

プロダクトベース・プロセス支援の提案と実現

山本里枝子† 上原忠弘† 森偉作‡ 吉田裕之†

lisun@flab.fujitsu.co.jp

† 富士通研究所 ソフトウェア研究部

‡ 富士通 システム事業推進本部

† 〒 211 川崎市中原区上小田中 1015

‡ 〒 261 千葉市美浜区中瀬 1-9-3

ソフトウェア開発作業のモデル化に関して、作業の管理の対象としてプロダクトを中心に捉えるプロダクトベースプロセス支援を提案する。プロダクト開発を動的に計画し管理する枠組を導入し、各プロダクト開発を支援するプロセスを、動的かつ並行に制御する。さらにワークフロー管理システムを利用する実現を報告する。ソフトウェア開発作業を一プロダクト回覧型と多プロダクト開発型の2タイプに分類し、各タイプで提供すべき機能を報告する。

Proposal and Implementation of Product Based Process Support

Rieko YAMAMOTO† Tadahiro UEHARA† Isaku MORI‡ Yoshiyuki YOSHIDA†

† Software Laboratory, Fujitsu Laboratories Ltd.

‡ System Business Group, FUJITSU LIMITED

† 1015, Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa, 211 Japan

‡ 9-3, Nakase 1-Choume, Mihama-ku, Chiba-shi, Chiba 261, Japan

We propose a new process support approach, Product Based Process Support. In order to model software development activities, products are our main target to manage those activities. We introduce a new product management framework which software developer can plan and manage products. The framework controls product-development-management processes concurrently and dynamically. We also propose implementation of this approach with a work-flow system. We divide software development activities into two groups, one product circulation type and many documents development type, and describe functions to support each type.

1 はじめに

Osterweil[1]がプロセスプログラミングという概念を提唱して以来、ソフトウェア開発のプロセスをモデル化し、形式的に記述する研究が盛んに行なわれている。一方、CMM(Capability Maturity Model)、ISO9000、SPICE(Software Process Improvement and Capability dEtermination)などのソフトウェアの品質保証の枠組が提案されており、契約取り引きに認証を必要とする等の社会的な動きがある[2][3]。これら枠組の記述方法の提案と枠組間の比較が報告される[4]等、ソフトウェアプロセスの技術ソフトウェア開発現場の実務作業へ適用することに期待が高まっている。

我々は、実際のソフトウェア開発現場へのソフトウェアプロセスの導入を目的に、ソフトウェア開発環境においてプロセス管理ツールはそれと同様なサービスを実現する既存ツールと連係を図るべきであるという立場を提案し、プロセス管理ツールとプロジェクト管理ツールの連携の実現を報告した[5]。この研究活動を通じて、実際に開発現場にプロセス支援を導入する際の問題点が明らかになってきた。本稿ではこの問題点とソフトウェア開発作業の特徴を論じ、そこからプロダクトベース・プロセス支援を提案する。さらにワークフロー管理システムを利用する実現方式を報告する。ここでの「プロセス」は作業のフローととらえる。各作業は、フォーム、ロール(役割)を持つとする。フォームはその作業ステップの情報を表示するユーザインタフェースであり、ロールはその作業の担当者の仮の名前である。また、定義したプロセスをプロセステンプレート、それを実際に起動させたものをプロセスインスタンスと呼ぶ。

2 ソフトウェア開発作業の特徴

我々はソフト開発におけるプロセス支援に、次の特徴があると考えている。

前の作業が進まないとい次の作業の構成が決まらない 前工程の作業により、その次の工程の作業構成が決まる。

作業の順序関係を強制しにくい 前の作業が終了しなくても着手可能な作業を開始する、修正作業が生じた場合に前作業の結果を修正をしながら本流の作業も進めている、等。大きな作業粒度では現状にあわず、細かな作業粒度では作業管理は望まれない。

