

ソフトウェア開発プロジェクトの進捗管理に関する一方式

樋山 淳雄¹古宮 誠一²

情報処理振興事業協会 技術センター

著者らはオブジェクトデータベースを中核とした総合的なソフトウェアプロジェクト管理システムの構築を目指している。本稿では、まずシステムの中核となるソフトウェアプロジェクト管理のためのプロセスモデルについて述べている。このモデルに基づいて立案された計画により、プロジェクト管理者が各作業者に作業指示を与え、作業員から報告される進捗情報に基づき進捗把握を行い、リスクを検出し、その影響を解析するまでの一連の作業を支援できるような枠組を提案している。このような枠組を実現するために、プロジェクト管理の視点から単一の作業に対して引き起こされるアクションを分析し、アクションにともない遷移する状態(ステータスと呼ぶ)とともに状態遷移モデルとして表現している。そして状態遷移をひき起こすアクションの起動時に、プロセスモデルに基づいて作成された作業ネットワークで規定した関係を利用して、適切な人に適切な情報を通知する枠組を提案している。この枠組を使うことにより、複数の作業が並行して行われるプロジェクトの管理者を効果的に支援することができるとしている。

A Method for Progress Control of A Software Project

Atsuo Hazeyama

Seiichi Komiya

Information-technology Promotion Agency, Japan (IPA)

Software Technology Center

Shuwashibakoen 3-chome BLDG.,

3-1-38 Shibakoen, Minato-ku, Tokyo 105, Japan

(hazeyama/komiya)@stc.ipa.go.jp

The authors aim at constructing an integrated software project management system with an object database. This paper describes a process model for software project management that forms the basis of the system. This paper also describes a framework to induce a series of works that a project manager have to perform as follows: giving work instructions in accordance with the plan that have been made based on the process model to workers, progress ascertainment based on progress reports sent from workers, analysis of the impacts by problems detected. In order to realize these facilities, they analyse actions triggered for a unit activity from the point of view of project management. Based on the consideration, they propose a state transition model involving the actions as well as states changed by actions. When an action is triggered, whether the action is effective or not is checked, information accompanying the action is notified to appropriate persons by referring the relationships specified by the process model. By using this mechanism, the authors believe it is possible to effectively support a manager of the project in which multiple activities are performed in parallel.

¹日本電気株式会社から出向²日立製作所から出向

1 はじめに

情報化社会の進展にともない、ソフトウェア開発に対するニーズが高まっている。しかも開発されるソフトウェアは大規模化、複雑化傾向にあり、このようなソフトウェアを要求された品質で、納期通りに、しかも予算の範囲内で作成することが難しい状況にある。このような状況のもと、近年ソフトウェアプロセスに関する研究開発が活発に行われており(例えば[2])、ソフトウェアプロセスに基づいた開発環境の構築が行われている(例えば[3][8][9]等)。大規模なソフトウェアの開発は、一般に多人数により構成されるプロジェクトを組んで行われる。このようなプロジェクトを成功に導くためには、プロジェクトで行われる作業を定義し、プロジェクト計画を立案し、それに基づいて作業者が作業を行い、その進捗を確認するといった管理指向の強いアプローチで行うのが現実的である。なぜならばソフトウェア開発では各工程での作業が実質的にいつ終了したのかわからないことが特徴であるが、このため、工程ごとの作業と成果物を明確にし、各工程の作業の途中/終了時点でその作業を確認する方式を採用することが現実的だからである。従って、このようなプロジェクト管理作業を強力に支援する技術の開発が望まれる。ところがこれまでに開発されているソフトウェアプロセスに基づいた開発環境は、作業者間の協調の側面を支援するもの[9]や、プロジェクト管理の側面を支援しているもの[3][8]についても管理者と作業者間の動的な協調を支援しているものは少ない。

これに対して、我々はデータベースを中核とした総合的なソフトウェアプロジェクト管理システムの構築を目指している。これまでにプロジェクト管理の基盤となるプロセスモデルを提案した。そしてこのモデルをベースに、作業の遅延がリソースの日程上の制約を考慮した場合、プロジェクト全体にどのように影響が波及するかを示す影響波及解析に適用し、その有効性を示した[1]。本稿では、工程管理の視点に対して、管理者が計画に基づいて行う作業指示から、進捗把握に基づくリスクの早期発見、リスクを早期に回避するために取る意思決定フェーズまでの作業を計算機支援により誘導するための枠組を提案するとともに具体例を挙げてその有効性を主張するものである。

