

## ウィンターワークショップ in 神戸報告

中谷 多哉子<sup>†</sup> 松田栄之<sup>‡</sup> 横山和俊<sup>‡</sup> 藤井拓<sup>§</sup> 松下誠<sup>\*\*</sup>  
平山雅之<sup>††</sup> 青木利晃<sup>††</sup> 大木幹雄<sup>§§</sup> 佐伯元司<sup>\*\*\*</sup> 綿引健二<sup>\*\*\*</sup>

1997 年来毎年開催されてきたソフトウェア工学研究会主催のウィンターワークショップ をスペースアルファ神戸にて 2003 年 1 月 23 日, 24 日の 2 日間開催した。本稿では, 組込みソフトウェア, 要求工学, 軽量ソフトウェアプロセス, Web サービス技術, ソフトウェア可視化技術の各ワークショップで行われた討論内容を報告する。

## A Report on Winter Workshop in KOBE

Takako Nakatani<sup>†</sup> Shigeyuki Matsuda<sup>‡</sup> Kazutoshi Yokoyama<sup>‡</sup> Taku Fujii<sup>§</sup>  
Makoto Matsushita<sup>\*\*</sup> Masayuki Hirayama<sup>††</sup> Toshiaki Aoki<sup>††</sup> Mikio Ohki<sup>§§</sup>  
Motoshi Saeki<sup>\*\*\*</sup> Kenji Watahiki<sup>\*\*\*</sup>

WinterWorkshop is a anual event held by SIGSE since 1997. This year, it was held in Space Alpha KOBE from January 23<sup>rd</sup> to 24<sup>th</sup>. For this paper, the workshop is reported by five groups: Embedded System, Requirements Engineering, Lightweigt Software Process, Web Service Technology and Software Visualizing Technology.

### 1. はじめに

ソフトウェア工学研究会では, 1997 年以来, 毎年ワークショップを開催し, テーマを絞った集中的な議論の場を持ってきた。本年度のウィンターワークショップは, 富士ゼロックスのスペースアルファ神戸で二日間に渡り, 組込みソフトウェア, 要求工学, 軽量ソフトウェアプロセス, Web サービス技術, ソフトウェア可視化技術の 5 つの討論グループに分かれて討論を行った。各分野は, ソフトウェア工学の研究者および実務者の注目を集める分野であり, 将来の発展も期待されている。参加者は欠席者を除いて 53 名となり, 討論の時間不足という参加者の意見も多かったが, いま何をやるべきか, そのために解決すべき研究・技術課題は何かを, 集中的かつ活発に討論できたと思う。

### 2. 討論報告: Web サービス

#### 2.1 討論の目的

Web サービス技術は, インターネット上に公開されたサービス(ソフトウェア部品)を, 疎な関係で(動的に)結合させる技術である。アプリケーションの開

発プロセスを改善しソフトウェア開発の効率化に大きく貢献する期待がされていると共に, オープンなビジネス取引基盤への期待が高まっている。現在の Web サービス技術は, 基本的な規約がほぼ制定され, その拡張機能の検討が行われる段階に達していると考えられる。また, パブリック UDDI が公開され, 企業間取引に SOAP を使うなど, Web サービス技術を適用したビジネス取引システム構築も始まっている。

本トラックでは, 産学の技術者が集まり, Web サービスの現状の課題や将来の展開について討論を行った。具体的には, SOAP, WSDL, UDDI, ebXML などの基本的なサービス技術の導入方法や導入の課題に加えて, 安全に Web サービス技術を利用する企業間取引のためのセキュリティ, サービスシナリオの記述方式, ビジネスモデリングについて議論した。また, Web サービスの適用の新しい領域の可能性についても議論を行った。

#### 2.2 参加者と討論内容

##### 2.2.1 討論参加者

討論参加者は討論リーダーの松田栄之(NTT データ), 横山和俊(NTT データ)をはじめとした, 青山幹雄(南山大学), 和泉憲明(産総研), 今城哲二(日立),

<sup>†</sup> エス・ラゲーン

<sup>‡</sup> NTT データ

<sup>§</sup> オージス総研

<sup>\*\*</sup> 大阪大学

<sup>††</sup> 東芝

<sup>††</sup> 北陸先端大学院大学

<sup>§§</sup> 日本工業大学

<sup>\*\*\*</sup> 東京工業大学

坂田祐司（NTT データ），田中達雄（野村総研），新田英之（日立），本俊也（日立ソフト），福田直樹（静岡大学），藤原克哉（秋田大学），松塚貴英（富士通研），山縣明子（日立中国ソフト），由良俊介（NTT），横関大子郎（NTT），吉田英嗣（NTT データ）の 16 名であった。大学からは 3 人，研究開発機関（部門）から 9 人，ベンダから 4 人の参加者となった。

### 2.2.2 討論内容

本トラックには 12 件のポジションペーパーがあり，トラックの前半で，全部のポジションペーパーの紹介を行った後，ポジションペーパーを分析して，以下の 3 つに分類してグループごとに討論した。