これらの特徴をふまえてプロセス支援を実現し、かつ、ユーザに受け入れられることを目的に、我々は工程管理ツールからのデータ変換アプローチを提案した。現在工程管理ツールで管理されている開発作業の階層レベルを利用し作業の粒度はそのまま、工程管理ツールのスケジュールからプロセス管理ツールへとデータ変換する。さらに工程管理されている作業を一部追加し作業手順を規定する。

図1において、各楕円が作業、楕円中の直線の上がロール名で下が作業名を表し、有向線がフローの方向を示す。工程管理ツールで規定された作業(図中の上部の線表に定義された作業)はそのままプロセス定義中で、UI設計、システム構造設計、統合テスト仕様(の作成)、統合テストとして作業ステップ化する。さらにそれに必要なレビューを追加している。また、線表の空き部分に相当する作業(プログラム開発)を追加する。また、前工程の作業をうけて次工程の作業構成を決める機構を導入した。例えばシステム構造設計の結果を受けて、プログラム開発の数を動的に変更する。

工程管理ツールのデータ変換アプローチに対して、ソフトウェア開発の実態とは異なるとの指摘が開発技術者からされた。このアプローチでは作業の順序関係が現状の比較して強制され過ぎる、と作業者は感じる。確かに作業手順は存在するが、それはこの提案よりも、「より小さな断片」として存在し、かつ緩やかに運用されている。手順を単純に詳細化し細かな粒度で定義すると、作業者の違和感が増大される。また、開発作業を認識する時、「作業項目」を主体として認識することに開発者は違和感を覚える。現実の実体として認識できる「オブジェクト(プロダクト)」を主体として、作業者は開発作業を捉えている。

現状は例えば、「レビューをしたら○○図と××図を修正することになった。これを修正するとも資料Aも直さないといけない。各担当者が次のレビューまでにやっておく」というような小さな手順が動的に発生し、それが順次、遂行されていく。実体として存在するプロダクトを主体として作業が認識されている。

以上から、ソフトウェア開発作業の特徴として次をも認識すべきである。

作業の順序関係は、小さい断片でなら認識されている

管理の対象をプロダクトで捉えている 作業の実体として「作業」を中心にモデル化するとユーザにはなじめない。

3 プロダクトベース・プロセス支援

これらの特徴をふまえてプロセス支援を実現し、かつ、ユーザに受け入れられるための前提は、

- 小さな断片として作業プロセスを利用する。最初から定義するのではなく、場面に応じて工程に応じて動的に定義させて利用させる。これを何らかの形で大きな開発作業のカテゴリで管理する。
- プロダクトを管理の対象とする。作業(ある意味で人)の管理はその裏側にあるが、それを強調しない。

である。

システムレベル

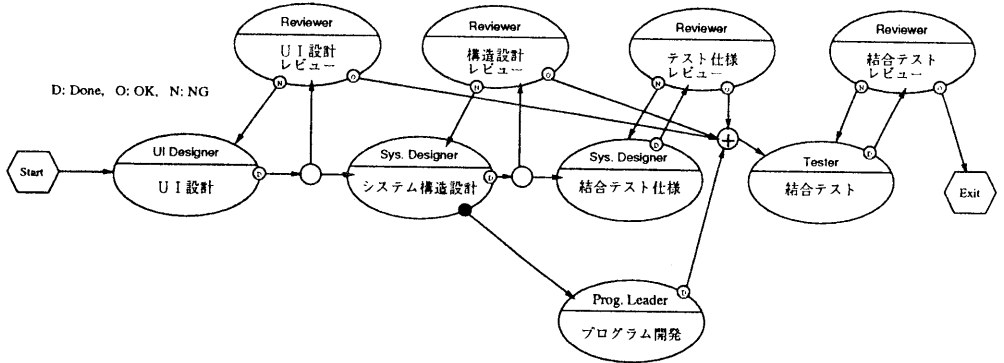
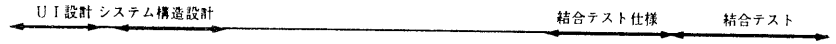


