ポートフォリオ（運用資産）を開発し、コモディティ化し、それを武器にして市場占有率を高めることに成功している企業がいくつか存在することは周知のとおりである。本稿で述べたソフトウェ

13 各生産セルは、それぞれの意味領域に特化することを許されているため、セル間で交換される「見える化」された図式に誤解が生じることがある。

14 CRM：customer relationship management

アファクトリは、このような戦略を持つソフトウェア企業からのニーズに応じて構想されている。

日本のソフトウェア事業者の一部には、顧客の要求に沿う単発開発に終始し、利益確保の手段をコスト低減が可能なオフショアアウトソーシングに求め、イノベーション段階までもオフショアアウトソーシングしながら、貴重なノウハウを流出させるケースもある。これらの事業者にとっては、ソフトウェアファクトリ構想は、夢と映るかも知れない。しかし、日本にも、いくつかの問題領域において強い国際競争力を持ったソフトウェア企業が存在し、日本が世界をリードするソフトウェアファクトリへの回帰という夢をつないでくれている。

米国では、オフショアアウトソーシングに対する議論がきわめて活発に行われ、主なプロフェッショナルソサイエティは、Web上でこれに関する公式見解を公表している¹⁵。日本では、学会¹⁶等からの公式見解はみられないが、オフショアアウトソーシングに活路を求めている日本の事業者は、アウトソーシングによって、将来、自分たちが苦しむ事態にならないよう、十分な政策と戦略を立てて実施すべきである。

オフショアアウトソーシングのための手段の1つとして、本稿で述べたソフトウェアファクトリを海外で確立する方法も検討されている。例えば、イノベーション段階は日本の主管事業者が責任を持って遂行し、コモディティ段階だけをオフショアし、コモディティ段階で得たキャッシュフローを使ってイノベーションへの投資を回収できるようにする、という方法である。

6. むすび

前節で述べた議論にみられるように、わが国のソフトウェア業界の大勢は、まだ本稿で述べたような新しいソフトウェアファクトリ構想を実現できるような環境には達していないとされる。しかし、将来を見据えて行動する能力を備えた、賢明なソフトウェア事業者の中には、このようなソフトウェアファクトリ構想へ向けた研究姿勢もみられる。

一方、日本の伝統的なソフトウェア工場は、それぞれニーズの変化に合うように形態を変えながら進化を続けてきた。例えば、TSFの場合は、納入済みの全ソフトウ

ェア製品の保守・改善及び構成管理を効率的に行えるような体制に改めている。これらの遺産は今後も引き継いでいかねばならない。

ソフトウェア事業が対象とすべき問題領域は、エンタプライズ系及び組込み系だけではない。INCOSEがカバーしようとしている広い領域（宇宙、地球環境、防衛、福祉、教育、行政等）を視野に入れ、市場を開拓することによって、さらに大きな利益創出を目指すようにしなければならない。広い視野から見た、ソフトウェアファクトリの計画を望みたい。