- ・ Web サービス技術の実用性（5 件）  
Web サービスを実際に適用するためのモデル化の検討および Web サービスを適用して実証に関する議論
- ・ 本来の Web サービス機能を活かす動的な Web サービス環境の課題（5 件）  
ビジネスプロセスの設計，既存システムの Web サービス化，セキュリティの確保に関する議論
- ・ Web サービス技術の新しい展開領域（2 件）  
Web サービス技術の分散コンピューティング環境への適用に関する提案

各グループでは，各テーマの課題について議論したが，時間が十分取れなかったために，課題を出しただけで，方向性や結論までたどり着かなかった項目もあった。

## 2.3 討論成果

### 2.3.1 Web サービス技術の実用性に関する討論結果

Web サービス技術の中で，主に SOAP を用いた実用システム構築上の課題を議論した。

#### (1) システム開発コストの観点

Web サービス（SOAP）を利用したシステム開発の開発コストは，新規開発の場合，既存のシステム構築コストとあまり変わらない。そのため，再利用可能な Web サービスの充実が必要である。また，既存のビジネスコンポーネントを利用するために，半自動で Web サービス化する仕組みが望まれている。ただし，Web サービス化を行うためには，ビジネスコンポーネントの粒度，品質，信頼性に関して考慮する必要がある。

#### (2) 性能面の観点

SOAP を利用することによる性能の問題はないと考えられる。しかしながら，XML ドキュメントをデータベースに保存する際の処理がボトルネックになる可能性がある。そのため，高負荷なトランザクション処理へ

適用する場合，データベース等の既存パッケージとの接続検証を十分に行う必要がある。

#### (3) 相互接続性

Web サービスの仕様のバージョンアップがあったときの実装製品の追従方法が問題である。また，Web サービス技術の仕様およびバージョンがいろいろ出てきており，どの仕様に準拠しているのかを通知する仕組みが必要である。相互接続に関しては WS-1 (Web Services Interoperability Organization) の活動に期待する。

#### (4) 実ビジネスへの適用への阻害要因

実際のビジネスに本格的に Web サービスを適用するためには，セキュリティやトランザクションの仕様が制定され，実装製品が出てくる必要がある。また，適用事例が少ないため，適用方法が分からないという指摘もある。既存システムが Web サービス化され再利用可能となると，ビジネスコンポーネントを利用してユーザが簡単にサービスを構築できるようになり，システムインテグレータの仕事がなくなるので，普及しないのではないかという意見もある。一方，複雑なシステムを構築できるようになり，ますますシステムインテグレータの仕事が増えるので，Web サービスの普及に弾みがつくという意見があった。

### 2.3.2 動的なサービス連携に関する討論結果

Web サービスが本来指向していた動的なサービス連携に向けての課題を議論した。

#### (1) ビジネスプロセスのモデル化・設計

動的なサービス連携のためには，標準的なビジネスプロセスへ参加できる Web サービスを迅速に構築することが重要である。現在，ビジネスプロセスに関して，様々なビジネスプロセス記述仕様が提案されている。これら仕様では，組織間にまたがる取引手順の記述から組織内の手続きを記述するものまで，サービスの粒度や記述レベルが異なっている。また，計算機上での実装を意識した記述や，手順のみの記述など仕様が乱立している状態であり，動的なサービス連携に向けた Web サービスの構築を阻害していると考えられる。そのため，サービスの粒度を意識したサービス連携のモデル化が必要である。例えば，組織間のビジネスプロセスモデル，サービス実装などの階層構造によるモデル化を進めなければならない。異種のサービス間でのセキュリティ確保や相互接続も重要な課題である。

#### (2) 標準化

Web サービス技術の標準化は，IBM，マイクロソフトといったベンダが検討した仕様を中心に展開されてい

る。また、W3C や OASIS のような標準化団体や非営利団体によって、検討、承認された仕様もある。両者の動向についても注意する必要がある。

#### 2.4 将来の Web サービスの展開

##### (1) 次世代の Web サービスの姿

サービスプロバイダとコンシューマが独立して進化を続け、高いダイナミズムを実現する。また、サービスおよびデータはさらに分散する。これらの有機的な結合を実現するために、ネットワーク、サービスなどに意味を持たせるようになる。これらは、ユビキタスの世界を実現する要素となる。

##### (2) アプローチの方法

個々の要素が独立に進化し、かつ有機的に結合できる必要がある。そのために、疎な関係を保つ連携モデルが必要になる。グリッドコンピューティングは、疎な連携モデルの対象の一つである。疎な連携モデルを実現するインフラとして Web サービスのオープン性と連携技術を利用できる。ただし、初めから重要なシステムに導入するのではなく、非ビジネス領域から導入して、実績を積み込む必要がある。

##### (3) 次世代 Web サービス実現の課題

疎な関係を保ちつつ有機的な結合を目指すために、それぞれの要素に対する意味づけが必要になってくる。Semantic Web との関係が重要な検討項目になる。リソースに着目するとデータ量が膨大になり、処理量も増大する。従来の Web サービスからのスケーラビリティの確保とともにハードウェアおよびソフトウェアの双方の進化が必要になってくる。また、疎な関係の中で、有機的な結合を行うためにセキュリティおよびプライバシーの確保も重要な課題となる。

