

コンポーネント指向のプロセスモデルに基づく ソフトウェアプロセス測定・分析支援手法の提案

岡井 洋樹[†] 飯田 元^{††} 松本 健一[†] 田中 康^{†‡}

本論文は、ソフトウェア開発組織の成熟度モデルである CMM や CMMI がソフトウェアプロセスに対して要求する様々な測定・分析活動を、コンポーネント指向のプロセスモデルに基づいて支援する仕組みを提案する。具体的には、まず、CMM・CMMI に基づいたベース尺度と導出尺度の記述をプロセスコンポーネントに施す事によって、測定・分析担当者の活動を確実に行わせる仕組みを提案する。次に、プロジェクト個々のソフトウェアプロセスを作成・変更する際に、測定データと分析活動の間の対応がとれているかどうかの整合性を検証する方法を提案する。これらの仕組みを用いることで、測定・分析活動におけるソフトウェア開発プロセスの品質と信頼性の向上が期待される。

Representation Method of Measurement and Analysis Activities in Component-Based Software Process Model

Hiroki Okai[†], Hajimu Iida^{††}, Ken-ichi Matsumoto[†] and Yasushi Tanaka^{†‡}

This paper proposes a support mechanism for various measurement and analysis activities demanded from CMM / CMMI software development organization maturity models based on a component-oriented process model. At first, The mechanism for measurement of measurement and analysis activities by specifying employment of base measure and derived measure, demanded by CMM/CMMI, to a process component. Next, the method of verifying the correspondence between measurement data and analysis activities when the project defined software process is created and changed is proposed. By using these mechanisms and methods, the improvement of quality and reliability of software development process in measurement / analysis activities is expected.

1. はじめに

近年、ソフトウェアは社会のあらゆる場面で使用されており、その適用範囲は拡大の一途をたどっている。その結果、ソフトウェアの規模は増大し、またソフトウェアの欠陥が社会に与える影響も増大し続けている。

ソフトウェアの生産性と品質の双方に大きく影響するのがソフトウェアを開発する工程、いわゆるソフトウェアプロセスである。ソフトウェアプロセスを適切に管理・運用することは、生産性と品質の維持に不可欠で、その改善は、ソフトウェアを開発する組織にとって非常に重要な技術的要件である。CMM (Capability Maturity Model)[1] や CMMI (Capability Maturity Model Integration) [2] は、組織のソフトウェア開発能力をプロセスマネジメントの観点から評価・改善する枠組みで、近年多くの組織による利用が進んでいる。

一方、成果物やプロセスに対する定量的な尺度を用いた計測・分析は、ソフトウェアの品質向上のための重要な技

術として、数多くの研究がおこなわれてきた[3]。計測・分析作業を各開発工程で適切に実行することは、ソフトウェアプロセスの観点からも重要な課題である。CMM や CMMI においても、計測と分析は重要なプロセス分野のひとつとして扱われてはいるが、具体的な支援の仕組みを与えるものではない。

本論文では、ソフトウェアプロセスをコンポーネント化したプロセスコンポーネントを用いて、CMM や CMMI に基づいた測定・分析活動の支援手法について提案を行なう。

本論文で提案する手法は、著者らの研究グループで提案を行っている、コンポーネント指向のプロセスモデル[4][5] に基づいたものである。このプロセスモデルでは、あらかじめ用意されたコンポーネントを選択して組み合わせることでプロセスを策定可能にするものである。

本研究では、まず計測と分析の活動を予めコンポーネント内に記述しておく方法を定めた。計測と分析に関連する活動をその他の開発作業間に織り込んでプロセスを構成することは、煩雑な作業であったが、提案方法を用いることで、容易に織り込むことが可能となる。また、過去のプロセス記述より計測データ等が容易に参照可能となるため、過去のプロセス資産の再利用促進にも寄与すると期待される。

[†] 奈良先端科学技術大学院大学情報科学研究科 Graduate School of Information Science, Nara Institute of Science and Technology
^{††} 奈良先端科学技術大学院大学情報科学センター Information Technology Center, Nara Institute of Science and Technology
[‡] ソニー(株) ネットワーク&ソフトウェアテクノロジーセンター Sony Corporation, Network & Software Technology Center

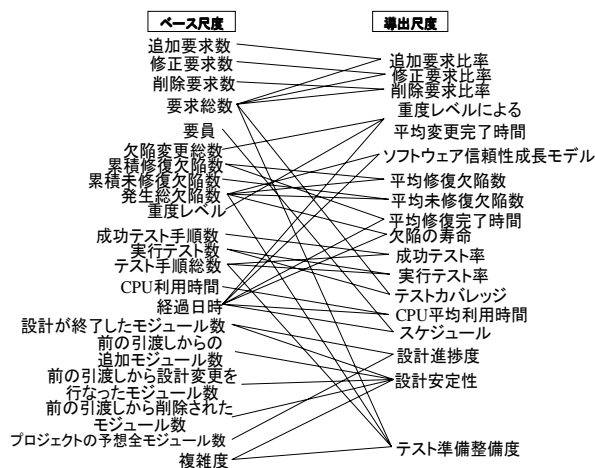


図1 ベース尺度と導出尺度の関係

次に、計測データの生成と利用の間の対応関係がコンポーネントのインターフェースによる規定を充足し、かつ、上流工程での計測活動と下流工程での計測データ利用との対応がプロセス全体でとれているかどうかを検証する方法を提案した。