参考文献

- [AAEN97] Aaen, I. et al., The Software Factory : Contributions and Illusions, in Proceedings of the Twentieth Information Systems Research Seminar in Scandinavia, Oslo, 1997
- [BALDWIN00] Baldwin, C.Y. and K.B. Clark : Design Rules, The MIT Press, 2000
- [BASIL89] Basili, V. R., The Experience Factory : Packaging Software Experience, Proceedings of the 14th Annual Software Engineering Workshop, NASA Goddard Space Flight Center, Greenbelt MD., 1989
- [CLEMENTS02] Clements, P., et al. : Software Product Lines, Addison-Wesley, 2002
- [CUSUMANO91] Cusumano, M.A. : Japan's Software Factories, Oxford University Press, 1991
- [CZARNECKI04] Czarnecki, K., et al. : Staged Configuration Using Feature Models, Proc. Software Product Line Conf., 2004
- [EPPINGER91] Eppinger, S.D. : Model-based Approaches to Managing Concurrent Engineering, Journal of Engineering Design 2 (4) , 1991
- [GREENFIELD04] Greenfield, J. et al : Software Factories, Wiley Publishing, 2004
- [HABERMAS84] Habermas, J. : The Theory of Communicative Action : Reason and Rationalization of Society, Vol.1, Boston Press, 1984
- [INCOSE04] INCOSE/Systems Engineering Handbook, International Council on Systems Engineering, INCOSE-TP-2003-016-02, Version 2a, 1 June 2004
- [KITANO01] 北野宏明編 : システムバイオロジーの展開, シュプリンガー・フェラーク東京, 2001
- [MATSUMOTO81] Matsumoto, Y., SWB System : A Software Factory, in H. Hunke (Ed.) : Software-Engineering Environments, North-Holland, Amsterdam, 1981
- [MATSUMOTO87] Matsumoto, Y., A software factory : An overall approach to software production, in "Software Reusability" ed. by P. Freeman, pp.155- 178, IEEE Computer Society, March 1987
- [MATSUMOTO92A] Matsumoto, Y. : Toshiba Software Factory, in "Modern Software Engineering", P.A. Ng and R.T. Yeh (eds.) , pp.479-501, Van Nostrand Reinhold, 1990
- [MATSUMOTO92B] Matsumoto, Y. : Japanese Software Factory, Advances in Software Science and Technology, Vol.4, Japan Society for Software Science and Technology, pp.21-42, Iwanami Shoten, Tokyo, 1992
- [MATSUMOTO94] Matsumoto, Y. : Japanese Software Factory, in "The Encyclopedia of Software Engineering", J.J.Marciniak (ed.) , 1st Edition, pp.593-605, John Wiley & Sons, New York, 1994
- [MATSUMOTO06] <http://www.5d.biglobe.ne.jp/~y-h-m/CBSF.pdf>, April 2006
- [MCLROY69] McIlroy, M. D. : Mass-Produced Software Components, in "Software Engineering Reports" on a Conference Sponsored by NATO Science Committee, Brussels, 1969
- [NTTDATA06] 見える分かるシステム開発 - プロセス透明化のススメ, 株式会社NTTデータ, 2006
- [PARNAS72] Parnas, D.L. : A Technique for Software Module Specification with Examples, Comm. ACM, 15 (5) , pp.330-336, 1972
- [SEC06] 長岡良蔵 : プロジェクト見える化とは, SEC journal, Vol.6, pp.28-29, 2006
- [SEARLE69] Searle, J. R. : Speech Acts - An Essay in the Philosophy of Language, Cambridge University Press, 1969
- [WINOGRAD97] Winograd, T. : The Design of Interaction, P. Denning and B. Metcalfe, Eds., Beyond Calculation, Springer-Verlag, 1997

15 <http://www.ieeeusa.org/policy/positions/offshoring.html> (IEEE/USAのオフショア・アウトソーシングに対する公式見解)

16 日本の学会は、プロフェッショナルソサイエティに位置づけられている。

東北地域の産業競争力強化に向けた取り組み 東北地域産業クラスター形成戦略「TOHOKUものづくりコリドー」 ～モノ作りとITの融合による地域の競争力強化～

<http://san-cluster.icr-eq.co.jp/>

東北経済産業局 地域経済部 情報・製造産業課 情報産業係長

中山 陽輔

激化する国際競争の中で、我が国全土に所在する産業が生き残っていくためには、全国各地で地域の産業資源を活用したイノベーションが生まれ、新事業・新産業が創出されるとともに、さらにイノベーションの連鎖が全国に波及していくことが重要である。

係る観点から経済産業省では、産学官の連携強化等により世界に通用する地域産業・企業を創出するため、平成13年度から産業クラスター計画を実施しており、平成18年度から第一期（18年度～22年度まで）に移行することとなった。

これを受けて東北経済産業局では、地域の産学官の有識者からなる「東北地域産業クラスター形成戦略懇談会」を設置して1年間にわたって徹底的に議論し、新たな産業クラスター計画「TOHOKUものづくりコリドー」をとりまとめた。

本稿では、当該計画の概要と併せて、読者の関心が高いであろうと思われるIT分野の概要について紹介する。

1 なぜTOHOKUでものづくりか？

産業クラスターの形成は、多くの産業分野で形成可能であるが、我が国が将来にわたって競争力を維持、強化し、今後ますます激化する国際競争の中で生き残っていくためには、高度な研究開発力を発揮し、次世代に通じるより高度な製品、技術を次々と生み出していける産業分野でクラスター形成を急ぐ必要がある。

こうした観点で東北地域に目を転じると、クラスターを形成し、競争と協調を図りつつ世界と伍していける製品、技術を次々と輩出する産業は製造業ではないかと考えられた。すなわち、東北地域は、伝統的には農林水産業を中心に発展してきた地域ではあるが、これらの産業は、国際競争力という観点では極めて脆弱であり、また、サービス産業も概ね同様である。一方、東北地域は、機械金属、電気機械といった製造業の集積に厚く、中には幾多の不況や困難を乗り越え世界に通じる技術や製品を作り出し、我が国はもとより世界の市場を相手にしている企業が数多く見られる。こうしたモノ作り企業の製品や技術を生み出す力を中心にクラスターを形成することができれば、東北地域の活性化につながり、ひいては我が国の競争力の強化に資すると考えられた。