#### 2.5 今後の動向

Web サービス技術は、基本的な技術の成熟を終え、世の中で実用化され始めている。現在は、異なる組織のシステムをインターネットにより接続するシステム構築基盤として利用が本格化している。そのため、Web サービスが本来指向していた動的なサービス連携の基礎が固まってきたと考えられる。この動向が、一時的な流行にならないように、サービス提供者が容易にサービス連携へ参加できるビジネスプロセス設計方法の確立が急務である。そのため、ビジネスプロセスの階層化に基づくモデル化やビジネスプロセスから Web サービスを実装するシステム構築方法論が今後重要な技術開発分野である。また、Web サービスの新しい展開として、グリッドやユビキタスの融合を考え、新しい分散コンピューティングアーキテクチャのインフラと

しての展開を検討していく必要がある。

### 3. 討論報告：組み込みソフトウェア

#### 3.1 目的

現在、日本で開発される工業製品の多くには、何らかの形でコンピュータが組み込まれている。これに伴い製品に組み込まれる組み込みソフトウェアの比重が急速に拡大する傾向にある。しかし、その一方でこのような組み込みソフトウェアの開発を支援する方法論・ツールなどは必ずしも十分に整備されておらず、組み込みソフトウェア領域でのソフトウェア工学は未開の分野になっている。

このトラックでは、こうした問題意識のもと、「組み込みソフトウェア開発のどのようなところに課題があり、それを如何にして工学として解決していくか」というテーマで討論を行った。

#### 3.2 参加者と討論内容

##### 3.2.1 参加者

本トラックへの参加者は、青木利晃（北陸先端大）と平山雅之（東芝）の討論リーダーと、安部田章（九州日立マクセル）、今城哲二（日立）、今関剛（豆蔵）、片山卓也（北陸先端大）、川田哲也（東京電機大）、河野善彌（元埼玉大）、佐々木明博（デンソー）、佐藤啓太（デンソー）、杉浦英樹（富士ゼロックス）、中島震（法政大）、橋本隆成（ソニー）、渡辺晴美（南山大）の 12 名の参加者によって行われた。

メンバーは大学から 6 人、企業から 8 人となっており、それぞれのニーズや問題意識、シーズなどをディスカッションするには適したメンバー構成であった。

##### 3.3 討論内容

ワークショップでの討論時間は実質約 7～8 時間と限られており、かつ、メンバーも初対面であった。このような条件の中で、冒頭に設定したテーマの解を早急に求めるのは難しいと判断し、長期的活動の一部としてワークショップの討論を位置付けた。長期的活動とは、2003 年度よりソフトウェア工学研究会の組み込みソフトウェアワーキングを立ち上げることを前提としたものである。

このためワークショップでは、組み込みソフトウェアワーキングの方向付けを目的として、その意識合わせ的な側面を重視して議論を行った。参加者それぞれが、「組み込みソフトウェア」をキーワードに、企業側の参加者は「それぞれが考えている/抱えている組み込みソフトウェアの課題」などを、大学側の参加者は「現在進めている研究」などについて報告を行った。討論の主

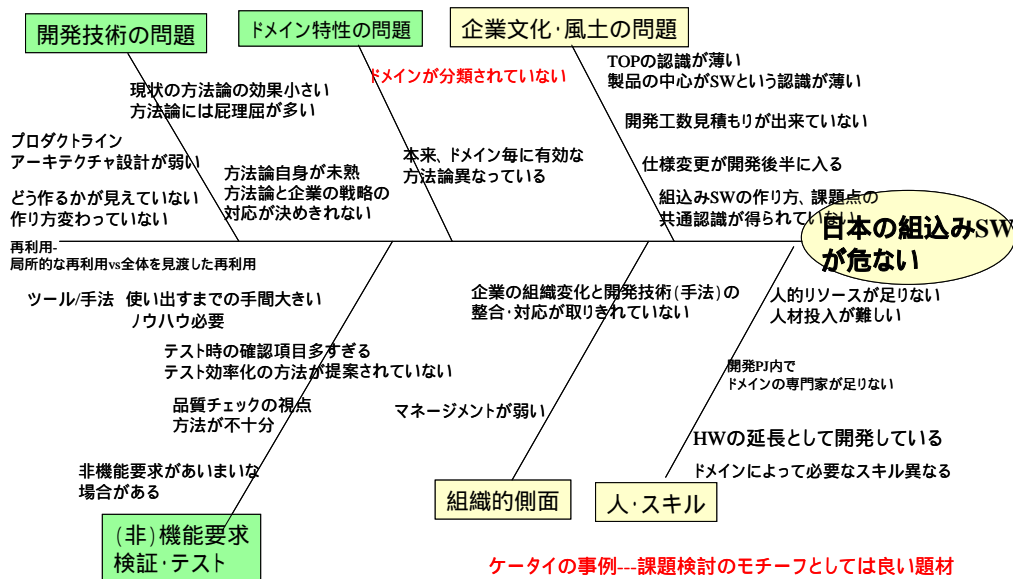


図1 携帯電話を事例とした特性要因図

なポイントは、(1)組み込みソフトウェア開発における要求フェーズの課題、(2)アーキテクチャとプロダクトライン、ソフトウェアの再利用、(3)設計・検証、(4)プロジェクト管理とした。