プロジェクトでは多くの作業が並行して行われ、その進捗情報や終了した作業に対する承認の依頼などの情報は作業者から管理者に集中する。さらに作業実施中に何らかの問題が生じた場合には管理者はそれに早期に対処しなければならない。このような作業は、プロジェクトが大規模になればなるほど、それを即時に判断して対処することは難しく、プロジェクトに内在するリスクを見逃すことにもなりかねない。例えば、

各作業者に対して作業指示を与える場合、あるいは進行中の作業の進捗状況把握に際し、管理者は次のような判断を余儀なくされる。

- (i) ある作業の終了により後続の作業に対して作業指示をかける場合、ある作業が終了ばどの作業が開始できるのか、誰に作業指示を出せば良いのかを管理者は見極めなければならない。

また、ソフトウェア開発プロジェクトでは、作業指示をかける場合でも、作業の順序制約を注目しているだけでは不十分である。なぜならば、作業の実行主体である作業者はプロジェクトにおいて複数の作業を担っていることがあるし、さらにプロジェクト以外の作業に時間を取られてしまう(例えば、教育受講、出張など)こともあるからである。すなわち、

- (ii) 作業者に対して当初立案された計画通りに作業指示を与えてよいか。あるいは、進行中の作業を計画通りに進めてよいか。すなわち、作業や作業者の状況から考えて、作業者は指示通りに作業を実施可能かを判断できなければならない。

このようなプロジェクト管理作業に対して、管理者の意思決定を支援するために、また、プロジェクトにおいて作業を行う作業者に対して必要な情報を必要な時に必要な人に伝え、作業を誘導することが重要である。

以下本稿の構成を示す。次節では、本節で述べた問題点を解決するために枠組として満たさなければならない要件について考察する。3節ではこのような問題点を解決するための1つの要件を満たすソフトウェアプロジェクト管理のためのプロセスモデルについて概観する。4節ではソフトウェア開発の工程管理における作業指示からリスクの早期発見までの管理者が意思決定をするために必要な情報を知らせたり、管理者が採った意思決定に基づき必要な人に必要な情報を伝えるための枠組について述べ、5節でシステムの振舞いを例題により説明する。

2 問題点解決のための要件の考察

本節では、1節で提起した問題を計算機により支援するためには、どのような要件が必要かを検討する。

- (i) について：

- (a) 「作業の終了」のようなイベントを取得でき、その時に引き起こされるアクション(この場合は、「後続の作業とその作業者に作業指示を出す」)を起動できるような枠組、すなわち、Event-Condition-Actionの枠組が必要である。

表 1: 管理サイクルの各フェーズで役割ごとに行われる作業とその間で受け渡される情報

フェーズ	プロジェクト管理者	受け渡される情報	作業者
Plan	<ul style="list-style-type: none"> ・プロセスモデルを実体化したインスタンスの作成 ・作業に対して (作業者の日程上の制約と作業負荷を考慮した) スケジューリングとリソースの割り当て ・作成された作業計画に基づいた作業指示 	作業名, 作業内容, 参照プロダクト, 期限, 作業指示が出された日付 →	
Do	<ul style="list-style-type: none"> ・作業者の日程の変更通知 	← 作業名, 状況 (作業開始), 着手した日付 ← 作業名, 期間消費工数/期間開発規模, 累積消費工数/累積開発規模 ← 作業名, 状況 (作業終了) イベント名, 計画開始/終了日 →	<ul style="list-style-type: none"> ・割り当てられた作業の開始 ・仕掛中の作業の途中進捗状況の報告 ・作業終了報告
Check	<ul style="list-style-type: none"> ・作業者から報告された進捗状況に基づく問題点の検出 ・作業者の日程の変更による問題点の検出 ・検出された問題点によるプロジェクト全体への影響波及解析 ・作業終了時のチェック 		
Action	<ul style="list-style-type: none"> ・問題点 (進捗報告から検出された問題/作業者の日程の変更による問題) 解決のための計画の見直し (作業へ代替作業者の割当、作業の縮退、作業期間の延長等戦略に基づく) ==>新たな作業指示 ・作業終了時のチェックによる作業のやり直し指示 ・作業終了時のチェックによる作業完了承認 	作業名, 作業内容, 期限 → 作業名, 状況 (完了承認) → 作業名, 状況 (着手可能), 期限 →	(該当作業の作業者に対して) (該当作業の後続作業で着手可能な作業の作業者に対して)