このことを基本に、「東北地域産業クラスター形成戦略懇談会」では議論を重ねた。

2 検討の経緯・結果

2.1 基本的な考え方

クラスター政策は、地方圏の雇用を拡大することで国土の均衡ある発展を標榜した工業再配置政策等従来の地域産業政策と異なり、国が特定の地域を選定・指定し、多額の財政的支援措置を講じるものではない。クラスター形成の主役である地域や企業の取り組みを支援し、伸ばしていくことがその要諦である。よって、産学官関係者がクラスター形成のための議論を通じ、ビジョンを共有し、実現のためのベクトル合わせを目指した。

2.2 具体的な検討方法

諸外国を含む他地域の事例を分析した結果、産学官のネットワークの存在、地域を牽引するリーダの存在等クラスターを形成する要素が多い地域に、市場の大幅な拡大が見込まれる等ポテンシャルの高い技術が集積し、クラスターが形成されていることが明らかになった。東北地域においても、産業集積地のクラスター形成要素のポテンシャルを明らかにするとともに東北地域で取り組まれている技術・研究開発プロジェクトを分析することで、クラスター形成の核を探求した。

(1) 地域のポテンシャル

「八戸地域」、「北上川流域地域」、「広域仙台地域」、「山形・米沢地域」等の10の産業集積地域内におけるネットワーク活動、イノベーションの担い手であるベンチャー企業の創出状況等クラスタ構成要素を分析した結果、5～10年以内にクラスタを形成するポテンシャルが高い地域として、次の4地域が上がった。

北上川流域地域	広域仙台地域
山形・米沢地域	広域郡山地域

(2) 技術分野のポテンシャル

大見忠弘東北大学名誉教授による超LSIの多品種少量生産方式の研究開発プロジェクト、城戸淳二山形大学教授及び山形県による有機エレクトロニクスバレー構想をはじめとする16の技術・研究開発プロジェクトについて分析を行い、今後の市場ニーズ等の予測も踏まえ、5～10年の短期間でクラスタ形成の核となる可能性を有する技術分野について図1の5分野を同定した。

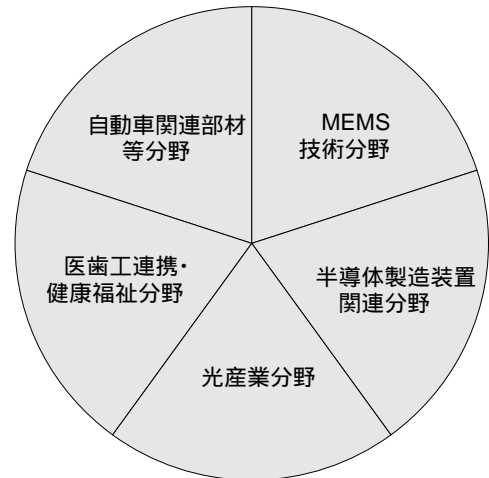


図1 クラスタ形成の核となる可能性を有する技術分野

なお、「非鉄金属リサイクル分野」及び「IT分野」の2産業分野については、東北における当該産業の競争力や産業集積の状況を考慮すると、短期間にクラスタを形成する可能性は少ないものの、製造業のクラスタ形成を優位に推進するために不可欠の産業であり、モノ作りを

東北地域紹介

東北地域は、青森県、秋田県、岩手県、山形県、宮城県、福島県の6県で構成され、地域の総面積約62,928平方キロメートル(全国対比:16.7%)、総人口約971万人(全国対比:7.6%)となっている。工業団地、高速道路等の産業インフラの整備に伴い、昭和50年代に電子系大企業の生産拠点工場が多数進出し、平成5年には完成車組み立て工場も立地。この結果、平成16年度の製造製品出荷額は約17兆円、全国の6%を占め、モノ作り産業の集積地として発展している。なかでも、組み込みソフトと密接な関連を有する電機機械工業は、事業所数で全国の10%、製造品出荷額で11%を占めており、東北の製造品出荷額全体の34%を占めるリーディング産業となっている。一方、情報サービス産業の売上高は全国の1.5%にとどまっているほか、売り上げも前年から9%下落している等低調な状況にある。

東北地域の情報サービス産業の実態

全国売上高 14兆5,271億円(前年比2.5%増)
 ・東京都売上高 8兆8,581億円 全国比61%
 (前年比8.7%増)
 ・東北6県売上高 2,100億円 全国比1.45%
 (前年比8.9%減)