(1) 組み込みソフトウェアの要求フェーズ

組み込みソフトウェアの領域の大きな問題の1つとして要求獲得の問題がクローズアップされた。組み込みソフトウェアでは最終的にハードウェアと一体となった動作を確認した時点で、顧客の要求が変更されることがしばしばあり、こうした要求の変更を如何に柔軟に裁くかが重要な課題になっているとの共通認識が得られた。

(2) アーキテクチャとプロダクトライン、再利用

如何にして組み込みソフトウェアを効率的に開発するかという問いに関して、そのアーキテクチャを整理し、再利用可能な形に整えることで、プロダクトラインの考え方に基づいた開発スタイルを指向すべきというディスカッションを行った。その一方で、ビジネスソフトウェア領域では今や当たり前に使われているオブジェクト指向技術などが、組み込みソフトウェア開発の場合、今ひとつうまく利用できていないといった課題なども共通認識した。また、これらの成功体験の共有化など技術情報の流通などの面でも、組み込みソフトウェアの開発担当者はあまりの忙しさに眼が向けられていないといった課題も再認識できた。

(3) 設計・検証

設計・検証は、国内の大学で組み込みソフトウェアを対象に

なポイントは、(1)組み込みソフトウェア開発に研究が進められている数少ない技術の1つである。研究としては形式的仕様をベースに、モデルチェックを行うものであるが、実際の適用モチーフなどの面で、より産学がうまく連携して、その成果を利用する方法などの議論を行った。組み込みシステム規模の増大に伴い、その全てを形式的仕様で表現しきれないといった問題や、このような方式で仕様を表現すること自体の難しさなど、今後、連携して解決していくべき課題の一部も明らかになった。

(4) プロジェクト管理

ソフトウェア開発では開発者の能力や組織に依存する要因が多い。プロジェクト管理もその1つであるが、こうしたソフトウェア開発に関する属人性をどのようにして薄めていかをもう1つの大きな課題として認識した。一方で、組み込みソフトウェア開発がうまくいかない理由を、この属人性のみ求めてしまうと、工学的なアプローチの芽を摘んでしまうことにもなりかねない。

3.4 成果

ワークショップでは、上記の5つのトピックについて、参加者の本音に近いレベルでのディスカッションを行った。今回のワークショップは組み込みワーキングを足させるにあたっての、メンバーのポジションの確認と意識共有などを主たる目的とした。このためディスカッションの結果については、組み込みソフトウェアに関する課題を特性要因図として整理し、ワーキングの方向付けのための資料とすることにした。

特性要因図の切り口として「開発技術の問題」「ドメイン

特性の問題」「機能要求、検証・テスト」「企業文化・風土の問題」「組織的側面」「人・スキル」などの切り口でディスカッションの中でたキーワードを整理した。

このうち「開発技術の問題」「ドメイン特性の問題」「機能要求、検証・テスト」については特に工学的なアプローチがふさわしい領域と考えられる。一方で、「企業文化・風土の問題」「組織的側面」「人・スキル」などはソフトウェア工学に根ざした技術論以外の要因をも含めたより広い見地からの検討が必要な領域と考えられる。

また今回のワークショップでは「組込みソフトウェア」といっても様々な業種・業態のメンバーが参加した。このため、まず、「組込みソフトウェア」に関する定義と分類を行う必要があるとの共通の意見でまとまった。

### 3.5 今後の方向

今回のワークショップは組込みソフトウェアワーキンググループの発足準備の第1段階として位置付けられる。このワーキングの成果をもとに、2003年4月より組込みワーキンググループが正式に発足した。ワーキンググループとして組込みソフトウェア開発技術の体系化、および組込みソフトウェア開発技術という2つのタスクフォースを走らせることとなった。前者は、今回のワークショップで作成した特性要因図を拡張して、組込みソフトウェア開発における課題の整理分類、およびそれらに関するソリューションのマップ化などを行う。後者は検証技術など組込みソフトウェア開発の課題を実際に解決するための技術の深耕をととして産学の連携を加速させることを狙っている。

## 4. 討論報告：軽量ソフトウェアプロセス

### 4.1 目的

本ワークショップでは、2年前より継続してソフトウェア開発プロセスに関するセッションを設け、その時期に注目されている話題を取り上げて討論を行ってきた。3回目となる今回のワークショップでは「軽量ソフトウェアプロセス」と題し、オープンソース開発、XP(eXtreme Programming)、Agile Processなど、ここ数年で急速に普及しつつある、比較的軽い作業を連続的に繰り返す形の開発プロセスをテーマとして取り上げることとした。

今回のセッションは、軽量ソフトウェアプロセスを実際に用いて開発を行っている方々を対象として、以下のような議論のテーマを設定した。

- 「軽量」とは具体的に何が軽量なのか。
- この種の手法の持つ本当に新しい点はなにか。
- 軽量ソフトウェアプロセスを新規に導入する際の注意点や問題点にはどのようなものがあるか。
- 軽量ソフトウェアプロセスに欠けているとされる品質管理、スケジュール管理をどうするか。