図 1: データ変換アプローチにおけるプロセス定義の例

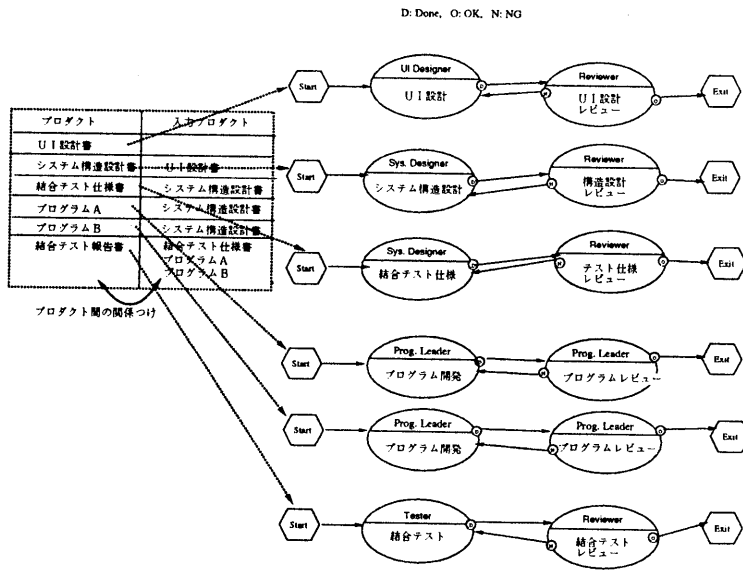


図 2: プロダクトベースプロセス支援におけるプロセス定義の例

我々はこの前提を基に、作業の管理の対象としてプロダクトを中心に捉えるプロダクトベースプロセス支援を提案する。ここでは作業プロセスを小さな断片として、一プロセス定義が一つのプロダクトを対象とする大きさにし、プロセス定義を単純化する。作業の順序関係に「プロダクトの作成順序」は作り込まない。

これまでの作業プロセスの構成との比較する。図1のプロセス定義は、プロダクトベース・プロセス支援では図2の右部分として捉える。図1のプロセス定義を、作成されるプロダクト毎の作業に分割する。それらが並行に動くことを可能とする。

これら作業の断片からソフトウェア開発作業全体を構成するために、これらに関係づける機構を導入する。

まず、プロダクト開発プロセスの集合に、緩い順序制約を設けるための枠組を設ける。例えば、図2の左部分では、プロダクト間の関係に基づく順序制約を、人間の判断を介在させながら定義する。それを基に小さな規模の各プロセス間に順序関係を与える。

また、この枠組では、開発作業を進めながら、動的にユーザがその先の作業を決める。作成する（修正する）プロダクトがわかった時点で、小さな作業断片である、そのプロダクトの作成を支援するプロセステンプレートをインスタンス化していく。これまでのプロセス支援が、一旦想定される全ての作業が定義されたプロセスを、（もし階層的に定義されるなら適切な階層レベルにおいて）生成するのに対し、この枠組は部分的にプロセスを生成し順次追加していく点が異なる。図1と図2に関して、図2のプロセスは、図中のプロセスを作業開始時点ですべてを作成する必要はなく、必要になった時点で、すなわちあるプロダクトを作成しようと人間が判断した時点で作成する。図1ではあらかじめ認識される作業全てが一つのプロセスとして定義されているので、その中の作業は一旦すべてインスタンス化され、その後一部を動的に変更する。

プロダクトベース・プロセス支援は、プロダクト間の関係を管理する機能と、プロダクトとプロセスインスタンスを関連づける機能、及び、各プロダクト（その背後にあるプロセス）の作業状況等を提示するプロダクトベースインタフェースを持つ、動的計画管理機構により実現する。動的計画管理機構は、作業の小さな断片を組み合わせ、あたかも順序関係に基づいた大きなプロセス定義のように振舞わせる。また、ユーザにはあくまでの“あるプロダクトの作成状態”を管理対象として見せる。各プロダクト開発を支援するプロセスを動的かつ並行に制御する。