(平成16年度特定サービス産業実態調査より)

青森県 総面積:8,918平方キロメートル
 総人口:約145万人 域内総生産:42,515億円
 秋田県 総面積:11,434平方キロメートル
 総人口:約116万人 域内総生産:37,227億円

岩手県	総面積:約15,279平方キロメートル 総人口:約140万人 域内総生産:45,638億円
山形県	総面積:約6,652平方キロメートル 総人口:約122万人 域内総生産:40,379億円
宮城県	総面積:約6,862平方キロメートル 総人口:約237万人 域内総生産:84,764億円
福島県	総面積:約13,783平方キロメートル 総人口:約211万人 域内総生産:76,593億円



資料出典: <http://www.tohoku.meti.go.jp/koho/panf2006/zenbun.pdf>

「支える産業分野」と位置づけ上記5分野に準じるものとして競争力強化に努めるべきとされた。

これらの検討結果を踏まえて、東北地域の産業クラスターは前述の5つ（7つ）の技術分野において4地域を中心に形成される可能性が高いと結論づけられ、クラスターの形成を早期に達成するために、5分野あるいは4地域を相互に緊密に連携させるためのコリドー（回廊）でつなぐ「TOHOKUものづくりコリドー」が東北地域のクラスター形成戦略とされた。

「TOHOKUものづくりコリドー」は、自らが発展しようとする企業・大学等の研究機関等「民」と「民」をつなぎ、相乗効果を増すため、「民」と支援する側の「官」をつなぐため、あるいは緊密な支援体制を構築するため「官」と「官」をつなぐための「コリドー」であり、官民協働による地域の競争力強化のためのビジョンである。

このビジョンの遂行により東北地域に産業クラスターを形成し、地域の競争力強化を実現する。

3 IT分野のクラスター形成の担い手

東北地域のIT企業の動向を俯瞰すると、全国のIT産業の売上高が順調に拡大しているのに対し、東北地域は平成12年の2,460億円をピークに、平成16年には2,100億円にまで減少している。これは、東北のIT企業は下請型の企業が多く、独自の分野で強みを発揮する企業が少ないことが原因と考えられる。

しかしながら、近年のオフショア開発の進展等に強い危機感を感じた企業が、下請型の業務形態を打破し、競争力のある企業へと変革することを目的とした企業間ネットワークの形成や、製造業とIT企業が連携して組込みソフトウェアに関する人材育成や技術・研究開発を行うプロジェクトも顕在化している。

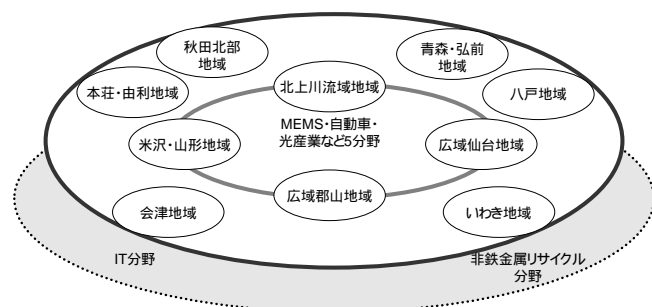


図2 TOHOKUものづくりコリドーのイメージ

今後、これらのプロジェクトの取り組みが加速化・深化することでクラスター形成の担い手となる競争力のある企業が続々と輩出されることが期待される。

(1) 東北ITクラスター・イニシアティブ (TIC)

宮城、山形、福島県の3県のIT企業が互いに連携を図り、下請型の業務形態から脱却するために、戦略的な技術開発、マーケティングを実施するために立ち上げたネットワーク組織。民主導で、企業間での連携による新製品開発、共同研究等を行っている。平成17年5月の設立以来、すでに会員企業間アライアンスを形成し新製品開発に取り組み、商品化を達成できた事例も存在する。

(2) とうほく組込み産業クラスター (ET tohoku)

山形、宮城、岩手の製造業、IT関連企業、大学、高専等の高等教育機関が連携し、東北の企業が、競争力のある製品を製造していくためには、開発技術力を有する組込み技術者の育成と保有が不可欠であるとの認識に基づき設立されたコンソーシアム。産学官が連携し、体系的な人材育成等を行うこととしている。理事長に赤塚孝雄山形県産業技術短期大学校長、副理事長に門田浩SEC組込み系プロジェクトリーダー、宮崎正俊東北大学名誉教授、鎌田信夫ソリトンシステムズ株式会社代表取締役を迎え、平成18年8月2日に設立された。

