

## ソフトウェア部品庫 EvoMan におけるツール連携機構

田村 直樹<sup>†</sup> 川名 康仁<sup>††</sup> 山本 康夫<sup>‡</sup> 小林 貴訓<sup>‡‡</sup>

<sup>†</sup>三菱電機(株) 情報技術総合研究所 〒247-8501 神奈川県鎌倉市大船 5-1-1

<sup>††</sup>三菱電機(株) 鎌倉製作所 〒247 神奈川県鎌倉市上町屋 325

<sup>‡</sup>三菱電機(株) 名古屋製作所 〒461 愛知県名古屋市東区矢田南 5-1-14

<sup>‡‡</sup>三菱電機(株) 設計システム技術センター 〒247-8501 神奈川県鎌倉市大船 5-1-1

E-mail: <sup>†</sup>ntamura@isl.melco.co.jp, <sup>††</sup>Yasuhito.Kawana@kama.melco.co.jp, <sup>‡</sup>yamamoto@kan.mei.melco.co.jp, <sup>‡‡</sup>yosinori@dsec.hq.melco.co.jp

あらまし 我々は Web 環境上で運用するソフトウェア部品庫 EvoMan を機能拡張し、Product Line 型ソフトウェア開発を支援する機能を整備している。Product Line 型ソフトウェア開発を支援する環境として、まずソフトウェア資産の理解、変更管理、品質保証の3つの機能を整備した。ここでは、ソフトウェアプロセスの定義とそれに併せた機能拡張を行っている。また、簡単な Web アプリケーション連携の機構を用いてこれらの機能拡張を実現し、開発プロセスの変更への対応を実現している。

キーワード プロダクト・ライン、ソフトウェア部品庫、ソフトウェア構成管理、ソフトウェアプロセス

### Tool Collaboration Mechanism of Software Repository EvoMan

Naoki TAMURA<sup>†</sup> Yasuhito KAWANA<sup>††</sup> Yasuo YAMAMOTO<sup>‡</sup> Yasunori KOBAYASHI<sup>‡‡</sup>

<sup>†</sup> Information Technology R&D Center, Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura-shi, Kanagawa, 247-8501 Japan

<sup>††</sup> Kamakura Works, Mitsubishi Electric Corporation, 325, Ue-Machiya, Kamakura-shi, Kanagawa, 247, Japan

<sup>‡</sup> Nagoya Works, Mitsubishi Electric Corporation, 5-1-14, Yada Minami, Higashi-Ward, Nagoya, Aichi, 461, Japan

<sup>‡‡</sup> Design Systems Engineering Center, Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura-shi, Kanagawa, 247-8501 Japan

**Abstract** We are developing a software repository EvoMan, which is designed to operate in the Web environment. To support Software Product Line approach, a software repository should provide various facilities such as reverse engineering, change control, and quality control. These facilities should be integrated under a description of software development processes. In this paper, we describe the concept of EvoMan and its mechanism for tool collaboration to integrate various CASE tools.

**Keyword** Product Line Approach, software repository, software configuration management, software processes

#### 1. はじめに

ソフトウェア・システムに対する顧客要求の高度化が進む中で、近年、開発プロジェクトの大規模化と SW 製品の品質改善への対応が重要な課題となっている。ソフトウェア開発現場では、これらの課題の解決策として、プロセス成熟度モデル [1] に基づくソフトウェアプロセスの改善活動や、既存の SW 資産の効果的な運用を目的とした Product Line 型 SW 開発アプローチ [2] の採用が進められている。

我々は、こうしたソフトウェア開発現場の課題への対応策のひとつとして、ソフトウェア開発環境 EvoMan (Software Evolution Manager) の開発を進めている [3][4]。EvoMan は、チームでのソフトウェア開発を支援対象とする、ソフトウェア構成管

理(以下 SCM)機能を核とした Web 環境上で運用するソフトウェア開発支援環境である。現在、EvoMan に各種ソフトウェア開発支援ツールを連携させる機構を整備、運用を開始している。