個人の作業は、プロセスインスタンスを管理するプロセス管理システムにより各人の作業が認識される。一般に何らかの手段で各人に自分がすべき作業が提示されて、各作業の対象プロダクトの編集ツールの起動等の支援がなされる。

3.1 プロセスの構成

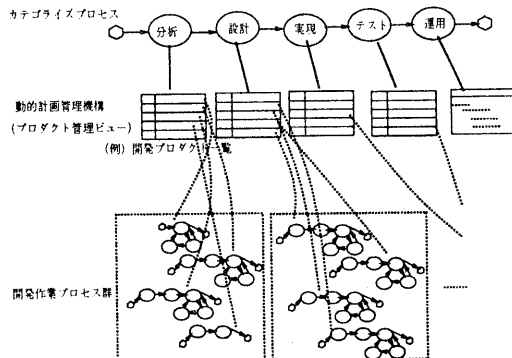


図3: プロダクトのカテゴライズ

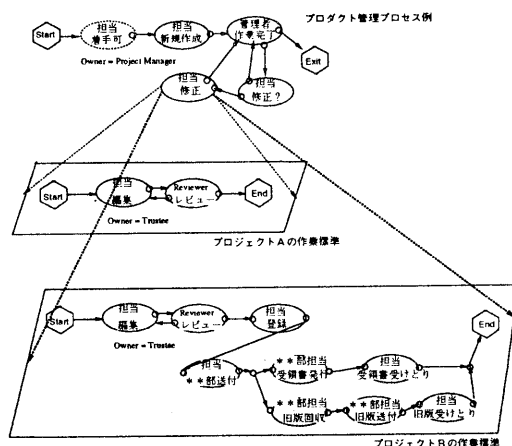


図4: プロダクト作成支援プロセスの階層例

プロダクトベースの場合、プロダクトのカテゴライズの基準が問題となる。カテゴライズにはある程度大きな粒度を用いたが、順序関係を強く規定するのは望ましくない。一手段として、例えば図3のような開発フェーズに対応するプロセスがある。

図3において、カテゴライズプロセスは、開発フェーズに大きく分けたプロセスである。開発作業プロセスはそれぞれの作業が発生したときに作成されるプロセスで、プロダクトの種類によりその構成が異なる。動的計画管理機構は上記2層のプロセスをリンクする役割を果たし、カテゴリ毎のプロダクトベース・インタフェースを持つ。この場合、プロダクトベース・プロセス支援は次の3種類のプロセスから構成される。

但し、このカテゴライズプロセスも一度に複数の作

業ステップの起動を可能とする、等、プロダクト開発の作業順序関係に強い規定を持ち込まない。これまでの作業中心のアプローチでは、カテゴライズプロセスから作業をより小さい粒度で捉えて詳細化し、さらにそれら詳細な作業間に順序関係を定義して、開発作業プロセスとした。開発作業プロセス郡で、作業の最初から最後までが定義され、強い順序関係が規定されていた。プロダクトベースプロセス支援では、動的計画管理機構によりこれまでのプロセス定義の捉え方との異なる、緩い順序関係をもつ開発作業プロセス郡を実現する。

実際にプロダクト開発作業を支援するプロセス構成は、図4のように、プロダクト作成を管理する一般のプロセステンプレートとプロジェクト固有の作業テンプレートの階層をわけて提供することで、各プロジェクトのプロダクト管理運用に合わせたカスタマイズを可能とする。図4では、「修正」作業を、プロジェクトAでは担当者による編集とReviewerによるレビューに詳細化するが、プロジェクトBでは修正したプロダクトを他の部署に送付する作業群がさらに付加されている。

3.2 プロダクトベース・インタフェース

動的計画管理機構で提供するプロダクトベース・インタフェースの概要を、動作例により示す。図5、図7は、プロダクトベース・インタフェースの一例である。プロジェクト管理者または開発作業者は、作成するプロダクトの名前と種別、必要であればプロダクト間の依存関係を「入力資料」を指定することで定義する。