(3) 戦略的IT産業強化育成プロジェクト

岩手県、財団法人いわて産業振興センター、株式会社岩手ソフトウェアセンターが連携し、産学官によるネットワーク組織「組込みコンソーシアムいわて」の立ち上げ、「取引情報の開拓・斡旋」、「中小企業グループを対象とした技術者育成」等の事業を一体的に展開することとしている。また、岩手県立大学では「組込みソフトものづくり塾」の開催、「組込技術研究所」の設置等、組込み分野へ積極的に取り組んでいることから、今後同大学と連携した人材育成や技術開発を視野に入れている。

(4) 会津みらいプロジェクト

福島県会津若松市にはITベンチャー企業の集積が進み、約20社、約30億円の市場を実現している。これらITベンチャー企業群がさらなる成長を遂げ、2010年には約50社、約100億円、約500名の市場へ成長させるため、株

式会社津リエゾンオフィスや会津若松市、会津大学等が連携し、ITベンチャー20社の参加を得て、以下の事業を実施している。

- ・企業間連携の構築
- ・人材の育成
- ・成長原資の導入
- ・社会的認知の確立
- ・販路の拡大に資する各種事業

以上の他、会津地域（会津大学及びITベンチャー企業）と新潟地域（財団法人新潟産業創造機構、長岡技術大学）が組込みソフトウェア分野の技術・研究開発において「機能安全」をキーワードに連携を開始しつつある。

4 IT分野の目標・本年度の 当局の活動計画

今後5年の間に、東北地域のIT企業の競争力を強化し、モノ作り分野のクラスターのアクターである企業に良質なソフトウェア（組込みソフトウェアを含む）を供給し、クラスターから創出される製品・サービスの競争力強化を図る。また、製造業（とくに中小企業）にITを活用したソリューションを提供することにより、製造業の生産管理の合理化、製品の短納期化等のプロセス・イノベーションを図ることを通して製造業の競争力を強化し、モノ作り分野のクラスター形成の加速化に資するとともにIT産業のクラスターの基礎を形成する。

このような取り組みをとおして、今後10年程度の間に関内地域のIT企業と製造業の相互の競争力強化を実現する東北独自のIT - 製造業連携型のクラスターの構築を目指すことを目標としている。

18年度の当局の活動計画としては前述のようにクラスター形成の主体である「民」の自発的な取り組みを深化・加速化あるいは補完するための支援を行う。その概略は以下のとおりである。

(1) 連携促進・ネットワーク形成支援

広域的な新事業支援ネットワーク強化事業（ネットワーク補助金）を活用しクラスター・コアの活動を支援する。

- ・「とうほく組込み産業クラスター」の設立・活動支援
- ・「東北ITクラスター・イニシアティブ」の活動支援
- ・組込みソフト分野における会津地域と新潟地域の連携支援

- ・「組込みコンソーシアムいわて」の設立・活動支援
- ・「会津みらいプロジェクト」の活動支援

(2) 技術・研究開発プロジェクト発掘・支援

地域新生コンソーシアム研究開発事業、新連携対策事業、サポーターディングインダストリー支援策等を活用した技術・研究開発の発掘・支援

(3) 販路開拓・域外への情報発信

ET2006（11月15日～17日 会場：パシフィコ横浜）に「TOHOKUものづくりコリドー」としてパビリオン出展する（8月8日時点で14社がエントリを表明している）。

来年以降は、これらの拡充・強化に努めていく予定である。

5 今後の展開

当局をはじめとした「官」が主導でプロジェクトメイクを行うのではなく、経済・産業の担い手である企業等「民」の自発的な活動の深化・加速化を支援することでクラスターを形成するためのビジョンが「TOHOKUものづくりコリドー」であるならば、「政策」というツールにより支援する我々「官」が現場を直視する重要性は増すことはあれ、減ることはないとは筆者は認識している。

政策の投下後、構造変化の兆しは現場での現象に現れるのであり、必ずしもマクロの統計データからは読み取ることのできない産業の実態を的確に把握し、政策に反映させていくことが肝要である。

このような認識の下、当局職員は、積極的な企業訪問等を通して、産業の実態の的確な把握に努め、政策的対応が可能なものは速やかに政策に結びつけるとともに、マーケットと地域企業、地域の企業同士、あるいは「民」と「官」の間に必然的に存在する情報の非対称性を解消するための役割を果たす所存である。

「TOHOKUものづくりコリドー」は本年度開始されたばかりのプロジェクトであり、当局としても試行錯誤の段階ではあるが、今後、施策の拡充や他地域との連携にも取り組んでまいりたいと考えている。IPAをはじめ全国の皆様の御指導・御助言を賜れば幸いである。