以降 2 章では本論文において前提条件となる事項を準備として説明を行う。次に 3 章ではプロセスコンポーネントを用いた測定・分析手法について提案を行なう。更に 4 章では、提案内容が CMMI のプロセス要件をどの程度充足可能かについて考察を行い、最後に 5 章ではまとめとして現状と今後の課題について記述する。

2. 準備

2.1 ソフトウェアメトリクス

ソフトウェアメトリクスとは、一般的に、ソフトウェアプロダクトの様々な特性(複雑度, 信頼性, 効率など)を判別する客観的な数学的尺度である。ソフトウェアプロダクトの品質や生産性を評価し、ソフトウェアプロセスの改善を行う為に様々な種類のソフトウェアメトリクスが提案され、実際の開発現場において使用されている[6]。

ソフトウェアメトリクスは理解の仕方や利用目的によって様々な形に分類される。例えばソフトウェアメトリクスの対象物を基準に注目した場合、最終プロダクトや中間プロダクトの特性を評価する為の「プロダクトメトリクス」、変換作業における資源の消費量や消費される資源の特性を評価する為の「資源メトリクス」、個々の変換作業や開発作業全体の特性を評価する為の「プロセスメトリクス」に大分する方法がある[3]

一方、今回本論文で対象とする CMM / CMMI では、直接の測定によりデータが得られる「ベース尺度」と、ベース尺度を複数組み合わせることでデータが得られる「導出尺度」の 2 つにメトリクスを大分している。ベース尺度とは、測定活動に関するソフトウェアメトリクスを指す。例えば LOC (Line Of Code) や実施テスト数などがベース尺度に当たる。一

方、導出尺度とは、測定されたデータを組み合わせる事により得られるソフトウェアメトリクスである。導出尺度の例としては、テスト率や設計進捗度などが挙げられる。図1にベース尺度と導出尺度の関係の例を示す。

ソフトウェアメトリクスを有効に活用する一つ的手段として、ベース尺度と導出尺度の定義を網羅的に用意し、それらの間の関係をメタデータとして予め記述しておく、測定と分析の活動を含むソフトウェアプロセスを作成する際の手引きとする事が考えられる。ただし、ベース尺度と導出尺度の定義は、組織の環境や文化に応じて様々であると考えられるため、必要に応じて、尺度の定義を柔軟に追加・変更できるような枠組みが同時に必要である。

2.2 CMM / CMMI

CMM (Capability Maturity Model) 及び CMMI (Capability Maturity Model Integration)は、米国のカーネギーメロン大学のソフトウェアエンジニアリング研究所 (SEI = Software Engineering Institute)によって作成された組織の開発能力改善のためのフレームワークである。CMM 及び CMMI は開発プロセスに視点を置いた組織能力の改善を目的としており、組織の持つプロセス実効能力を成熟度にあわせて改善していく事ができるモデルである。CMM や CMMI の段階表現では組織の能力は 1 から 5 の成熟度レベルで表される。組織は、まず、現在の成熟度レベルを評価した上で、更に上位のレベルの認定を目指してプロセス改善活動を行う。

CMM では、成熟度レベル毎に満たすべき項目として複数の主要実践分野 (KPA=Key Practice Area) が定められており、それぞれの KPA には 5 つのコンフィーチャが用意されている。コンフィーチャとは、KPA のゴールを満たす為の属性として定義され、KPA の実装・制度化が効果的で反復・永続できるかを示す。測定と分析の活動は、全ての KPA に関与するコンフィーチャの 1 つとして記述されている。

一方 CMMI は CMM から派生した改善フレームワークを CMM Version2 を基に再統合・拡張したものである。CMMI では主要実践分野 (KPA) という用語はプロセス分野 (PA=Process Area) へと変更された。また、CMM においてコンフィーチャとされていた測定と分析は成熟度レベル 2 の PA として定義されており、改善の初期段階において、測定と分析の活動を組織的に確立することが要求されるようになった。

このように CMM や CMMI を基にしたプロセス改善活動においては、測定と分析が非常に重視されているが、実践の具体的な内容については具体的な指針を与えていないわけではない。本論文では CMMI の PA である「測定と分析」に焦点を当て、プロセスコンポーネントを用いた測定・分析活動の支援手法を提案する。

2.3 プロセスコンポーネントモデル

我々の研究グループではソフトウェア開発プロセスを部分タスク毎にコンポーネント化し、後の開発において再利用できる仕組みを持つプロセスモデルを提案している[4].