参加者には事前にポジションペーパーの提出をお願いして、軽量ソフトウェアプロセスへの取り組み方や現状に対する意見

などをあらかじめ表明してもらうこととした。

### 4.2 参加者

本セッションへの参加者は8名であった。他のセッションに比べると参加者は少なめであるが、全員が積極的に参加するためには、この程度の人数が適切ではないかと思われる。参加者は、藤井拓(オージス総研)、松下誠(大阪大学)を討論リーダーとした以下の方々であった。河合克己(日立)、川端光義(アジャイルウェア)、塚田三緒(富士通コミュニケーション・システムズ)、牧野克治(三菱電機マイコン機器ソフトウェア)、宮内茂樹(富士通コミュニケーション・システムズ)、鷲崎弘宜(早稲田大学)



図2 軽量ソフトウェアセッションの討論風景

### 4.3 議論の流れ

我々に与えられた時間は初日の午後および夜と二日目の午前であったため、大きく分けて3回の討論枠があると考えることができた。そこで、時間の配分を枠ごとに行うこととした。

最初の枠では各人のポジションペーパーの内容について紹介をしてもらい、その内容についていくつかの質疑を行った。この質疑を通じて、今回のセッションでどのような話題に集中して討論を行うかを決定することとした。

夕食をはさんで夜になってからの枠では、前枠で取り上げた議論テーマについてさらに突っ込んだ討論を行った。翌日午前となる最後の枠では、昨日の議論に対する総括を行い、将来への課題について議論を行った。

1泊2日と短い討論時間ではあったが、初日の夜は会場の制限時間いっぱいとなる午後11時まで議論を行うなど、終始白熱した議論が続けられた。

### 4.4 議論のまとめ

今回のセッションでは、「軽量ソフトウェアプロセス」というタイトルの通り、特定の技術に依存せず、いわゆる lightweight process 全般を範囲とするつもりであったが、結果的には XP のみを取り上げた格好となった。これは、近年 XP の考えが広く理解されており、その考え方を生かそうとする参加者が多く参加していたことがその理由として考えられる。このため今回の議論、XP の考え方を既存のソフトウェア開発の中で生かす取り組みの考案、その実践、結果の評価を通じて、軽量ソフトウェアプロセスの現在、そして将来の姿を模索する議論となった。

以下、本セッションの議論の中で得られた、XP を中心とする軽量ソフトウェアの現在と将来に関する議論について、議論の時系列に沿って紹介する。紙面の都合上、XP 自体の紹介については割愛する。

#### 4.4.1 現状分析およびXP がもたらした効果

特に Web 関連のソフトウェア開発で典型的であるが、従来のソフトウェア開発と比較して非常に短い納期を要求されることが多い。従来のプロセスでは、開発作業が確実に行われることを保障するために、いくつかの最終成果物には直結しない生成物に対する作業があるが、納期を短縮するためには、これらの作業を行うだけの時間を確保できないという切実な問題があげられる。また、開発規模に応じて人員が増えた場合に、全体を管理しきれず品質や生産性が低下するといった問題も指摘された。

また、予測できない市場ニーズに沿った開発を行うためには、ニーズの変化に応じて即座に追従できる開発体制が必要であることや、プログラム作成を効率的に行うために、開発や検証の効率を向上したいという要求も挙げられた。

実際に XP を開発現場に適用する際に、抵抗感が組織や個人に生まれる例が紹介された。新しい手法に対する抵抗感は一般的に存在するが、XP の手法が従来の考え方を根本的に否定するようになってしまったために、その抵抗感が非常に強いものになる傾向があることが指摘された。また、仮に XP を受け入れる環境ができたとしても、具体的にどのようにして取り入れていけばよいかという方法に苦慮することも指摘された。

多くの参加者から、ペアプログラミングを導入したことによって、最終生成物におけるコード量が削減できたことが挙げられた。また、常に作業に対するチェックが行われるということを理由として、生成物の品質も向上したという例も紹介された。ただ、XP の持つ特徴はペアプログラミングだけに限ったものではない。同時にいくつかの XP で述べられている要素が既存の開発環境に持ち込んでいることや、作業者の意欲が新しい開発手法によって高められたことなどもあり、何がコード量削減や品質の向上に寄与したか、現状では因果関係を明確にできているわけではないことも確認した。

#### 4.4.2 将来への課題

初日の議論を踏まえ、二日目には今後 XP を実践するにあたっての解決すべき課題について議論を行った。議論は多岐にわたったが、主に以下の 5 つのトピックについて話が進められた。

- XP の教育方法について

現状は企業における開発改善運動などに依存した活動が行われている程度であり、大学や学会などにおける活動が十分であるとはいえない。いかに XP の考え方を初心者に教えるか、という問題に取り組んで欲しいという要望が企業の方などからあげられた。また、XP を既存のソフトウェア工学の枠組みからとらえなおすことにより、既存の XP に不足している側面を整理することができるのではないかと考えられる。

- 事例の整理・体系化・定量化

本セッションでは、複数の企業による XP 適用事例が報告された。しかし、日本における XP の適用事例はまだ数が少なく、散発的に行われているのが現状である。海外で行われている取り組みに習い、XP 適用事例を統一的な枠組みを用いて収集、整理することによって、新たに XP を適用する組織に対する情報提供ができる。また、XP が述べている項目と実際の効果との対応関係や、XP の項目間の依存関係を整理することにより、XP 適用時の効果をより高めることができると考えられる。