管理者判断の要不要

	種別	名前	入力資料名	担当者	状況
<input type="checkbox"/> 進める	現行システム資料(D2)	処理手順規定	無	高野	作業完了
<input type="checkbox"/> 進める	現行システム資料(D2)	SYOR1001	無	高野	作業完了
<input checked="" type="checkbox"/> 進める	管理対象定義 (ETD)	JYUTYUU	SYORIGAI SYOR1001 JYUTYUU	大西	-
<input type="checkbox"/> 進める	イベントシーケンス(ESO)	-	-	-	-

図5: インタフェース動作例(1)

図5の一覧表は、一行が一プロダクトに対応し、その背後に一プロセスインスタンスが作成され管理されている。プロダクト管理のプロセステンプレートの例を図6に示す。プロセスインスタンスで起動中の作業ステップがこの一覧表の「状況」に表示され、ユーザは進捗を把握する。あるプロダクトの作成作業が着手可能となった場合を示す。

1. あるプロダクト(SYOR11)の収集作業が終了した旨が、そのプロセスインスタンスから動的計画管

理機構により把握され、インタフェースに反映された。動的計画管理機構は、「入力資料」のすべてが「作業完了」となったプロダクトを検索し、それが着手可能であることを、「管理者判断の要不要」欄を反転して示す。図5。

2. 管理者あるいは作業者その管理者判断を認識し、着手すると判断するなら、「進める」ボタンをクリックする。図5。
3. このプロダクトに対応する開発作業プロセス、図6がインスタンス化されて担当ロールに「担当」のデータが割り付けられる。そのプロセスインスタンスが「新規作成」状態となったことがインタフェースに反映される。(つまりSYOR11作成作業の次作業としてJYUTYUU作業が実行された。)担当者は、プロセス管理システムにより作業支援される。図7。

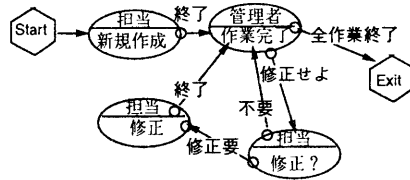


図6: プロセステンプレート例

管理者判断の要不要

	種別	名前	入力資料名	担当者	状況
<input type="checkbox"/> 進める	現行システム資料(D2)	処理手順規定	無	高野	作業完了
<input type="checkbox"/> 進める	現行システム資料(D2)	SYOR1001	無	高野	作業完了
<input type="checkbox"/> 進める	管理対象定義 (ETD)	JYUTYUU	SYORIGALD1 SYOR1001.D2 JYUTYUU.BFD	大西	新規作成
<input type="checkbox"/> 進める	イベントシーケンス(ESO)	-	-	-	-

図7: インタフェース動作例(2)

同様にプロダクト間の関係を利用した修正の指示も支援できる。あるプロダクトが「修正中」から「作業完了」に進んだことを、動的計画管理機構が把握し、その修正済みプロダクトを入力とするプロダクトを検索して、それらプロダクトに修正作業の必要性があることを、一覧表に表示。プロダクト全体の管理者が、プロダクトの修正作業の必要性を判断し、必要なら修正作業指示。そうでなければ「判断要」の表示を消す。

また、プロダクトの作成順序を強く規定する手段として、プロダクトの種別が利用できる。開発方法論で規定されたプロダクトの開発順序を、次の作業対象候補プロダクトを選択する際に用いる。

4 プロダクトベース・プロセス支援の実現

プロダクトベース・プロセス支援の実現例として、プロセス管理システムにワークフロー管理システムを用いた実現を報告する。

構成概要を図8に示す。動的計画管理機構とワークフロー管理システムを融合した、ソフトウェア開発作業支援システムである。作業指示、各プロセスの進捗管理、個人作業環境でのツール起動等は、ワークフロー管理システムにより行なわれる。プロダクト間の関係、プロダクトとワークフローの関係は動的計画管理機構のリポトリに格納する。