プロセスコンポーネントモデルでは、ソフトウェアプロセスは、要求分析や、ソフトウェア設計、コーディングといった、プロセス要素の集合で構成される。各プロセス要素は様々な成果物(ワークプロダクト)を入出力として持ち、その入出力仕様をタスクと呼ぶ。タスクで規定される仕様を実装したものがプロセスコンポーネントである。仕様が同一であれば、実装(つまり細かい作業内容)が異なるコンポーネントに差替えを行なえるようにすることで、プロセス全体のカスタマイズを容易に行えることが本モデルの大きな特長である。なお、以下の議論においては、プロセス要素、タスク、コンポーネントを区別して扱う必要がないので、主にタスクという用語を用いる。

各タスクは図2に示すように6種類のデータの入出力のインターフェースを持っている。各入出力データの詳細は以下のとおりである。

- ・ 主入力/主出力 (Source Input/Output) … タスクの開始・終了条件としての作業成果物。PDSP におけるタスクの実行順序は、主入出力の依存関係によって基本的な制約を受ける。
- ・ 計測入力/計測出力 (Measurement Input/Output) … タスク内の測定・分析活動において利用される作業成果物。
- ・ 指示/報告 (Instruction Input/Output) … タスクの実行に際しての作業指示/報告。

本プロセスモデルは、プロジェクト管理者が「プロジェクト定義ソフトウェアプロセス」(PDSP=Project Defined Software Process)を構成する際に役立てることを想定している。プロジェクト管理者は、組織のプロセス資産として提供されるプロセスコンポーネントを組み合わせる事で容易にPDSPを構成することができる[3][9]。より詳細なスケジュール等を含んだプロジェクト計画がPDSPを元に策定され、これらを基にソフトウェア開発プロジェクトが実行される。プロジェクト完了後は、PDSP 自体や測定データなどの様々な情報を組織のプロセス資産として保存し、再利用に役立てられる。

本モデルは、PDSP における全体的な測定・分析活動を各タスクで行われる測定・分析作業の結果得られる測定データのやりとり(計測入出力)として実現可能なように設計されているが、その第1バージョン[4]においては、具体的な運用の方針は未策定であった。本研究は、このモデル上でのソフトウェアメトリクスの扱いを詳細に定め、それを利用した支援手法を提案するものである。

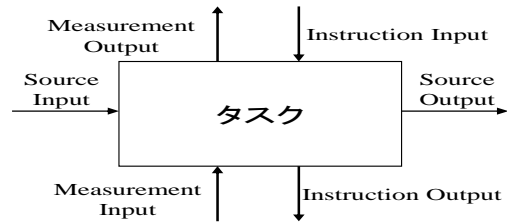


図2 プロセスコンポーネントの構造

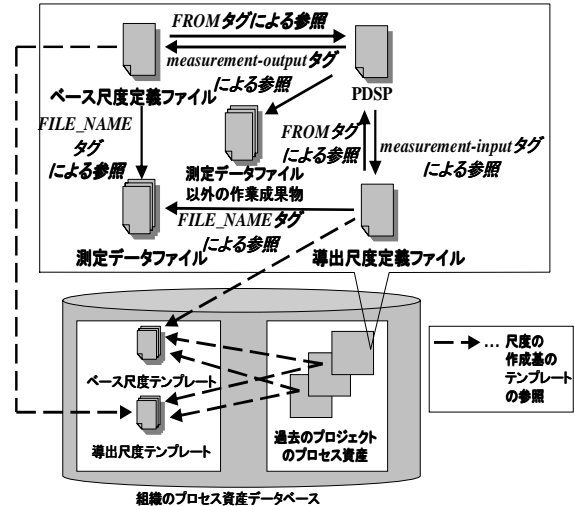


図3 プロセス資産における測定・分析関連データの配置

表1 ベース尺度定義ファイルの記述内容

要素名	意味
METRICS_BASE_DEFINITION_NUMBER	ベース尺度のテンプレート番号
METRICS_BASE_DEFINITION_NAME	ベース尺度の名前
FROM	参照されるプロセスコンポーネントファイル名
FILE_NAME	ベース尺度のデータファイル名
WORKER	ベース尺度の測定・分析担当者名
HOW_TO_MEASURE	ベース尺度の測定方法
DATA_MODEL	ベース尺度のデータの型
CALCULATE_INTERVAL	ベース尺度の測定間隔
MEASUREMENT_TOOL	ベース尺度の測定に使用するツール
COMMENT	ベース尺度についてのコメント

表2 導出尺度定義ファイルの記述内容

要素名	意味
METRICS_DERIVED_DEFINITION_NUMBER	導出尺度のテンプレート番号
METRICS_DERIVED_DEFINITION_NAME	導出尺度の名前
FROM	参照されるプロセスコンポーネント名
METRICS_BASE_DEFINITION_NUMBER	参照するベース尺度の番号
METRICS_BASE_DEFINITION_NAME	参照するベース尺度の名前
FILE_NAME	参照するベース尺度のファイル名
WORKER	導出尺度の測定・分析担当者
HOW_TO_CALCULATE	導出尺度の計算方法
DATA_MODEL	導出尺度のデータの型
ANALYSIS_TOOL	導出尺度の分析に使用するツール
COMMENT	導出尺度に関するコメント