参考情報

東北ITクラスター・イニシアティブ（TIC） <http://www.tic-ijp/>

会津みらいプロジェクト <http://aizu-mirai.com/>

BOOK REVIEW

ウェブ進化論

梅田望夫 著

ISBN : 4-480-06285-8 筑摩書房刊
新書判・256頁・定価777円(税込) 2006年2月刊

ほとばしり出る新時代の息吹

読まないわけにゆかず、読んだら人に薦めないわけにはゆかず、そして何よりも自らの行動に反映しなくては意味がない、そういう本である。著者の、新時代の到来についてできるだけ多くの人にわかって欲しい、という気持ちが伝わってくる内容である。

はやりの新書判ながら字数が多く、じっくりと説明されている。Web 2.0、ロングテール、アドワーズ、アドセンス、「あちら側」「こちら側」などトレンドイ用語のオンパレード。Googleの凄さについてもたっぴり紹介されている。

本書を読んだら是非著者の前著「シリコンバレーは私をどう変えたか」を読んでほしい。この2つの著述の間

に大きな景気循環の区切りをはっきりと見てとれる。前著では著者は熱狂のシリコンバレーの中でユーフォテリア

(根拠なき陶醉感)だった、つまりポーッと思考停止していたということを自信なげに述懐していた。それが今回は新しい明確なトレンドを見抜いて自信に溢れ、これできるだけ多くの人に伝えたいと燃えている。いわば大きな新しいサイクルの立ち上がり、氏の豊かな感受性と見識に乗り移ってほとばしり出てきた、そういう感触の好著である。筆者は本書読後にブログを始めてみた。

(神谷芳樹)



ソフトウェア開発オフショアリング完全ガイド

S-open オフショア開発研究会 著

ISBN : 4-8222-2970-X 日経BP社刊
A5判・250頁・価格: 3,990円(税込) 2004年10月刊

オフショア開発、先人の知恵を有効活用できます

近年、オフショア開発は増加し、組込みソフトウェア開発会社でも約3割がオフショア開発を行っており、計画中のものをを含めると5割近くなる。オフショア開発を成功に導くのは、問題点と対策をキチンと認識することであり、その中でもブリッジSEという役割が重要である。

本書は受発注の双方、日本から出す側、海外で受ける側の両視点から知見が述べられている。これまで雰囲気はわかっていたても具体的に明示された文献を目にすることはなかったが、本書では、オフショア開発失敗に関する特性要因等により体系的に整理されている。またETSSキャリア基準で定義しているブリッジSEのスキルに関して、職務毎に求められるスキルの傾向等の解説があり参考になる。

オフショア開発に不向きな要件を、期間やライン数等を

使い定量的に表現していることにも注目したい。しかし、この不向きな要件に該当する、組込みソフトウェアのなんと多いことが。

本書は、私がぼんやりと考えていたことを、ズバリと具体的な事例を挙げつつ解説・整理している。よって、これまで私の中に蓄積された情報が一気に整理整頓されることでの気持ちの良さを感じる。

このような書籍が増えることは知識の共有として望ましい。直面する問題に対して、企業の枠を超えた知見を集め、整理することがますます増えて欲しい。先人の知恵を有効活用し、同じ失敗を繰り返さないことが、開発力強化に必要なことは間違いのないのだから。

(渡辺 登)