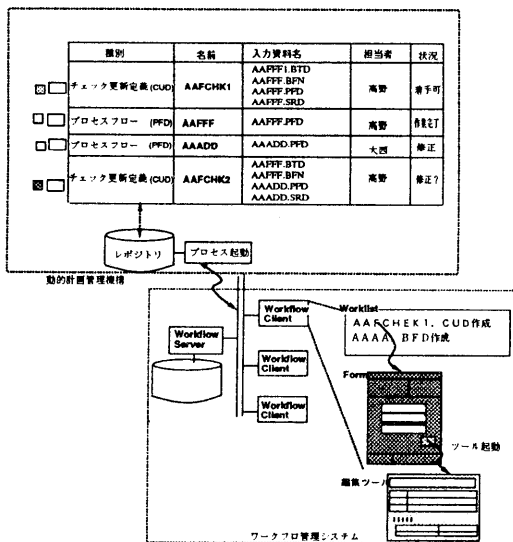


図8: 動的計画管理機構とワークフロー管理システム

動的計画管理機構は、プロダクト開発プロセス間の緩やかな制約の定義とその制約に基づくプロセス間の順序関係の制御等を行ない、主に作業管理者や開発責任者を支援する。ワークフロー管理システムは、個人に割り付けられた仕事をワークリストに提示し、さらに各仕事に応じたツール起動等を行ない、主に作業担当者を支援する。

我々は、ソフトウェア開発作業を一プロダクト回覧型と多数プロダクト開発型の2タイプに分類する。一プロダクト回覧型は、主に一つの帳票の処理を目的とする作業であり、障害票等の連絡用帳票の処理はこれに含まれる。また、多プロダクト開発型は、仕様書、プログラ

ム等の複数のプロダクトの開発を目的とする作業である。どちらのタイプも複数作業による共同作業を想定しているが、タイプにより動的計画管理機構の機能が異なる。各タイプの実現機能を検討し報告する。

4.1 多プロダクト開発型支援

多プロダクト開発型支援は、相互に依存関係のある多数の開発プロダクトの作成を支援する。開発の進捗により作成プロダクトとそれに対する作業が変化する。作業は作業者間で適宜調整されて並行に実行される。

多プロダクト開発型の場合、動的計画管理機構を操作するのは計画を書く人（主にプロジェクトリーダー）であり、期待される支援機能は、プロダクト間の作成順序の制御や依存関係等の緩やかな制約の定義である。このため、プロダクトベース・インタフェースは、プロダクトの名前や種別が必須項目となる。

多プロダクト開発支援のシステム構成を図9に示す。

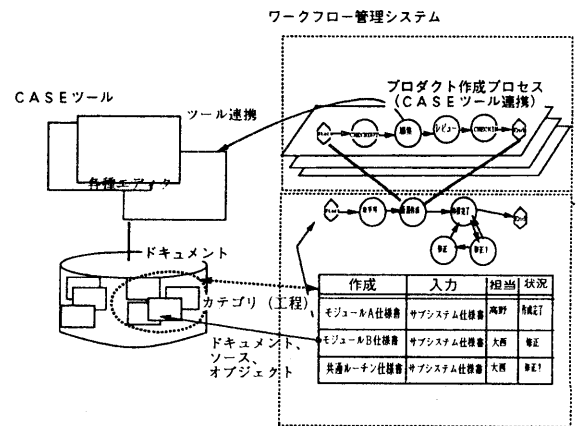


図9: 多プロダクト開発支援のシステム構成例

個々のプロダクト作成は、CASEツールとの連携により支援する。異なる種別の複数のプロダクトに関して、名前、担当者、期限等を指定して作業計画を立てる。計画作成担当者が作業全体の進捗に応じて作成予定プロダクトを追加する。計画は内容を変更されながら、動的に規定されていく。その計画の内容にあわせてワークフローのテンプレートからデータを生成し、各プロダクト作成作業をワークフローを用いて進捗管理する。