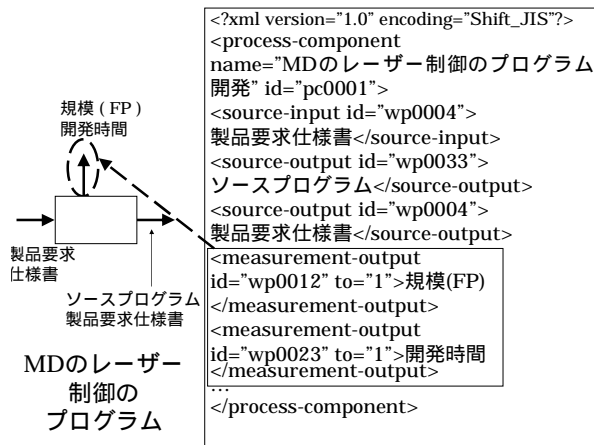


図4 測定活動を持つタスクの概念図と記述

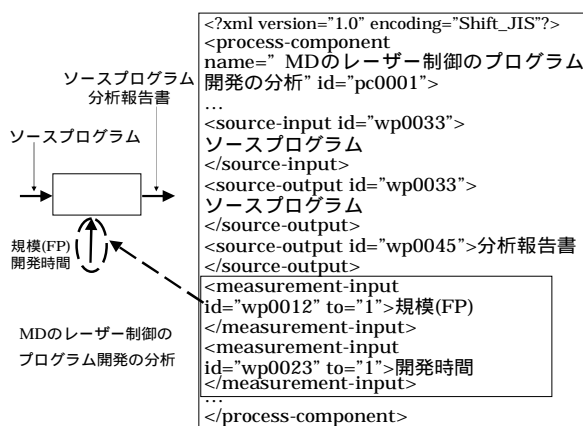


図5 分析活動を持つタスクの概念図と記述

3. 測定・分析支援手法の提案

2.3 のプロセスモデルを基にしたソフトウェア開発では、図3に示されるように、プロジェクト毎に、PDSP やプロジェクト計画書、プロセスコンポーネントが記述されているファイル群、ベース尺度定義ファイルと導出尺度定義ファイルの組、ベース尺度と導出尺度のテンプレート、ベース尺度による測定データファイル群とプロセスコンポーネントの主力(Source Output)と報告(Instruction Output)で得られる作業成果物が作成される。

測定と分析の活動は PDSP、ベース尺度定義ファイルと導出尺度定義ファイル、ベース尺度による測定データファイルを基に行なわれる。これらのファイルは XML で記述される。ベース尺度定義ファイルと導出尺度定義ファイルはプロジェクト毎に一組存在し、各プロジェクトで定義されるベース尺度や導出尺度の全てがそれぞれの定義ファイルに記述されている。これらの資産を利用する事で、プロセスコンポーネントの再利用が可能になる。

3.1 ベース尺度・導出尺度定義ファイル

ベース尺度定義ファイルや導出尺度定義ファイルはテン

プレートを基に作成される。それぞれのテンプレートには組織全体で定義されるベース尺度・導出尺度の仕様が記述されている。

ベース尺度定義ファイルでは測定活動に関する情報として表1に示されるメタデータ(ベース尺度の測定を行なう責任者や測定方法、使用するツール、コメントなど)が記述される。要素名であるFROMとFILE_NAMEは、他の要素ファイルへのリンク情報を格納しており、リンクを辿る事で測定活動を行なうプロセスコンポーネントや測定データを格納するデータファイルにアクセスする事が可能となる。ベース尺度の測定方法はHOW_TO_MEASUREに記述され、自然言語によって測定方法の内容が記される。

導出尺度定義ファイルも同様に表2のようなメタデータを用いて分析活動に関する情報が記述されている。測定活動と異なりベース尺度を組み合わせて分析する作業が必要である為、計算方法の項目がある。計算方法とは参照する測定データから導出尺度データを得る演算を意味し、自然言語で記述される。具体的な算出の機構は、組織における導出尺度の算出方法に委ねられる。本手法では、分析手順の自動化を必須要件とせず、分析手順の記述に自然語を含む任意の言語を用いることを許容する。ただし、Java アプレットなどの外部ツールへの参照リンクを記述することで、自動化についても考慮する。

いずれの尺度においても、基本的な仕様として、自身のデータ型についての記述(真値値・整数値・小数値・ランク値などの指定と、連続値・非連続値、の別の指定)を持ちDATA_MODEL タグを用いて記述する。導出尺度から参照されるベース尺度のデータ型は導出尺度定義ファイルのメタデータとして記述されている。

3.2 ベース尺度・導出尺度による測定・分析支援

測定や分析の活動は PDSP 中に定義され、PDSP を基にプロジェクト計画が作成される。その後ベース尺度定義ファイルや導出尺度定義ファイルは PDSP を基に記述される。これらの記述は組織のプロセス担当者によって行なわれる。そしてソフトウェア開発は PDSP とプロジェクト計画を用いて行なわれ、開発が進む中で測定や分析のプロセスコンポーネントに達した時、測定・分析担当者によって活動が行なわれる。

図4と図5は測定と分析のプロセスコンポーネントの概念図と記述を表している。プロセスコンポーネント内の記述における measurement-input/-output が測定・分析活動の判定タグに当たる。開発プロセスの進行中に測定・分析の判定タグが含まれるプロセスコンポーネントに作業が達した場合、図6、図7に示される活動が行なわれる。ここで測定・分析活動を支援する部分を「測定・分析支援部」と呼ぶ事にする。