(b) ある作業が終ると着手できる作業とは、その作業の成果物を参照しながら行う作業であり、作業間が関連づけられている必要がある。また、着手できる作業の作業者に着手指示を出す必要があるが、そのために、作業と作業の担い手が関連づけられていなければならない。

(ii) について:

(c) 作業を行う作業者の日程上の制約を考慮しなければならない。

(b), (c) の要件を満たすための枠組として、我々はソフトウェアプロジェクト管理のためのプロセスモデルを提案した [1]。このモデルについては次節で概観する。

(a) に関して、プロジェクト管理に関連して行われる作業をもう少し詳細に分析してみる。

ソフトウェアプロジェクトは大きくプロジェクトを管理する管理者とプロジェクトにおいて実作業を行う作業者から構成される。プロジェクト管理者は、プロセスモデルを実体化した作業ネットワークを作成し、各作業に対してリソースの割り当て、日程の割り当て等を行い計画を立案する。この時作業を行う作業者には割り当て可能な期間があり、作業者の都合を勘案しなければならない。そして立案された計画に基づき、作業者に作業指示を与える。作業は、その作業を行うた

めに必要な制約 (先行作業が終了していること) が満たされれば開始することができるが、その行動を起こすのは各作業に割り当てられた作業者である。作業が始まってからは、作業者は作業の進捗を管理者に報告する。管理者はその進捗状況を把握し (作業者の日程の変更の把握も含む) 予測することにより、このまま作業を進めてよいか、また、対処が必要な場合、どのようにすればよいのかということが重要となってくる。

プロジェクト管理の視点から管理者と作業者の関係を情報の流れに照らすと図 1 のように示すことができる。

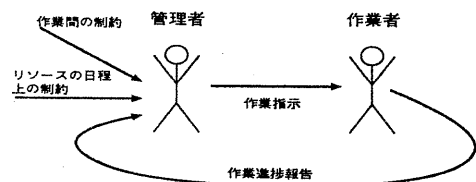


図 1: 受け渡される情報から見た管理者と作業者の関係

また、以上のようなプロジェクト管理に関わる作業を管理サイクルのフェーズごとに、どの役割によって行われるのかを表 1 にまとめる。表では、それぞれの役割間でやり取りされる情報とそのタイミングも示す。

表の中の「受け渡される情報」欄の矢印は、情報の発信者と受信者を表す。例えば、Plan フェーズで管理者が作業指示を行う時は、作業名、作業内容、参照プロダクト、期限が作業者に通知されることを示している。

この表から、管理者が計画を作成したり、問題点の検出を行うことを支援する機能と、管理者が意思決定をするために必要な情報、及び、管理者が意思決定として起こしたアクションにともない受渡される情報が記録され、しかも必要な人に必要な時に通知することが必要であることがわかる。

3 ソフトウェアプロジェクト管理のためのプロセスモデル

本節では、システムの基盤となるソフトウェアプロジェクト管理のためのプロセスモデルについて述べる。本システムが対象とするソフトウェア開発プロジェクトは、最終的に開発されるソフトウェアシステムを作り出すために多くの作業が行われ、その過程で多くの中間生成物(プロダクト)を複数人の作業者がツールやマシンを使いながら生成する過程であると考えられる。このことは概念的には SADT[5] と類似の表記を使って図2のようなプロセスモデルとして表現できる。

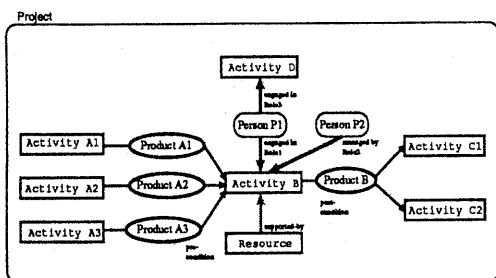


図 2: ソフトウェアプロジェクト管理のためのプロセスモデル

工程管理のための主要なオブジェクトとして、作業(アクティビティ)、プロダクト、リソース、役割、プロジェクトの5つを考えた。プロセスモデルを構築するためにに行った考察等の詳細については [1] を参照されたい。

- アクティビティ (作業): 割り当てられたプロダクトを作成するために行われる作業のこと。計画開始/終了日、実績開始/終了日、見積り工数、進捗データなどプロジェクト管理に必要なデータを属性として持つ。
- プロダクト: アクティビティで作り出されるべき