本報告では、Product Line 型 SW 開発の支援環境に必要な機能と、環境構築のために整備した EvoMan のツール連携機能を紹介する。

#### 2. 狙い

##### 2.1. Product Line 型 SW 開発

ソフトウェア開発現場では、ソフトウェアの規模が増加する中で、SW 製品に対する幅広い顧客要求に応えつつ、SW 開発効率を向上する方法として、ソースコードをはじめとする SW 資産の流用を広く行っている。既に稼働している SW 資産に対して顧客要求に併

せた修正を加えることで、SWの品質を維持しつつ市場投入時間（Time-To-Market）を短縮することが狙いである。

しかし、従来の流用を主体とするSW開発作業では、改修作業の多くの部分がSW開発者個人の活動に委ねられがちになる等、頻繁に繰り返される修正作業の中で本来の設計の意図や個々の修正の目的等の設計情報が記録され無い等の問題が起きていた。この結果、個々の改修作業に大きな作業時間を費やしたり、SW製品の品質上の問題を引き起こす等の問題に繋がることも多かった。

これらの課題への対応策として、近年Product Line型SW開発[2]のアプローチが提唱されている。Product Line型SW開発では、SW製品系列を早い段階から意識することで、既存のSW資産の効果的な再利用を実現するものである。SW資産の維持管理、プロジェクト管理、製品開発の3つの手法を組み合わせて組織的にSWの再利用・流用を進める点が特徴である。既に様々な手法・事例の報告もあり、これらは概ね次の2つのアプローチに分類できる。

- top-downで製品系列を構築する方法  
FAST手法[6]等に見られる捉え方で、ソフトウェア要求定義の段階からSW製品の将来の機能拡張の可能性を検討し、SW製品開発につなげていく。新規にSW製品系列を構築する際に有効な手法である。
- bottom-upで既存SW資産を成長させる方法  
既存のSW資産から再利用可能なSW資産を抽出・体系化し、このSW資産を利用してSW製品の継続的な開発を進めるアプローチで、たとえば[7]がある。特に、膨大なSW資産が存在する状況でSW流用の効率化を図る手段として有効である。

## 2.2. ソフトウェア開発支援環境への要求

我々は、既に流用対象となるSW資産を多く保持しているSW開発チームは非常に多いと判断している。また、流用を主体としたSW開発でも開発プロジェクトの大規模化は進んでおり、開発チームとして系統的なソフトウェア再利用手法を導入することが重要と考えている。このため、Bottom-upでのproduct lineの構築の支援を検討している。

開発支援ツール側からこうした開発現場の状況に対応する方法は大きく2つある。ひとつは個々のSW開発者を対象にSW製品開発作業を支援する機能の提供であり、もうひとつは開発チーム全体に対するプロジェクト管理の支援機能を提供することである。

前者に対する開発支援機能としては、次の機能がある。

- 既存SW資産の解析・理解への支援  
現実のSW流用では、既に設計情報が消失していた

り、設計仕様書とソースコードが一致していないことも多く、既存のSW資産の理解に多くの時間が費やされている。この作業の効率化のため、例えばソースコードのリバース解析ツール等、既存のSW資産に対する解析支援は重要となる。

- SW資産の維持・変更管理の支援  
長期間かけて蓄積されたSW資産について、基本となるソフトウェア・アーキテクチャを維持し、当初の設計方針を崩す修正を抑制するため、変更管理の支援が重要となる。具体的には、ソフトウェア構成管理[5]の確実な運用を実現することが必要である。
- 変更成果の品質保証に対する支援  
SW資産に対する修正作業の中で、不用意な品質劣化を起こさせないように管理することも重要である。例えば、lint等のソースコードに対する静的解析ツールを用いた評価を一定のタイミングで必ず行う、不具合に対する修正作業を確実にを行う、等の対応が必要となる。  
一方、プロジェクト管理の観点からは、以下の機能が重要となる。