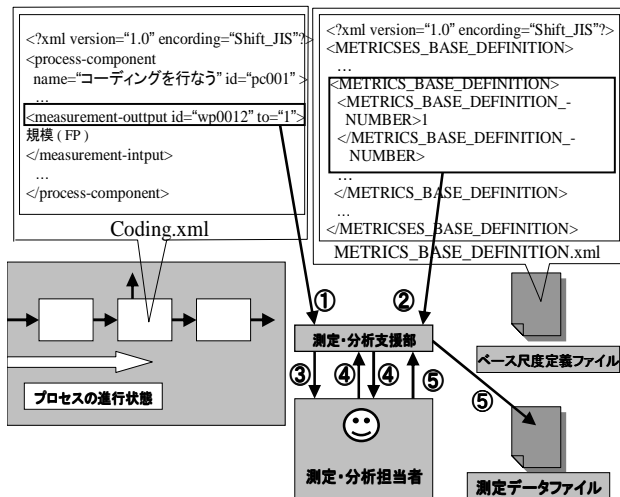


図6 測定活動

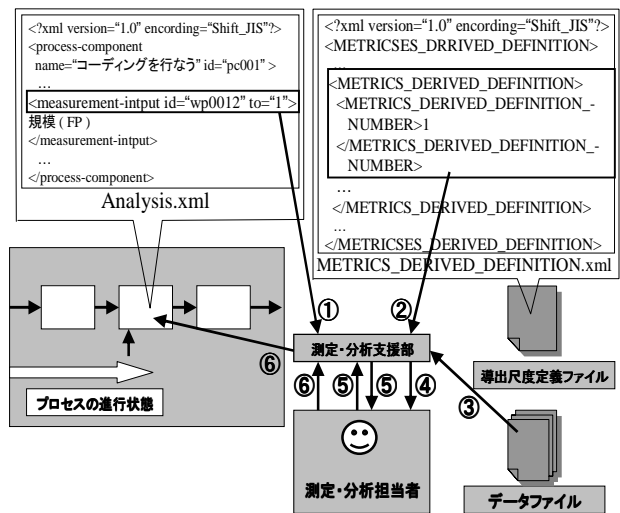


図7 分析活動

3.2.1 測定活動支援の手順

測定活動支援の手順は以下のとおりである。

ソフトウェア開発のプロセスが測定のプロセスコンポーネントに到達すると、測定支援部に対して measurement-output の開始タグの情報が知らされる。測定支援部は measurement-output の開始タグから、属性 to の値を格納しておく。

ベース尺度定義ファイルを読み込み、①の to の属性値と METRICS_BASE_DEFINITION_NUMBER タグの要素の内容とを比較し、情報が合致する METRICS_BASE_DEFINITION_NUMBER タグのデータセット(1つのベース尺度の全ての情報)を格納しておく。

METRICS_BASE_DEFINITION_NUMBER タグでくられる内容にはベース尺度における測定活動に関する情報が含まれており、測定・分析支援部は測定・分析担当者を WORKER の要素の内容から特定し測定活動を通知する。

通知を受けた測定・分析担当者は測定支援部にアクセ

スし、行うべき測定作業の内容を測定・分析支援部が格納しているベース尺度の記述を基に提示される。

測定・分析担当者は提示される情報を基に測定活動を行なう。そして測定されたデータを測定・分析支援部に入力する事で FILE_NAME タグの要素の内容の名前で測定データファイルが組織のプロセス資産として格納される。

3.2.2 分析活動支援の手順

一方分析活動の支援作業の詳細は以下のとおりである。

ソフトウェア開発のプロセスが分析のプロセスコンポーネントに到達すると、測定支援部に対して measurement-input の開始タグの情報が送られる。測定支援部は measurement-input の開始タグから、属性 to の属性値を格納しておく。

導出尺度定義ファイルを読み込み、①の to の属性値と METRICS_DERIVED_DEFINITION_NUMBER タグの要素の内容とを比較し、情報が合致する METRICS_DERIVED_DEFINITION_NUMBER タグのデータセット(1つの導出尺度の全ての情報)を格納しておく。

METRICS_DERIVED_DEFINITION_NUMBER タグでくられる内容には導出尺度における分析活動に関する情報が含まれており、測定・分析支援部は FILE_NAME の要素の内容から参照されるベース尺度を取り出し、HOW_TO_CALCULATE と HOW_TO_ANALYSIS の要素の内容に従ってベース尺度データを導出尺度データに変換しておく。

測定・分析支援部は測定・分析担当者を WORKER の要素の内容から特定し分析活動を通知する。

通知を受けた測定・分析担当者は測定支援部にアクセスし、行うべき分析作業の内容を測定・分析支援部が格納している導出尺度の記述を基に提示される。

測定・分析担当者は提示される情報を基に分析活動を行ない、分析結果を利害関係者に対して通知すると共に分析結果報告書を分析のプロセスコンポーネントの作業成果物として返す。