- 品質管理

ソフトウェアの品質に対する考え方として、すでに ISO9000 や CMM などが浸透している。これらの規格が述べている立場と XP の立場は確かに大きく異なっているが、ソフトウェア開発を対象として、その効率を高めるという観点では一致しており、両者は矛盾するものではない。「どちらが良いか」という優劣論ではなく、XP が指摘している内容は、たとえば ISO9000 ではどのように説明されているのかといった議論が今後必要ではないかという指摘があった。

- ソースコードとドキュメントとの関係

XP では、ソースコードやテストデータなど、従来では仕様などのドキュメントよりも後に書かれるものを先に記述する傾向がある。このような場合、ソースコードをどのように可視化するか、ドキュメントとの整合性をどのように確保するかといった問題点が残されていることが指摘された。

- 日本人とアジャイル

欧米社会に見られる契約中心のソフトウェア開発と異なり、日本では契約の範囲外になるような要求の変更などにも応じてきたという経緯を指摘した上で、実は日本人こそが XP の目指している軽量ソフトウェアプロセスを昔から実践していたのではないかと、という考え方が示された。しかし、日本の開発組織が硬直化したことにより、組織自体がアジャイルの要素を失いつつあるという指摘が行われた。XP の手法を通じて、組織活動をより柔軟にすることが期待される。

#### 4.5 おわりに

XP をはじめとする軽量ソフトウェアプロセスに関する議論はまだまだ始まったばかりである。今回、ワークショップ中でこのようなセッションを設けることができたことは、非常に有意義であったと考えている。機会があれば、同種の議論を今後継続的に続けていきたいと考えている。

## 5. 討論報告：ソフトウェア可視化技術

### 5.1 目的

物理現象を扱う分野では、現象の背後にある因果規則や複雑な作用を直感的に把握しやすくするため、CG やアニメーションを用いた現象の可視化が行われる。たとえば、障害物の背後で生じる渦の挙動や密度変化等は代表的な事例である。このような可視化によって、現象のもつ意味や洞察を一層深めることができる。

ソフトウェアに関して同様な可視化を試みるとき、物理現象に比較して、「現象の構成要素自体がもつ変化規則」の究明や「構成要素間の関係の変化規則」等の理解が著しく遅れていることに気がつく。実際、ソフトウェアの分類構造や静的な情報量（たとえば、クラス構造やモジュール構造、統計量、各種の分析設計図式等）の可視化は従来から行われてきたが、ソフトウェアのもつダイナミクス（たとえば、バージョンアップがもつ進化の原理や要求仕様が収束するときのプロセス等）に関しては、ほとんど可視化されていない。ソフトウェアの可視化に関しては、いまだプラトンの“what are things made of?”の時代であり、ニュートンの“Why do things move as they do?”の時代にすらいっていないと言えよう。

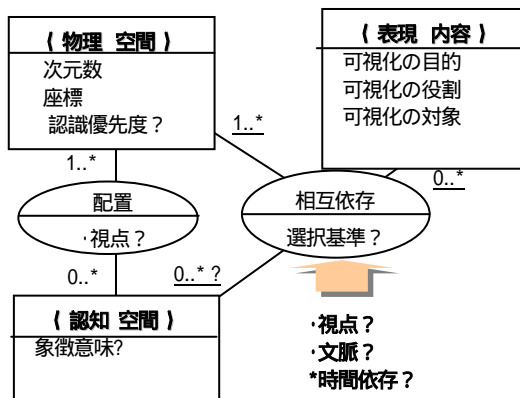


図3 ソフトウェア可視化技術の課題構造

さらにソフトウェアのもつ抽象概念の表現方法が物理現象の表現よりも、人間の認知過程に深く依存していることも可視化技術の研究を困難にしている。これは逆に考えると、人間の認知過程まで立ち入ったソフトウェア工学研究が必要とされている現況において、可視化技術は大きな役割を果たしていることを示している。

そこでワークショップは、図3のような可視化対象および課題モデル図をたたき台として描き、可視化目的と表現空間との対応関係、あるいは可視化目的と認知的・物理的表現空間の選択方法や効果予測等を中心に討論を展開することを予定した。

## 5.2 参加者と討論内容

### 5.2.1 参加者

参加者は大木幹雄（日本工業大学）を討論リーダーとした次の方々であった。臼井義美（臼井技術士事務所）、神林靖（日本工業大学）、黒坂靖子（日本電子計算）、清水誠介（立命館大学）、丸山勝久（立命館大学）。グループは少人数ではあったが、少人数がゆえに精鋭の方々ばかりで、可視化技術がソフトウェアの理解や分析設計・開発に与える効果について、それぞれの立場から活発に意見が出された。

### 5.2.2 討論内容

第1日目は、各メンバのポジションペーパーにしたがって、可視化に関する研究成果と課題の提起が行われた。以下、提起された課題のみを示す。

(1) プログラミング初等教育におけるプログラム動作可視化

### ツールの効果（大木幹雄）

ソフトウェアがもつ特性の状態変化や分析設計内容の動的な変化、プログラムの動作等のソフトウェアがもつ動的側面の可視化が理解に不可欠であるが、その実現は遅れている。

(2) クラス間関係の簡略によるフレームワークの可視化（清水誠介、丸山勝久）

アプリケーションを構成するクラスとフレームワーククラスとの境界に重要度付けを行う理解手法が必要になる。その際に、より有効な表示方法を考慮しなければならない。これはプロダクトの可視化全体について言える。