ソフトウェア・エンジニアリング関連イベントカレンダー

作成：SEC journal編集委員会

開催年月	開催日	イベント名	主催	開催場所	URL
2006年 9月	5日(火)	連続セミナー2006 第3回 「情報システム構築アプローチ」	社団法人 情報処理学会	東京都千代田区・ 東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
	14日(木)	第25回ソフトウェア品質シンポジウム	財団法人 日本科学技術連盟	東京都江東区・ 東京ファッションタウン (TFT)ビル	http://www.juse.or.jp/
	18日(月)	ASE2006(21st IEEE/ ACM International Conference on Automated Software Engineering)	国立情報学研究所、IEEE、ACM ¹	東京都千代田区・ 国立情報学研究所 学術研究センター	http://www.ase-conference.jp/
10月	2日(月)	情報化月間記念特別行事	経済産業省	東京都港区・ 全日空ホテル	http://www.ipa.go.jp/
	5日(木)～ 7日(土)	ネットワーク・セキュリティ・ワークショップ in 越後湯沢 2006	NPO新潟情報セキュリティ協会 (ANISec)	新潟県湯沢町・ 湯沢町公民館 / イナモト旅館	http://www.yuzawaonsen.gr.jp/conf/
	11日(水)～ 13日(金)	SPI Japan 2006(ソフトウェア プロセス改善カンファレンス第4回)	日本SPIコンソーシアム (JASPIC)	茨城県つくば市・ つくば国際会議場	http://www.jaspic.jp/
	17日(金)	SEC主催セミナー (テーマ:定量データと共同研究成果)	IPA/SEC	東京都千代田区・ (株)三菱総合研究所 2F 大会議室 A	http://sec.ipa.go.jp/
	17日(火)	ソフトウェアテストシンポジウム2006 札幌	特定非営利活動法人 ソフトウェアテスト技術振興協会 (ASTER) JaSST'06 in Sapporo 実行委員会	北海道札幌市・小樽商科大学 札幌サテライト	http://www.jasst.jp/
	18日(水)～ 20日(金)	Security Solution2006	日経BP社	東京都江東区・ 東京ビッグサイト	http://expo.nikkeibp.co.jp/secu-ex/
	19日(木)	連続セミナー2006 第4回 「情報システム部門のマネジメント」	社団法人 情報処理学会	東京都千代田区・ 東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
	19日(木)～ 21日(土)	ソフトウェアエンジニアリング シンポジウム2006(SES2006)	社団法人 情報処理学会 ソフトウェア工学研究会	東京都江東区・ 日本科学未来館	http://ses2006.sys.wakayama-u.ac.jp/
	19日(木)～ 21日(土)	組込みシステムシンポジウム 2006(ESS2006)	社団法人 情報処理学会 ソフトウェア工学研究会	東京都江東区・ 日本科学未来館	http://ess2006.media.kyoto-u.ac.jp/cfp.php
	24日(火)	IPA Forum 2006	IPA	東京都港区・ 明治記念館	http://www.ipa.go.jp/
	30日(月)	SEC主催セミナー (テーマ:組込み開発の形式検証)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
	31日(火)	SEC主催セミナー (テーマ:組込み開発のプロセス)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
	11月	15日(水)～ 17日(金)	Embedded Technology 2006 / 組込み総合技術展	社団法人 組込みシステム技術 協会(JASA)	神奈川県横浜市・ パシフィコ横浜
16日(木)		連続セミナー2006 第5回 「経営戦略とIT戦略」	社団法人 情報処理学会	東京都千代田区・ 東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
28日(火)		SEC主催セミナー (テーマ:組込み開発の プロジェクトマネジメント)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
12月	6日(水)	連続セミナー2006 第6回 「情報システム部門の役割と人材育成」	社団法人 情報処理学会	東京都千代田区・ 東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
	22日(金)	SEC主催セミナー (テーマ:組込み開発のプロダクトライン)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
2007年 1月	26日(金)	SEC主催セミナー (テーマ:組込み開発の利用品質)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/

¹ ACM Association for Computing Machinery

上記は変更される場合があります。参加の際に必要な詳細事項は主催者にお問合せをお願いします。

イベント報告

Embedded Technology West 2006 (5月10日～11日大阪・マイドーム大阪)：SEC-Web利用登録者の受付(来場者4,080名中367名)。最終日「SECセッション」では105名(満席)がご参加。

SEC Forum 2006 (6月12日～13日東京・大手町サンケイプラザ)：事前登録開始後3日間で定員400名が満席。

SODEC、ESEC (6月28日～30日東京・東京ビッグサイト)：アンケートご協力者へ発行書籍等配布(各1,400名)。Webからアンケート事前ダウンロード各100件超。

配付資料等、詳しくは、SEC-Webサイト(<http://sec.ipa.go.jp/>)をご覧ください。

編集後記

ICSE 2006 (ソフトウェアエンジニアリングにおいて最も権威のある国際会議とされています;2006年5月に上海で開催)において、久しぶりに日本からの提出論文が採択され、今号では同論文を翻訳し「ハイブリッドなコスト見積りモデルの反復的な構築方法について(和文タイトル)」として掲載いたしました。投稿者はAdam Trendowicz、Jens Heidrich、Jürgen Münch (IESEメンバー)、石谷靖、横山健次、菊地奈穂美 (SEC研究員) の6名でIESEとSECの共同研究成果です。また、神谷研究員による、海外レポート「ソフトウェア工学国際会議 ICSE2006 上海に参加して」も掲載いたしました。

今後のイベントでは、SEC journal論文発表会「IPA Forum 2006 (10月24日東京・明治記念館)」と、「Embedded Technology 2006 (11月15日～17日横浜・パシフィコ横浜)」に出展を予定しており、組込みソフトウェアに関連した事業を行っているIPAの他部署も出展予定ですので、是非ご来場頂きたくお願いいたします。