測定・分析担当者は測定・分析支援部から測定・分析の通知を得る事から測定・分析の時期を意識せずに他の作業が行える。

また測定活動においては測定結果を測定支援部に入力する事で、定められた測定データファイルに測定データが格納され、以降 FILE_NAME タグが測定データファイルへのリンク情報としての役割を果たす。

分析活動において、導出尺度のデータは測定・分析支援部が作成する事から、導出尺度データ作成時の誤りを防ぐ事が可能となる。これは導出尺度の記述が間違いなく行なわれる事が前提である。

このように 測定・分析活動に関して、ベース尺度・導出

尺度定義ファイルのメタデータを測定・分析支援部が解釈する事で、測定・分析担当者は半自動的に測定・分析活動を行なう事が可能となる。

3.3 計測活動と分析活動間の整合性

本節では、「分析活動に必要なデータ（ベース尺度）がどのプロセスコンポーネントでも計測されていない」というような、計測と分析の活動の観点からの矛盾がプロセス記述中に存在しないことを検証する方法を説明する。まずプロセスにおける「計測活動と分析活動間の整合性」（以下では計測整合性と略記）を定義し、次にその検証方法を示す。ただし、プロセス全体を見渡してモニタリングするような、高位のタスクに対する検証は以下の議論の範囲外である。

3.3.1 諸定義

以降の議論を簡潔にするための準備を行う。まず、タスク T の主入出力、計測入出力、指示・報告の表記を、 $T.SI/T.SO$, $T.MI/T.MO$, $T.II/T.IO$ とする。次に、二つのタスク間の関係として、以下の依存関係を定義する：

制御依存関係:「タスク T_1 がタスク T_0 に制御依存する」とは、 T_1 の実行前に T_0 が実行される可能性があることを意味する

成果物依存関係:「タスク T_1 がタスク T_0 に成果物依存する」とは T_1 が T_0 で生成される成果物を参照すること ($\exists w \in T_1.SI, w \in T_0.SO$) を意味する

計測依存関係:「タスク T_1 がタスク T_0 に計測依存する」とは T_1 が T_0 で計測される尺度を参照すること ($\exists w \in T_1.MI, w \in T_0.MO$) を意味する

3.3.2 計測整合性の定義

プロセスコンポーネントを組み合わせる事で $PDSP$ を記述する場合、計測・分析活動を滞りなく運用するには、下流工程における分析活動において必要とされる測定データが、上流工程の計測活動において収集されていなければならない。このように、計測活動において得られるデータの項目が、分析の活動において必要とされるデータの項目を満たしていることが「計測活動と分析活動間での整合性（計測整合性）」であるとする。

計測活動と分析活動間の整合性は、具体的には、単一タスク内での整合性とタスク間での整合性の 2 段階で規定される。

単一タスク内での計測整合性

あるタスク T が内部に分析活動を持つとき、そこで算出される導出尺度 ($T.MO$) が依存するベース尺度がすべて $T.MI$ の要素として宣言されており、かつ $PDSP$ 内で宣言されたワークプロダクトとそれぞれ対応付けされている必要がある。導出尺度がどのベース尺度に依存するかは導出尺度のメタデータ (尺度定義ファイル) にデータタイプと共に記されており、各ベース尺度のデータタイプは導出尺度の

各タスクの MD (上流で計測済みの尺度の集合) を求めるアルゴリズム

```

foreach task t in T{//Tはタスクのリスト;
//成果物依存タスクのリストの初期値は直接依存
するタスク
t.refer = t.si;
//直接成果物依存するタスクの計測出力を MD の
初期値とする
t.md = empty;
foreach t1 in (t.refer){
t.md = t.md + t1.mo;
}
} //初期化終了
do{//計測データの伝播計算
updated = false;
foreach task t in T{
//依存リストが空なら、計算不要
if t.refer is empty next;
//依存リストの先頭のタスクの持つ MD を加える
t1=head(t.refer); t.md += t1.md;
//依存リストの先頭のタスクをその依存リストで
置換
rplace(t.refer, t1, t1.refer);
//依存リストから自分を除去
remove(t.refer, t);
//データ更新フラグを立てる
updated = true;
}
}while(updated) //データ更新が無くなれば完了

```

図8 各タスクの T.MD (上流で計測済みの尺度の集合) を求めるアルゴリズム

メタデータで規定されたものか、そのサブタイプである必要がある。

厳密な意味での単一タスク内での整合性とは以下のとおりである：

あるプロセス P において、型 t の導出尺度を格納するワークプロダクト $w(w:t$ と表記) がタスク T の計測出力 ($T.MO$) の一つに割り当てられており、一方で $T.MI$ にはベース尺度を格納する複数のワークプロダクト $\{w_1:t_1, \dots, w_n:t_n\}$ が割り当てられているとする。さらに、 w の定義ファイル (メタデータ) 中には、 w の値導出に必要なベース尺度の型の集合 $w.B$ が宣言されているとする。

このとき、 T 単体の計測整合性とは、 $\forall w \in T.MO$ について $T.MI \supseteq w.B$ が満たされていることである。