- SW開発チーム内での情報共有の促進  
開発チーム内で常に、SW資産に対する修正状況やその成果を共有出来る機能が重要である。特に、SW資産に対する改修作業が継続する場合、改修作業のベースラインとなるSW資産を常に明確にすること、個別に行われた改修作業の統合作業がSW資産に安全に組み込まれること等も支援すべき活動となる。
- 開発プロセスの安定的な運用に対する支援  
複数のSW開発者が開発チームの抱えるSW資産の状況や個別の修正作業の意図を正しく記録・共有し、お互いの作業調整を行えるようにするには、開発チームで作業プロセスを定義し、これを運用することも必須となる。開発チームが採用するソフトウェアプロセスを個々のSW開発者が確実に実施するための機構が必要となる。  
さらに、SW資産として要求仕様や設計仕様も管理されればば、これらの仕様書への修正作業の管理や仕様書とソースコードの対応関係（トレーサビリティ）もまた管理対象となる。

## 2.3. ソフトウェア開発支援環境 EvoMan

我々が開発を進めているSW開発環境EvoMan[3][4]では、従来からSW資産の共有化とソフトウェア開発プロセスの安定化を狙った機能を提供している。ここでは、既存のEvoManの機能を概説するとともに、Product Line型SW開発への適用にあたっての課題を示す。

### (1) EvoManの機能の狙い

開発チーム内でのSW資産の共有化を促進するため、

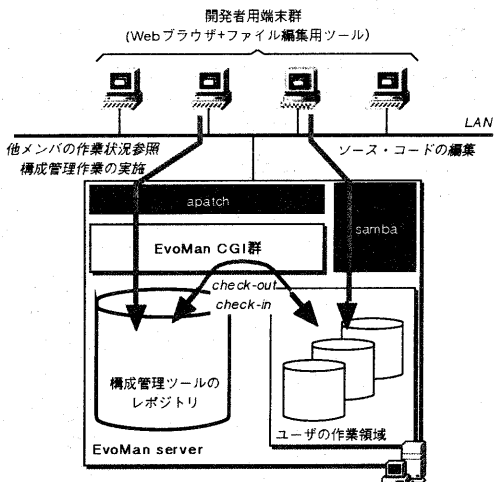


図 1 EvoMan のソフトウェア構成

EvoMan では SCM 活動を支援する機能を開発している。各 SW 開発者に委ねられる SW 改修作業に対し

- 開発チームとして、改修作業のベースラインを常に明確にすること
- SW 資産に対する個々の修正に対して目的、意図を必ず記録する
- 短い周期で改修結果を開発チームの共有 SW 資産として再公開する

ことで、改修成果と改修作業状況をチーム内で共有することを狙っている。この手段として、EvoMan では構成管理ツールを SW レポジトリとして採用し、これに Web 操作インタフェースをあたえて操作性を改善する方法を採った。SW 資産に対する修正作業は常に開発チームの中で共有化される。

一方、ソフトウェアプロセスの安定化としては、特に SW 資産に対する変更管理プロセスが必ず守られるように、操作インタフェースが開発作業者をガイドする構成としている。具体的には、SW 資産を改修する場合、必ず改修の目的、意図、変更内容を記録する手順を埋め込んでいる。こうした記録を残すことで、非同期型ではあるが個々の修正作業の評価や SW 資産の変更管理を可能としている。また、SW 資産が SW レポジトリに集約されるため、SW 資産の改修規模や改修作業時間等の計測も可能となる。これらの計測値は、既存 SW 資産を安定している SW 資産から改修が多い SW 資産へと分離したり、ソフトウェア改修作業に対するプロセス性能を評価することに利用する。