プロダクト。プロジェクト管理の観点からはアクティビティの進捗を定量的に計測する対象である。

- リソース: 作業を実行するための実行主体である。リソースには、人と、ツールやマシンといった非人間系のリソースがある。ここで重要なことは前節で述べたようにリソースの日程上の制約を表現するために、利用可能期間に関するカレンダー情報を持つことである。
- 役割: プロジェクトという組織においては、さまざまな役割形態がある。たとえば、プロジェクト管理者、設計者、プログラマ、品質保証技術者などがある。

ソフトウェア開発プロジェクトの工程と製造業における組み立てラインの工程とを比較して特徴的なことは、ソフトウェア開発プロジェクトでは、1人の作業者が複数の作業を担っており、しかも、作業ごとに異なった役割を担って関与することさえあるということである。また、プロジェクト管理者がある作業に対して作業者を割り当てる場合、その作業を行い得る役割を担うことができる作業者の中から選び出し、作業に割り当てているであろう。以上のことから役割という概念を人とは独立したものと考え、人と役割とを関連づけて考えている。

- プロジェクト: プロジェクトで行われるアクティビティやプロジェクトに従事するリソースはプロジェクトとの間にプロジェクトを構成するという関係が規定される。プロジェクト自身は、プロジェクト期間やプロジェクト予算、プロジェクトの対象領域の性質を表す属性を持つ。

これらの管理対象はさまざまな関係により関連づけられている。

4 単一作業の状態遷移モデルとプロセスモデルに基づいた情報通知機構

2節ではプロジェクト管理を支援するためには、PDCA サイクルの各フェーズを支援する機能と管理者と作業員間での情報の受渡しが重要であることを述べた。本節では上記考察に基づき、プロジェクト管理に関するプロジェクト管理者と作業員間の情報通知を支援するための枠組を提案する。

2節で考察したように、プロジェクト管理に関連して、さまざまなアクションが引き起こされ、アクション起動時にデータが設定/更新が行われる。更新された情報は必要な人に通知されなければならない。このような機能実現のためのアーキテクチャを図3に示す。ユー

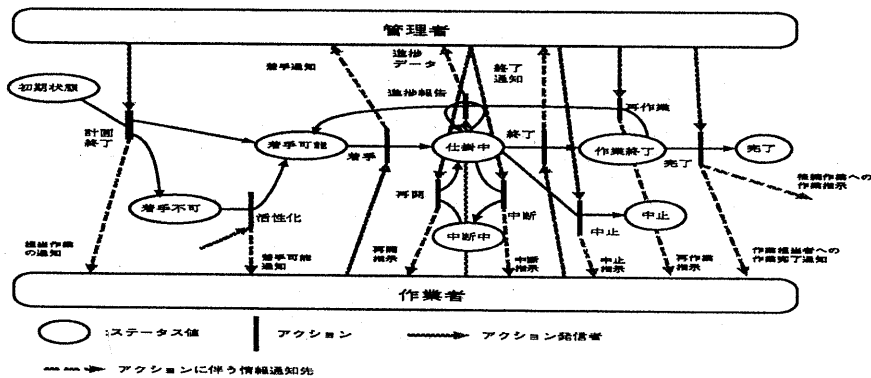


図 4: 作業に対して行われるアクションと状態遷移

表 2: アクションとその意味

アクション	意味
計画終了	管理者が作成した計画をプロジェクトの計画として確定し、各作業の担当者である作業者に通知するトリガー。
着手	作業間の先行後続関係の制約充足により着手可能な作業に対して作業者が作業を開始するアクション。
終了	作業者が作業を終了した時取るアクション。このアクションを契機としてシステムは管理者に作業の終了を通知する。
完了	終了と作業により宣言された作業に対して、管理者が作業の承認を与えるアクション。このアクションが起動されると、システムは該当する作業の作業者に作業の完了を通知する。また、完了した作業の後続作業に関して、作業間の先行後続関係に基づき、着手可能か否かをチェックする。着手可能な場合には、作業を着手可能にし（「活性化」というアクションをシステムが起す）、担当作業者に対して作業の着手可能を通知する。
進捗報告	作業者が作業の途中進捗を報告するアクション。
中断	管理者が作業の中断を決定した時に起こすアクション。
再開	中断されていた作業を再開させるために管理者が起すアクション。
再作業	終了と通知された作業に対してやり直しを求めるアクション。
中止	作業の取り辞めを指示するアクション。