タスク間での計測整合性

まず、単体で整合性が取れているタスク T について、すべての計測入力プロセス P 内の他のタスクの計測出力に関連付けられている必要がある。：

$\forall w \in T.MI$ について、 $\exists T_i \in P, w \in T_i.MO$

さらに、タスク T の内部で実際に導出尺度を計算する前提として、その尺度の元となるベース尺度が他のタスクによってすべて予め計測済みである (つまりそのタスクの実行に必要な計測データが充足されている) 必要がある。

具体的には、タスク T が制御依存するすべてのタスクの計測出力の和集合を $T.MD$ とするとき、 $T.MD \supseteq T.MI$ であ

るなら、T の計測データは充足している。プロセス記述 P 内のすべてのタスクについて計測データが充足されているとき、P の整合性が保たれていることになる。一般には、タスク間の制御依存関係を実行前に正確に把握することは不可能であるため、ここでは、タスク間の制御依存関係を、タスク間の成果物依存関係で近似したもの(タスク T が成果物依存するすべてのタスクの計測出力の和集合を T.MD としたときに $T.MD \supseteq T.MI$)を、タスク間の計測整合性として定義する。ただし、本検証法では、尺度の充足性を満たさないプロセス記述を発見することは可能だが、タスク内の細かい作業順序によっては、実行時のデータ充足性が満たされない可能性も存在する。

3.3.3 計測整合性の検証

単一タスク内での計測性合成は、タスクワークプロダクトのすべての関連と尺度を定義するメタデータが存在する場合、容易に可能であるので、ここでは省略する。タスク間の計測整合性を検証するには、PDSP に含まれるすべてのタスク T に対して、T.MD を求めればよい。T.MD は図 8 のような単純なアルゴリズムで求められる：

4. 考察

CMMI では PA の達成に関して、それぞれの PA に固有のゴールとしての Specific Goal と、他の PA と共有するゴールとしての Generic Goal を満たしている必要がある。それぞれのゴールには満たすべき具体的な作業項目として、Specific Practice と Generic Practice が定義されている。

PA「測定と分析」には 20 の Practice が含まれており、本提案手法で、それらをどれだけ達成可能かを評価した結果を表 3 に示す。ここでは Specific Practice を SP, Generic Practice を GP, と略記し、達成度の判定において特に有効な項目を◎、達成可能な項目を○、部分的に達成可能な項目を△、達成に寄与できない項目を×で示している。

表 3 の結果より本手法が前提とするプロセスコンポーネントモデルのみの利用により、PA「測定と分析」の 20 の Practice の内、達成可能な項目が 7 個、部分的に達成可能な項目が 8 個、達成に寄与できない項目が 4 個であった。これに、本稿で提案した手法を追加した結果、合計で 17 個の項目が、本手法の実践により、達成可能であると判断された。現状で達成が不十分か困難な残り 3 個の Practice は以下のとおりである：

- **SP2.1 測定データ収集**：データの一貫性や、符号者間信頼性、基準妥当性といった、測定データそのものに関する抽象的尺度については現在未対応である。
- **GP2.1 組織方針の確立**：ここでは組織が測定と分析に関する方針を立てて積極的に測定・分析活動を行なわなければならないとされている。計測・分析活動間に対するポリシーを本提案手法に基づいて文書化することは十分に

可能であり、今後の課題として検討を行いたい。

- **GP2.5 トレーニング**：ここでは測定・分析者活動の専門家でないソフトウェア開発に対してのトレーニングが重要であるとされている。このような日常的な訓練などの組織的活動については本提案では直接扱わない。

SP2.1 については、人間が介在した計測データの信頼性や客観性に関する議論が必要となるため、本稿においては対象外とせざるを得なかった。GP2.1 と GP2.5 については、本稿で提案する計測と分析の PA に関する枠組みのみではなく、他の PA 内での活動と関連した議論が必要であろう。

5. まとめ

本論文においてプロセスコンポーネントを用いた測定・分析活動支援手法について記述した。具体的には、まずベース尺度定義ファイルと導出尺度定義ファイルの 2 つのメタデータファイルを用いて測定・分析活動の支援を行う手法について提案した。次にプロセスコンポーネントを用いた PDSP において、測定・分析活動の整合性の判定の手法について提案を行なった。現状は測定・分析活動の支援に関する提案手法に基づいて、支援環境の要件を整理し試作を進めているところである。今後の課題として、考察にあげた未対応項目についても検討を続けたい。

参考文献

- [1] M. C. Paulk, et al, "Capability Maturity Model for Software Version 1.1", Software Engineering Institute, CMU / SEI-93-TR-24, 1993.
- [2] CMMI Product Team, "Capability Maturity Model Integration for System Engineering, Software Engineering, and Integrated Product and Process Development Version 1.1", Software Engineering Institute, CMU/SEI-2002-TR-004, 2002.
- [3] 井上 克郎, 松本 健一, 飯田 元, "ソフトウェアプロセス", 共立出版, pp97-119, 2000.
- [4] 後藤徹平, 飯田元, 松本健一, 田中康, "プロセス資産の再利用を支援するプロセスモデルの提案とそれに基づくプロセス中心型開発環境の試作", 電子情報処理学会ソフトウェアサイエンス研究会(SIGSS), 鳥取環境大学, 1, 2003.
- [5] Hajimu Iida, Yasushi Tanaka, and Ken'ichi Matsumoto, "Daibutsu-den: A Component-based Framework for Organizational Process Asset Utilization", 4th International Conference on Product Focused Software Process Improvement, Rovaniemi, Finland, December, 2002.
- [6] D. N. Card, R. L. Glass, "Measuring Software Design Quality", Prentice Hall, 1990.