## (2) EvoMan の実装

EvoMan の実装は、市販の構成管理ツールに Web

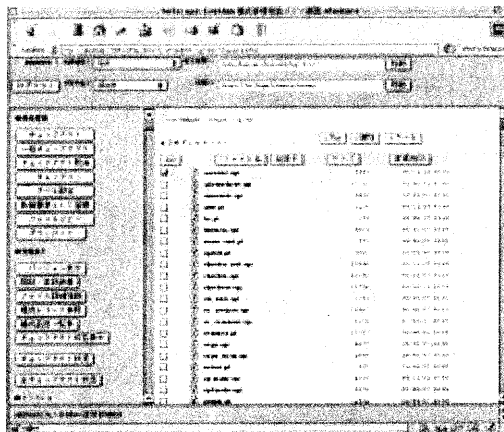


図 2 EvoMan の構成管理画面例

操作インタフェースを提供する形で行われている。EvoMan のソフトウェア構成を図 1 に示す。また、Web インタフェースの操作画面例を図 2 に示す。

なお、個々の開発作業者が実際に修正作業を行う環境は、図 1 に示すユーザの作業領域内で行う。EvoMan で管理された SW 資産を変更するために check-out 操作を行うと、対象ファイルが構成管理レポジトリからユーザ作業領域に移され、この SW 資産はユーザが占有する構成となっている。また、実際の運用では、ユーザの作業領域と各ユーザの操作端末を繋ぐために samba 等のファイル共有機構を利用している。

## (3) Product Line 型 SW 開発の支援に向けた課題

Product Line 型 SW 開発環境として EvoMan を捉えた場合、従来 EvoMan で提供して来た機能は、SW 資産の変更管理、構成管理に限定している。既存の SW 資産の解析や、SW 資産や修正結果に対する品質管理等活動は、ソフトウェア部品庫の機能の外としている。

しかし EvoMan を実開発現場で運用し、SW 資産の一元管理が進むと、ソフトウェア部品庫の機能として SW 資産に対する品質管理の支援との統合が求められるようになった。SW 開発現場での改修作業の手順は、既存の SW 資産の理解、改修、そして品質保証が、一連の開発プロセスとして運用され、相互に密接な関係付けを持っているためである。

このため、我々は以下の方針で、EvoMan の機能強化を進めることとした。

- 既存の SW 資産に対する理解支援や SW 品質保証については、個別に適した CASE ツールがあるため、これを活用することとする。
- 但し、採用するツールは、個々のソフトウェア開発現場で扱う SW 製品や資産の特性、開発組織

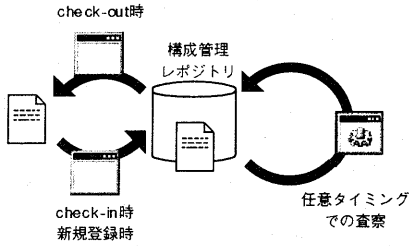


図 3 外部ツール起動のタイミング

の標準開発プロセスに応じて改訂可能とする。

- SW 資産の解析や品質保証等の支援機能を組み込むにあたり、各 SW 開発現場のソフトウェアプロセスに従ってこれらの機能と構成管理機能との組み合わせ方を決定する。
- 通常、構成管理ツールを利用した仕様書等のドキュメント管理は開発現場で広く行われてはならず、このため現場で普及している既存ソースコードの流用・改修を当初の支援をターゲットとする。

以下では、EvoMan を核にした、これらの支援機能の統合の実現方法について示す。

### 3. EvoMan におけるツール連携

ソフトウェア開発環境を構築する際の、ツール連携の方法については、大きく 2 つの観点からの議論が必要である。ひとつは、ソフトウェアプロセスの中での個々の CASE ツールの位置づけを定義する作業であり、もうひとつは物理的な実装方法という観点である。ここでは、この両者の側面から EvoMan におけるツール連携の方式を示す。

#### 3.1. SW 開発プロセスとツール連携

SW 開発プロセスの観点から、構成管理プロセスを軸としたツール連携について 2 つの方式が考えられる。ひとつは詳細作業手順の中での連携であり、もうひとつは作業全体の流れの中での連携である。

##### (1) 詳細作業手順の中でのツール連携

SCM ツールを開発チームで共有して運用する場合、構成管理ツール以外のツールとの連携を行うタイミングは、ひとつは、変更作業の開始時や変更作業の終了時、具体的には SW 資産を SCM ツールから check-out したり、check-in するタイミングである。図 3 にこのタイミングを示す。