(3) 分散オブジェクト環境におけるシステムの可視化に関する一考察（臼井義美）

データあるいはオブジェクトの流れにしたがって、ソフトウェアの構成を可視化するとシステムの合成や再利用が容易になる。このような可視化は、人間の思考過程の本質（ものの動きの把握）に根ざしているのではなからうか。

(4) プログラミング環境におけるテキストとグラフィック表現の平衡（神林靖）

可視化において、グラフィカルな表現とテキスト表現の併用が効果を高める上で必要になる。しかしどのような役割分担・目的で使い分けるべきか、定量的な研究や適用基準が明らかになっていない。

(5) AVS/Expressにみるコンポーネント可視化合成環境の課題（黒坂靖子）

可視化ソフトとして世界的に有名な AVS であるが、開発者の最近の関心は、人間の認知過程にまで立ち入った可視化方法に移行している。可視化プログラムがもつ機能の高度化に伴って、可視化コンポーネントの合成空間の次元を2次元から3次元に拡張する必要が出ている。

夕食後は、新たなアイデアやニーズを引き出す上で討論以上の意見交換が行われた。会場で他の討論グループから、最近の移動体システムのトラブルに関して、原因箇所の特定が困難であり、ソフトウェア品質保証や障害対策の面で可視化技術がどのような役割を果たすのかについて問われた。この出来事は翌日の討論に大きな影響を与えた。

第2日目は、1日目の討論を受けて、可視化すべき具体的な対象と優先度から討論を開始した。発散の防止するため、設計仕様の可視化にターゲットを絞り、目的、手段等を考察した。その結果、設計仕様は、開発時のみでなく、「障害発生において原因の特定や影響範囲を特定する」役割の方が大きいとの問題提起がなされた。たとえば、ハード製品やプラント開発では、設計書は開発時より、むしろトラブル発生時に重要な役割をもつ。しかし、ソフトウェアでは、設計書は開発が完了すると無用の長物になる。この違いはどこからくるのであろうか？最近のコピーマシンは、障害発生時に障害箇所と復旧手段を可視化して示してくれる。それを可能としているのは、設計時から障害対策機能の組込であり、人間にとって理解しやすい障害箇所の表示にある。

従来の捉え方では可視化技術は、単なる出力系に関する技術の色彩が強かった。しかし今後は、コピーマシン同様、可

視化技術は、機能や障害を理解する人間に対する入力系として、システム設計時の重要な制約となる。その意味で可視化技術は今後のソフトウェア技術のあり方に変更を迫るキー要因になるかもしれない。

### 5.3 成果

認知過程に立ち込んだ具体的な討論にはいたらなかったが、以下に述べる設計仕様の可視化目的と課題、実現手段等について検討し内容を掘り下げたことは、今後の可視化技術のあり方に一つの方向性を与えた。

#### 5.3.1 設計仕様の可視化に関して

可視化の主な目的は、ユーザ、開発メンバとのコミュニケーション、および理解の促進、障害発生時の原因、影響範囲の特定であるとの認識に至った。

#### 5.3.2 課題

理解しやすい内容の表現形式とは何か。図式表現については、すでに種々の方式が実用に供されているが、クラス図とOCL(オブジェクト制約言語)に見られるような図式とテキストの役割分担の境界線はどこに置くべきか。原因の特定に役立つ設計仕様のもつ「意図」を理解する方法とは何か。意図をどのように表現するか。意図を理解するためには、通常、設計仕様が決まるまでの過程とその履歴(議事録)が必要になるが、どのように表現したら、理解が容易になるか。

#### 5.3.3 具体的な理想像

ソフトウェアに障害が発生すると、その原因箇所と対処方法を理解しやすい形式で可視化してくれることが期待される。そのためには、意図も含めて、障害発生検出と対策の設計手法と設計図式とは何か。プログラム言語の設計との対応付けはどのように行すべきか。設計履歴に含まれる意図の理解手順、意図を理解した対応箇所の特定方法と可視化方法について、課題を解決する必要がある。

### 5.4 今後の方向

引き続きソフトウェア可視化技術を鳥瞰し課題を体系的に捉えると同時に、具体的な可視化対象を選択して議論を今後も地道に重ねる必要がある。今回のワークショップは障害発生と、その原因箇所と対処方法を可視化する方法について討論を重ねたが、今後も可視化の効用について具体的な応用例をとりあげながら、議論を重ねてゆく必要がある。

## 6. 討論報告：要求工学

### 6.1 参加者

要求工学セッションには東工大の佐伯元司、綿引健二を討論リーダーとし、以下の計10名が参加した。大西淳(立命館大)、海谷治彦(信州大)、鎌田真由美(日本IBM)、白銀純子(早稲田大)、中野所武司(明治大)、妻木俊彦(日本ユニシス)、中谷多哉子(Sラグーン)、廣田豊彦(九州産業大)、蓬萊尚幸(セレスター・レキシコ・サイエンシズ)、山田宏之(愛媛大)。