最後に、SEC journal論文投稿のお願いです。本SEC journal掲載論文は、厳しい査読体制が認められ、ドクター論文の投稿先として認められ始めております。社会人ドクターを目指している方々からの投稿も広くお待ち申し上げております。(ヒゲ)

本journalに対してのご意見は、<http://sec.ipa.go.jp/index.php>内の「SECへのお問い合わせ」をご利用ください(ご意見用メールアドレス<sec-journal@ipa.go.jp>でのご意見受付は終了しました)。

SEC journal 編集委員会

編集委員長

猪狩 秀夫

編集委員 (50音順)

青木 奈央

赤田 眞弓

菊地奈穂美

新谷 勝利

田丸喜一郎

樋口 登

松浦 清

神谷 芳樹

門田 浩

渡辺 登



SEC journal[®] 第2巻第3号(通巻7号) 2006年9月11日発行

© 独立行政法人 情報処理推進機構 2006

編集兼発行人 〒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
独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター 所長 鶴保 征城
Tel.03-5978-7543 Fax.03-5978-7517
<http://sec.ipa.go.jp/>

編集・制作 〒101-8460 東京都千代田区神田錦町3-1 株式会社オーム社 Tel 03-3233-0641

※本誌は、「著作権法」によって、著作権等の権利が保護されている著作物です。
※本誌に掲載されている会社名・製品名は、一般に各社の商標または登録商標です。

SEC journal 論文募集

独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センターでは、
下記の内容で論文を募集します。

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

- 開発現場への適用を目的とした手法・技法の詳細化・具体化などの実用化研究の成果に関する論文
- 開発現場での手法・技法・ツールなどの様々な実践経験とそれに基づく分析・考察、それから得られる知見に関する論文
- 開発経験とそれに基づく現場実態の調査・分析に基づく解決すべき課題の整理と解決に向けたアプローチの提案に関する論文

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
コーディング作法
テスト/検証技術
要求獲得・分析技術、ユーザビリティ技術
見積り手法、モデリング手法
定量化・エンピリカル手法
開発プロセス技術
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

論文の評価基準

- 実用性(実フィールドでの実用性)
- 可読性(記述の読みやすさ)
- 有効性(適用した際の効果)
- 信頼性(実データに基づく評価・考察の適切さ)
- 利用性(適用技術が一般化されており参考になるか)
- 募集テーマとの関係

応募要項

スケジュール

SEC journal掲載論文

- 9号(2007年1月発行予定)
応募締切 2006年11月20日
投稿締切 2006年11月27日
採録通知 2006年12月14日
- 10号(2007年4月発行予定)
応募締切 2007年2月19日
投稿締切 2007年2月26日
採録通知 2007年3月13日

● 採録決定後、1週間程度のカメラレディ期間があります。

詳細は別途通知されます。

両募集とも、採録の場合には「SEC journal」への掲載およびIPA SECのWebやイベント等での発表を行います。

提出先

独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター内「SEC journal」事務局

eメール:sec-ronbun@ipa.go.jp

その他

- 論文の著作権は著者に帰属しますが、採択された論文については「SEC journal」への採録、ホームページへの格納と再配布、論文審査会での資料配布における実施権を許諾いただきます。
- 提出いただいた論文は返却いたしません。
- 応募時の個人情報の取扱いは下記のとおりです。SEC内の審査事務局にて管理し、論文審査に係わる査読委員、審査委員とSECが行う広報活動(論文公募、各種イベントの案内、実態調査等を依頼)で使用することを許諾いただきます。

応募様式

応募様式は、下記のURLをご覧ください。



<http://sec.ipa.go.jp/secjournal/oubo.php>

SEC journal バックナンバーの ご案内

詳しくは
<http://sec.ipa.go.jp/secjournal/>
をご覧ください。

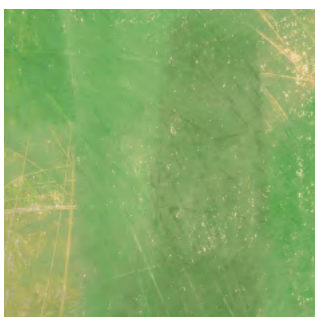


SEC Journal No.7
第2巻第3号(通巻7号)
2006年9月11日発行 ©独立行政法人 情報処理推進機構

編集兼発行人

〒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
独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター
所長 鶴保 征城

Tel.03-5978-7543 Fax.03-5978-7517
URL: <http://www.ipa.go.jp/>
定価1,470円(本体1,400円)



IPA®

独立行政法人 情報処理推進機構