ここで示したタイミングで利用するツールは、フィルタとして運用する単機能ツールを組み合わせること

不具合報告

改修

共有化・リリース

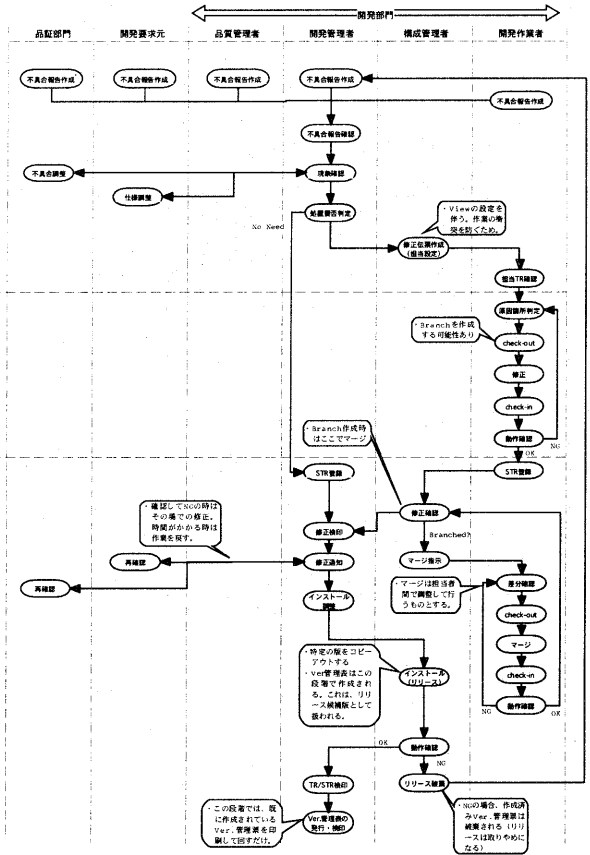


図 4 不具合改修作業のプロセス

を想定している。また、構成管理画面上でのユーザ操作の中に組み込まれて利用されるツールであるため、その処理結果は即座に得られるものが望ましい。

こうしたツールの組み合わせ例としては、改修作業成果を SW 資産としてソフトウェア部品庫に登録する際に、チームの共有資産として品質要件を備えているかをチェックするという使い方ができる。例えば、lint 等の既存ソースコードの静的解析ツールや、単体試験の自動実行ツールの実行による品質確認はその代表的な使い方になる。

##### (2) 全体の作業手順の中でのツール連携

もうひとつの連携は、構成管理作業手順と、その周囲のその他の作業手順を統合する方法である。例えば、図 4 はソフトウェアに対する不具合改修を行う作業手順を示している。ここでは、不具合の報告とそれへの対応という不具合対応プロセスと、変更管理・構成管理のプロセスが組みあわされている。図 4 中、「不

具合報告」のブロックは、不具合が発見された時にその現象が SW 開発チームに不具合報告として示される迄のプロセスである。これに引き続いて行われる「改修」作業は変更管理の作業手順を示す。最後に行われる「共有化・リリース」作業は、改修結果を集積して新しい SW 資産のベースラインを構成し、SW 製品としてリリースするまでの構成管理プロセスと、不具合報告に対する改修レポートを作成し提供する迄のプロセスが並行して行われる。通常こうした作業プロセスは、SW 開発チームが定めている開発規約として規定される。

一般に、このような長期に渡る作業プロセスに基づいたツール連携を実現する場合、作業員対応での作業支援ツールが既に個別に運用されてことが多い。例えば、図 4 のプロセスでは、改修作業からリリース作業迄の構成管理者と開発作業員の活動は構成管理ツールで支援される作業手順である。同時に、開発要求元等との不具合の報告と改修結果の報告は、これとは独立した不具合情報管理ツールで支援されている現場は多く見受けられる。

### 3.2. ツール連携の実装方式

ツール連携を実装する観点では、連携対象となる SW 開発支援ツールの動作環境等の特性があげられる。特に EvoMan は Web 環境で運用するツールである。この EvoMan と連携して運用するツールをネットワーク環境上でどのように配置するかは、開発環境の使いやすさや長期的な運用管理に直接影響を与える項目である。EvoMan のツール連携の実装では、この点を考慮して次の 2 つの方法を用意している。

#### (1) ソフトウェア部品庫サーバ上でのツール連携

ツール連携の実装のひとつの方法は、ソフトウェア部品庫サーバと、連携して運用する各種 CASE ツールを単一のサーバ上に配置する方法である。例えば、単体で動作する小規模なツールを、ソフトウェア部品庫サーバ上で動作させれば、新たな機能を手軽に追加することが可能である。