### 6.2 討論内容

最近、要求工学グループでは、要求仕様書の品質メトリックスとその活用法について議論を行っている。要求分析に関

連する品質メトリックスについては、(1)要求仕様書の品質と測定法、(2)要求定義プロセスの品質と測定法、(3)要求仕様書の品質から最終成果物の品質を測定する方法などが考えられる。

(1)については、2001年5月の要求工学ワークショップ(小諸)にて、IEEE 830-1998の中で定義されている要求仕様書の品質特性を対象としてその測定法の議論を行った。(2)についても2001年11月のワークショップ(宇和島)、2002年1月のウィンターワークショップ(伊豆)において取り上げ、その測定法を議論した。今回のウィンターワークショップでは、(3)の品質メトリックスをテーマとして取り上げた。

要求仕様書の品質としては(1)の場合と同様にIEEE 830-1998に定義されている品質特性を、最終成果物つまりソフトウェアの品質としてはISO/IEC9126において定義されている品質(副)特性を用いた。つまり、要求仕様書からIEEE830-1998の品質特性項目が測定されたとして、その結果からソフトウェアのISO/IEC9126品質特性を予測することを考える。この際、仮定として要求仕様書どおりにソフトウェアが作成されるものとした。

これを行うために、IEEE830-1998要求仕様書品質特性を縦軸(行項目)に、ISO/IEC9126ソフトウェア品質特性を横軸(列項目)にとり、セル(i,j)に要求仕様書品質特性iがソフトウェア品質特性jに影響を与える度合いを記入するマトリックスを用意し、相関関係の評価は、相関の強いものから、相関があるとは考えられないものまでの4段階で行うことにした。

例えば、要求仕様書の妥当性が向上すれば、最終製品の合目的性も向上すると考え、“ :相関がある ” が記入されている。また、要求仕様書の追跡可能性が向上すれば必ず最終製品の合目的性も向上するとはいえない(つまり、相関はない)が、合目的性の計測はしやすくなるといえる。このように評価として相関以外の関係がある場合は、それを具体的に記入することにした。

今回の議論の目的は、上記の相関表の作成を行うことである。相関表を作成することには、要求仕様書から最終製品の品質予測ができることや、最終製品のある品質項目を特に高めたいときに、要求仕様書のどの品質項目を高めればよいか分かるといった利点が期待できる。

相関表の作成は、まず、本ワークショップに先立って要求工学セッションの参加者12名それぞれに直感により相関表を完成させてもらい、全員の評価結果を定量化し、その平均点をもとめて当日のワークショップに臨んだ。ワークショップでは、集計結果の平均点を基に議論を行い、表中の各項目間の相関関係の評価を確定した。また、各項目間の相関関係の評価を確定するに当たってその根拠となった情報や、評価の際の仮定や制限事項などの補足情報を併記し、本表の利用者に対し十分な情報を提供できるよう考慮した。

### 6.3 成果

議論の結果得られた相関表は紙面の都合で省略する。ただし、今回は時間の制約上、すべての相関関係の評価について確定することはできなかったため、全168中71のセル(相関関係)



について議論した。

今回の議論の中で、要求仕様書の品質特性とソフトウェアの品質特性の間には、すべての値域において相関がみられるものと、ある特定の値域のみ相関が見られるものがあることが分かった(図4)。特に後者のような品質特性間の相関関係の評価では、参加者の意見が分かれることがあり議論に多くの時間を要した。このような品質特性間の評価は“ : 相関がないとはいえない”とし、どのような仮定のもとで相関関係が見られるかを議論の中で明らかにして、評価値とともに補足情報として記録した。

またこの他にも、どのような種類のソフトウェアを対象とするかによって項目間の評価が異なるという意見が多くみられた。このため、ドメインに特化した相関表を作成することも必要であると考えている。

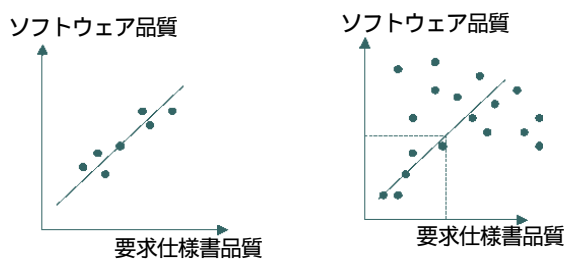


図4 相関関係

#### 6.4 今後の方向

今後、要求工学グループでは、2003年の5月に開催される予定のREWGにおいて相関表の残りの部分について議論を行い、同表の第一版を完成させる予定である。相関表は各参加者の経験によって作成されている。従って、その正しさをなんらかの形で実証することが望まれる。そのためには、実際の開発プロジェクトで要求仕様書やソフトウェア製品の品質項目を客観的に測定し、分析することも必要になってくるであろう。

#### 7. おわりに

最期にワークショップにおいて、活発な討論に参加下さった参加者に感謝する。参加者のアンケート調査によると、討論時間が短いという意見も寄せられている。これらの意見に配慮し、今後のウィンターワークショップをより有意義なものにしていきたい。また、今回参加されなかった方々には、次回以降の参加を是非お願いしたい。ワークショップ他の研究会主催行事については、ソフトウェア工学研究会ホームページを参照されたい。

(<http://www.ipsj.or.jp/sig/se/>)