表 3 PA「測定と分析」の充足度評価

「測定と分析」の SP と GP	コンポーネントモデルの利用による効果		提案手法の付加による効果	
	判定と説明		判定と説明	
SP1.1 測定目標の確立		PDSP 作成時に考慮され Measurement Output で実装	左記に同じ	
SP1.2 尺度の明確化		タスクで使用する尺度は決定されるが明確化はできない	ベース尺度と導出尺度の定義ファイルの記述を行なう際に尺度の明確化がされる	
SP1.3 データの収集手段及び格納手段を明記する		タスクの Measurement Input / Output の記述により収集・格納が明示される	分析されたデータの格納は組織の測定・分析担当者によって行なわれ明確化される	
SP1.4 分析手順の明記		分析のプロセスは記述できるが分析データ作成までではない	導出尺度定義ファイルによって分析のデータが作成される	
SP2.1 測定データ収集		測定活動の存在はタスクに記述できるが測定データ収集作業の詳細は規定していない	ベース尺度定義ファイルとタスクの組み合わせによって測定活動が実現される。ただし、一部の尺度については、未対応である。	
SP2.2 測定データ分析	x	分析タスクの記述はできるが分析作業記述の詳細は規定していない	導出尺度定義ファイルと PC によって分析活動が行なわれる	
SP2.3 データ・結果の格納		測定計画書や分析報告書は測定計画タスクの Source Output, 集められたデータの集合は Measurement Output で記述可能だが尺度細かい扱いが未定	左記以外に SP2.3 で求められる作業成果物として尺度の仕様があり、ベース尺度・導出尺度定義ファイルによって対応	
SP2.4 結果の伝達		結果伝達の対象者はタスク内に記述されるが測定・分析担当者の情報が無い	測定・分析担当者名はベース尺度・導出尺度定義ファイルに記述され、結果はタスク内に記述される作業対象者に伝達される	
GP2.1 組織方針の確立	x	タスクレベルでは組織の測定・分析ポリシーを明示できない	測定・分析活動の系統的検証の手段を提供できるが、ポリシーの明示までは対応していない	
GP2.2 プロセスの計画		プロジェクト計画によって PDSP を作成する事によって可能	左記に同じ	
GP2.3 資源の提供		測定タスクの Measurement Input の記述によって資源が提供される	左記に加えベース尺度定義ファイルには測定・分析担当者名が付加される	
GP2.4 責任の割当	x	測定・分析活動担当者をプロセスコンポーネントに記述はできない	測定・分析担当者名はベース尺度・導出尺度定義ファイル中で明示される	
GP2.5 トレーニング	x	トレーニング活動はソフトウェア開発以外の活動において行なわれる	x	左記に同じ
GP2.6 構成管理		データの収集・格納手順は Measurement Input / Output によって実現され、分析結果の報告書や草案は Source Output で実現され、ベース尺度や導出尺度の仕様書は含まれていない	ベース尺度・導出尺度の仕様書はベース尺度・導出尺度定義ファイルとして実現され、PDSP の一部としてプロジェクトごとに保管される。	
GP2.7 直接利害関係者の特定と関与		測定データのデータ源に関与する作業者はプロセスコンポーネント内に記述されるが測定・分析担当者に関する記述は無い	ベース尺度・導出尺度定義ファイルの記述において測定・分析担当者が特定される	
GP2.8 プロセスの監視・制御		タスクの組み合わせによって制御の機構を作成する事はできるが基準となる尺度のデータの生成まではできない	PC による制御の機構に加え判定の材料となる測定・分析のデータを提供する事ができる	
GP2.9 忠実さの客観的評価		レビューにおいてベース尺度・導出尺度の仕様、データの収集・格納手段、分析結果・報告草案などが利用される。ベース尺度・導出尺度の仕様書はベース尺度・導出尺度定義ファイルを指す それ以外は PC の Measurement Input / Output と Source Output で得られ、評価のレビューは PDSP において定義される	ベース尺度・導出尺度定義ファイルは支援手法によって作成される	
GP2.10 上位レベルの管理層と共に状況をレビュー		レビューの活動は PDSP で定義され、測定・分析活動の結果は Measurement Input / Output によって得られ、測定・分析活動や状況は測定・分析担当者に確認する事で確認される	左記に同じ	
GP3.1 定義されたプロセスの制度化		PDSP を作成する事で定義されたプロセスを作成する事が可能	左記に同じ	
GP3.2 改善情報を集める		PDSP に従ってソフトウェア開発が行われた場合、作業成果物、測定結果が得られ、改善情報に関しては PC に記述される全ての作業者に確認する事で得られ、しかし使用された尺度や測定・分析担当者の情報は含まれていない	左記の情報に加え、ベース尺度・導出尺度定義ファイルを参照する事で使用された尺度や測定・分析担当者の情報を得る事ができる	