EvoMan では、この方式でのツール連携を実現するために、ツール起動の標準インタフェースを提供している。このインタフェースでは、EvoMan で管理される特定バージョンの SW 資産を連携させるツールに渡すことことも可能である。EvoMan と組み合わせて既存の CASE ツールを利用する場合は、組み込む CASE ツールのインタフェースと EvoMan のツール起動インタフェースの整合を取るために、簡単なスクリプト (Wrapping script) を作成し、EvoMan の特定ディレクトリに登録して置く。この Wrapping Script は、通常の shell script や perl script として

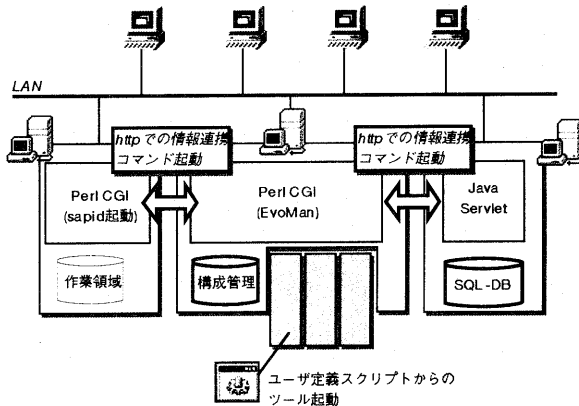


図 5 EvoMan を核としたツール連携機構

用意すれば良い。EvoMan を設置・運用する開発チーム毎に、開発チームの開発規約に沿って独自の拡張を行うことを可能としている。

#### (2) 独立した Web サーバを立ち上げて Web アプリケーション連携として動作させる連携

もうひとつの連携方式は、連携先のツールを Web 環境上で稼働可能な形態とし、Web アプリケーション連携の機構を利用してツールの統合化を図る方法である。例えば、大規模なソフトウェア開発プロジェクトでは、プロジェクト内での情報共有を目的に、既に Web 環境上で動作するグループウェア型の SW 開発管理ツールを運用していることが多い。不具合情報管理のツールは、こうしたツールの一例である。

このようなツールとの連携では、http プロトコルを使った Web アプリケーション連携の機構を利用してツール連携を実現する。連携対象ツールが提供する個別の機能が独立したサービスとして公開されることによって、開発チームで運用されるソフトウェアプロセスに従ってこれらの個別サービスを順次呼び出す方法が可能となる。

また、この他にも、

- ソースコードのリバースツールのように、ツールの実行負荷が高く on-line 型の運用が難しい場合
- 連携したい CASE ツールがソフトウェア部品庫サーバとは異なる環境でしか動作出来ない場合等も、これらのツールを起動するための Web インタフェースを用意し、ソフトウェア部品庫側から起動することが可能である。

### 3.3. 実現例

以上示した検討に基づき、EvoMan では図 5 に示す構成で、SW 品質保証の機能や SW 資産に対する理解

支援の機能を用意している。

具体的には、SW品質保証の目的では、

- 構成管理レポジトリへの登録時に lint や市販の静的解析ツールを通し、ソースコードの評価を必ず実施する
- 不具合修正作業の管理を目的として、開発チーム内で開発・運用している不具合情報管理ツールとの連携動作を行う

というツール連携を行っている。

一方、既存ソフトウェア資産の理解支援としては、現在 sapid[8]環境に含まれる spie と EvoMan の組み合わせでの運用を開始している。なお、sapid ツール群と EvoMan の連携では、対象とする SW の規模が大きく spie の出力を得るのに長時間かかることから、ネットワーク環境上に独立したサーバを用意し、http を使ったツール連携の機構を利用して運用している。このため、sapid による既存 SW 資産の解析は、ソフトウェア部品庫上でリリース作業を行い新しいベースラインを確立した段階で実施することとしている。

#### 4. 議論

構成管理ツールと不具合情報管理ツールの連携の例としては、bonsai 等が既に存在する。また両者の連携による SW プロセスの性能計測も報告がある[9]。今回の我々の報告では、特にツール連携を