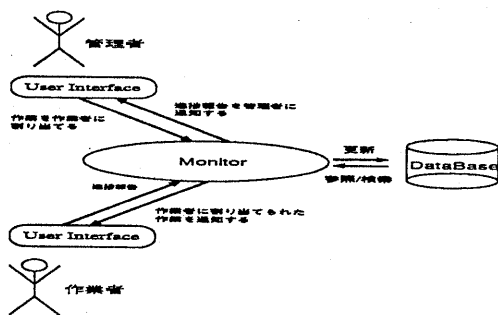


図 3: 情報通知機能実現のためのアーキテクチャ

ザインタフェースを通じて引き起こされるアクションをイベントとして捕捉し、対応する手続き（データの更新や情報の通知など）を起動するモニターと呼ぶコンポーネントと、プロセスモデルを実体化したネットワークリソースの情報を管理しているデータベース

から構成される。

また、表 1 で分析したようにプロジェクト管理に関連して管理者や作業者から起動されるアクションは、

- プロジェクトで実施される作業に対して、管理者や作業者が取るアクション
- 作業者に対して、その日程を変更するアクション

に大別される。

このうち、作業者に対する日程の変更がプロジェクトで行われる作業に対して影響が及んでしまう場合には、管理者はその日程の変更を取り辞めるか、あるいはプロジェクトの作業に対する何らかの計画変更（作業指示）というアクションとして反映されなければならない。すなわち、作業に影響を及ぼすような日程の変更を受け入れる場合には、管理者は作業に対する何らかのアクションを起こすことになる。

そこで、1つの作業に対して管理者や作業者が情報の受渡しを行う際に起動するアクションと、アクション

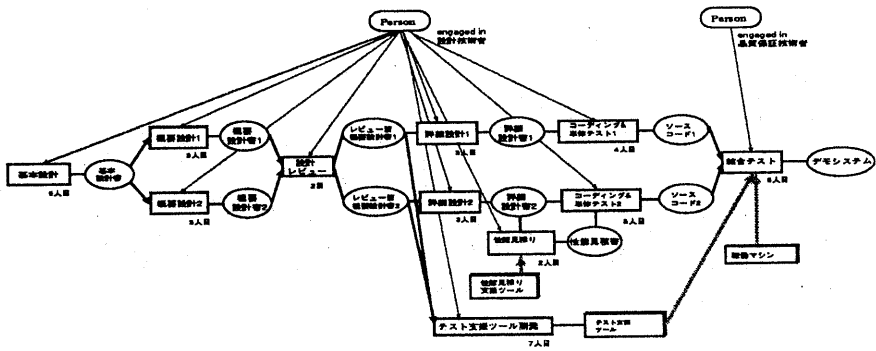


図 5: 例題プロジェクトの作業ネットワーク

ンにともない遷移する状態 (これをステータスという作業の属性として表現する) を状態遷移としてモデル化する。図 4 に各作業に対して行われるアクション、アクションの結果遷移する状態値、さらにそのアクションを誰が起動し得るか、それぞれのアクションが起動されるタイミングでどのような情報が受け渡されるかを示す。図 4 で丸は状態値を、縦棒はアクションを、黒の矢印は状態の遷移を表している。灰色の実線はアクションを誰が起動するかを示し、灰色の破線はアクションにともない、情報の通知先を示している。表 2 に図 4 で示された各アクションとその意味を示す。

管理者や作業者により各作業に対して引き起こされたアクションに基づき、システム (モニター) はデータベースを更新し、必要なデータを算出し、プロセスモデルを実体化した作業ネットワークに規定された関係を参照して必要なデータを必要な人に通知する。この機構により、ある作業の担当予定者が何らかの理由 (突然の出張等) で日程に変動があると、そのアクションが起動されたタイミングで影響を受ける作業を指摘し、管理者に通知する。また、ある作業を終える場合、その作業の作業者は作業を終了するというアクションを起こす。モニターは作業の終了というイベントを受け、消費期間や遅れの有無、その影響波及を算出し、それらのデータとともにプロジェクト管理者に作業の終了を通知する。プロジェクト管理者はその作業の終了を承認する意味で完了というアクションを起こすと、モニターはそのイベントを受け取り、作業者に対して作業が承認されたことが通知されるとともに、その作業の後続作業にメッセージを伝える。後続作業はそのすべての先行作業が完了していれば制約が充足され、「着手可能」状態になり、担当作業者に着手可能なことが通知される。それぞれのアクションが行われた時点の日付情報を履歴として記録する。プロジェクトの管理上重要な実績開始/終了日は、それぞれステータスを「仕