プロダクトベース・インタフェースは、異なる種別の複数プロダクトを一元的に管理してユーザに提示する。主な機能は以下である。

1. プロダクト群の情報を確認する/変更する。

- その一覧表に関して、名前、作業管理者の名前、プロジェクト名/システム名/サブシ

テム名等を確認する／変更する。

- 各プロダクトについて、名前、種別、プロセステンプレートの名前、作成作業の担当者、スタート／締切日時、現在の状況を確認する。
- 指定したプロダクトの作業情報を確認する。
- 指定したプロダクトの作業履歴を確認する。
- 各プロダクトについて、現在の状況は自動的に変更される。「スタート日時」は自動的に設定される。
- プロダクトを追加する、削除する。
- プロダクトを指定し、参照するプロダクトを設定する。
- プロダクトを指定し、その行に対応するワークフローの各作業ステップのロールに割り当てられた作業名を確認し変更する。

2. 実行する。

- 依存関係に基づき、作成／変更作業を行なうべきプロダクトが提案される。自動的に行なわれる。
- 作成／変更作業を提案されたプロダクトを選択する。選択したプロダクトに関して作業指示する。

4.2 一プロダクト回覧型

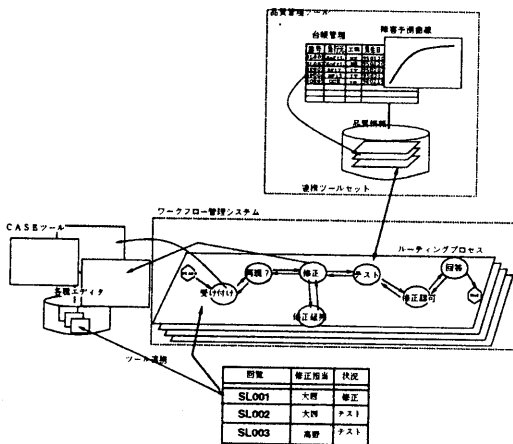


図 10: 一プロダクト回覧支援のシステム構成例

一プロダクト回覧型支援は、一つのプロダクトのルーティング作業の確実な運用を支援するもので、ルーティングの進捗によりプロダクトとそれに対する作業が変化する。作業は、多くの場合作業員間で独立、かつ、並行に実行されている。

一プロダクト回覧型の場合、動的計画管理機構を操作するのはそのプロダクトに関わる作業員全員である。

期待される支援機能は、プロダクトの一貫性を保証することであり、このため、小さな断片である作業を大きなプロセスに構築することは必要ない。

一プロダクト回覧支援のシステム構成を図 10 に示す。個々のプロダクト作成は CASE ツールとの連携により支援する。また、品質管理ツールとの連携が有用である [5]。各プロダクトに関して、名前、担当者、期限等を指定するが、進捗管理が主な目的である。回覧するプロダクトのルーティングテンプレートからデータを生成し、各プロダクト回覧作業をワークフローを用いて進捗管理する。

プロダクトベース・インタフェースは、種類の複数プロダクトを一元的に管理してユーザに提示する。主な機能は以下である。

1. プロダクト群の情報を確認する／変更する。

- その一覧表に関して、名前、作業管理者の名前、プロジェクト名／システム名／サブシステム名等を確認する／変更する。
- 各プロダクトについて、名前、種別、プロセステンプレートの名前、作成作業の担当者、スタート／締切日時、現在の状況を確認する。
- 指定したプロダクトの作業情報を確認する。
- 指定したプロダクトの作業履歴を確認する。
- 各プロダクトについて、現在の状況は自動的に変更される。「スタート日時」は自動的に設定される。
- プロダクトを追加する（プロダクトの一貫性を保証するよう、自動的に名前を作成）、削除する。
- プロダクトを指定し、参照するプロダクトを設定する。
- プロダクトを指定し、その行に対応するワークフローの各作業ステップのロールに割り当てられた作業名を確認し変更する。