- 開発プロセスを中心に据えること
- 開発プロセスの変化に対応したカスタマイズ性を考慮すること

で、プロセス計測の適用範囲を拡げるものと判断している。また、我々自身が今後、不具合情報管理機能と構成管理機能の連携ツールを運用する中で SW 開発チームのプロセス性能を計測し、その結果をもとに開発プロセスの改善を進めて行くことを計画しており、今回の開発環境の構築はこうした活動をするための基盤を整備したものと捉えている。

Product Line 型 SW 開発の支援環境としては、特にソースプログラムを対象とした支援環境の整備を済ませた段階にある。今後の課題として、ソフトウェア・アーキテクチャを核にした SW 開発プロセスの確立が重要と考えている。具体的には、以下の課題への対応が必要と判断している。

- ソフトウェア・アーキテクチャの仕様記述モデルの定義
- ソフトウェア・アーキテクチャの仕様情報とソースコードとのトレーサビリティを管理する機構の整備
- SW 資産の変更頻度や不具合発生状況等を追跡・計測した結果からのソフトウェア・アーキテクチャ見直しのための具体的な評価基準の確立

最後に、ソフトウェア開発プロセスを基礎とするソフトウェア開発環境としては、Web 環境上のアプリケーション連携の機構が利用出来ることを確認したと捉えている。ただ、現状では、個別のツール連携からこれらの連携機構が開発されており、SW 開発作業支援のための Web サービスという観点で整理しきれていない状況にある。今後、ツール連携を繰り返していく中で、この点の改善は進めて行く必要がある。

#### 5. まとめ

本稿では、Product Line 型 SW 開発の支援環境として、Web 環境上で運用する開発環境 EvoMan を取り上げ、開発プロセスに応じた開発環境の機能拡張の方法について示した。ソフトウェアプロセスを軸にしたツール連携のあり方を検討し、その検討結果に基づいてツール連携機能を開発した。

今回示したツール連携は、Web 環境上の SW 開発環境を長期的に運用し、開発組織の変化に応じて機能拡張をする手段を提供するものと考えており、今後これを利用した支援環境の高度化を進めて行く。併せて、Product Line 型 SW 開発支援環境としては、開発プロセスを具体的に定義し、本環境の機能に反映する。

#### 文献

- [1] Paulk, M., et al., *Capability Maturity Model for Software Version 1.1*, CMU Technical Report CMU/SEI-93-TR-025, CMU SEI, Pittsburgh, PA, 1993.
- [2] Clements, P., and Northrop, L.M., *Software Product Lines: Practices and Patterns*, Addison-Wesley Publishing, 2001.
- [3] 佐々木 幹郎 他, “ソフトウェア再利用環境 EvoMan における構成管理機能,” 情報処理学会ソフトウェア工学研究会 研究報告, No.121-12, Mar. 1999.
- [4] 佐々木 幹郎 他, “ソフトウェア開発環境 EvoMan における構成管理機能の評価,” 情報処理学会ソフトウェア工学研究会 研究報告, No.124-11, Oct. 1999.
- [5] Bersoff, E.H., “Elements of Software Configuration Management,” *IEEE Trans. Of Software Engineering*, Vol. SE-10, No.1, pp.79-87, IEEE Computer Society Press, Jan. 1984.
- [6] Weiss, D.M., and Lai, Chi Tau R., *Software Product-Line Engineering: A Family-Based Software Development Process*, Addison-Wesley Publishing, Reading, MA, 1999.
- [7] Brownsword, L. And Clements, P., *A Case Study in Successful Product Line Development*, CMU SEI Technical Report CMU/SEI-96-TR-016, CMU, Pittsburgh, PA, 1996-10.
- [8] 福安 直樹, 山本 晋一郎, 阿草 清滋, “細粒度ソフトウェア・リポジトリに基づいた CASE ツール・プラットフォーム Sapid,” 情報処理学会論文誌, Vol.39, No.6, pp.1990-1998, 1998-06.
- [9] Pooley, R., et al., “Collecting and Analyzing Web-Based Project Metrics,” *IEEE Software*, Vol.19, No.1, pp.52-58, IEEE Computer Society Press, Jan. 2002.