2. 実行する

- プロダクトの回覧を指示する。

4.3 開発作業員の作業支援

開発作業員が受ける主な支援は、以下である。

1. ワークリストを確認する。ワークリストには、作業の名前、作業対象プロダクトの名前、作業の締切期日や残日数、作業計画表の名前が提示される。常に最新のワークリストが表示される。
2. 割り付けられた作業から一作業を選択し、その作業を行なう。
 - (a) 作業とそれが属するワークフローの情報を確認する

- 作業に関する情報を確認する。ワークリストに提示された内容に加えて、作業内容を提示する。
 - その作業に至る履歴を確認
- (b) 作業が対象とするプロダクトやそれに関連するプロダクトを編集する。
- その作業の対象であるプロダクトを編集。
 - 参照プロダクト群を確認し変更する。各参照プロダクトについて名前や依存関係を確認する。
 - 参照プロダクトを選択して、参照する。
- (c) 担当者が判断の選択肢中から、報告すべき判断を選択し、次の作業へと制御を移す。

5 まとめ

我々は、以下のソフトウェア開発作業の特徴を抽出した。

- 前の作業が進まないと次の作業の構成が決まらない
- 作業の順序関係を強制しにくい。小さい断片には順序関係は認識される。
- 管理の対象をプロダクトで捉えている

これから、小さな断片として作業プロセスを利用すること、及び、プロダクトを管理の対象とすることを、ソフトウェアプロセスによる開発作業支援の前提としてプロダクトベース・プロセス支援を提案した。そして、小さな断片であるプロセスを「一プロセス定義が原則として一プロダクトを対象とする」とし、それら作業プロセス間に緩い順序関係を定義し制御する機能と、その定義と制御をプロダクトを対象に行なうためのプロダクトベースインタフェースを持つ、動的計画管理機構を示した。

さらに、プロセス管理システムにワークフロー管理システムを利用した実現例を報告した。これは、動的計画管理機構とワークフロー管理システムを融合した、ソフトウェア開発作業支援システムである。ソフトウェア開発作業を一プロダクト回覧型と多プロダクト開発型に分類し、それぞれについて動的計画管理機構がもつべき機能を提供する。これにより、開発ドキュメント管理機構の下での非定型な並行共同開発作業のプロセス支援が実現され、ソフトウェア開発作業全体の管理・効率化を可能にした。

現在、帳票管理体系についてはPMVIEW/Qの帳票類を、プロダクト管理体系としてAA/BR MODELLING[6]を対象としたシステムを、次期ワークフローシステム上で開発している。AA/BR MODELLINGで対象となるプロダクトは、分析/設計フェーズにおける各種ドキュメント25種類である。

今後は、ソフトウェアプロセス導入による作業効率の向上を確認し、組織のプロセス改善を図る試みへと発展させていく。

参考文献

- [1] L. J. Osterweil. Software process are software too. *ACM Transactions on Computer Systems*, pp. 2-13, 1987.
- [2] 飯塚悦功. ソフトウェア品質保証モデル iso9000-3 の意義. 情報処理学会 ソフトウェアプロセス・シンポジウム, pp. 107-114, June May.
- [3] 堀田勝美. プロセス成熟度モデル. 情報処理学会 ソフトウェアプロセス・シンポジウム, pp. 115-122, May 1994.
- [4] Katsuro Inoue, Atsushi Watanabe, Hajime Iida, and Koji Torii. Modeling method for management process and its application to cmm and iso9000-3. In *Proceeding of the 3rd International Conference on the Software Process*, pp. 85-98, Reston, October 1994.
- [5] 山本里枝子, 吉田裕之. プロジェクト管理ツールとプロセス管理ツールの連携の実現. 情報処理学会 ソフトウェア工学研究会, pp. 147-152, January 1995.
- [6] 富士通. システム分析設計技法 AA/BRMODELLING 解説書, September 1993.