掛中」「完了」にした時点での日付が記録される。

このようなステータスデータと作業ネットワークをデータベース化し、その変更が起きるタイミングでどのような処理を行うのかを手続きとして記述することにより、管理者や作業者がその役割に応じて起こしたアクションから、システムはその内容を監視し、必要な処理を行うことができる。その結果、管理者や開発者の負荷が軽減され、また問題点を早期に検出することができるようになる。

5 例題

本節では、前節で述べた枠組を具体例を通して説明し、その有効性を示す。

作業概要

例題プロジェクトとして、8月1日から開催される、ある展示会出展用のデモシステムを開発するプロジェクト (プロジェクト 1) を考える。このプロジェクトは7月2日に組織され、7月31日までにシステムを完成させなければならない。システム開発のために行う作業は、基本設計、概要設計、概要設計レビュー、詳細設計、コーディング&単体テスト、結合テストと、デモシステムは高性能を要求されるシステムであるため、詳細設計の一部分に対して性能見積りを行い、その結果をコーディング作業に反映させるものとする。性能見積りは性能解析技術を必要とする。

例題プロジェクトでの役割とリソース

例題プロジェクトには、計画立案/進捗管理を行うプロジェクト管理者、設計とコーディングを行う設計技術者、テストの計画と実施を行う品質保証技術者が関与するものとする。

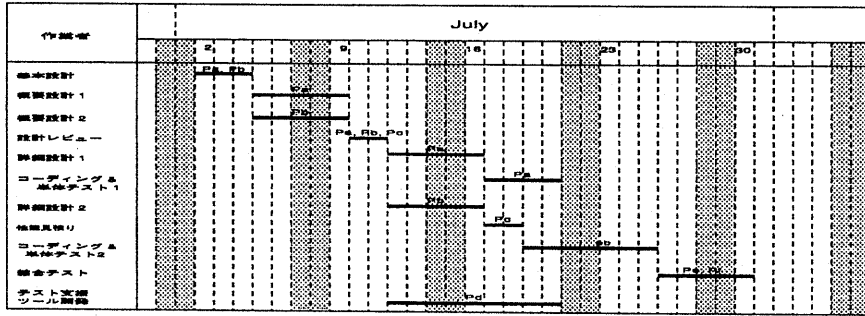


図 6: 計画時のスケジューリング結果

このデモシステム開発を行うプロジェクト 1 は組織 X のもとで行われており、この組織ではもう 1 つ別のプロジェクト 2 も行われている。プロジェクト 1 は、プロジェクト管理者に Px が、設計技術者に Pa, Pb, Pc, Pd の 4 名が、品質保証技術者に Pe, Pf の 2 名が割り当てられ、各役割に割り当てられた人はその役割に応じた作業を行う。また、プロジェクト 1 では、開発マシン、展示会で実働するマシン、性能見直しを行う際に使用するツール、テストを効率的に行うために開発するツールを使用する。リソース (プロジェクトに従事する人の属性のみを示す) に対しては、表 3 にあげるような制約があるものとする。

表 3: プロジェクト 1 に従事する作業者の属性データ (プロジェクト計画立案時点)

名前	スキル	割り当てられている作業	期間	プロジェクト区分
Px				
Pa	OOA, C			
Pb	構造化分析, C			
Pc	JSD, 仕様解析	出張 出張 出張	-7/9 7/9-7/31 8/6-8/8	プロジェクト外 プロジェクト外 プロジェクト外
Pd	構造化分析, C	研修	7/23-7/27	プロジェクト外
Pe		結合テスト	8/6-8/10	プロジェクト 2
Pf		結合テスト	8/6-8/10	プロジェクト 2

振舞い

●プロジェクトで行う作業、作業の結果生成されるプロダクト、作業の際使用するリソース、各作業に要求される役割並びに、作業間の先行後続関係をネットワークとして表現したものを図 5 に示す。各作業 (図中で□で表されたもの) の下に記された数字は、各作業に必要な見積もられた所要工数 (単位: 人日) である。各作業に必要な所要工数に基づき、時間軸へスケジューリングし、リソースの割り当てを行う (その一例を図 6 に示す)。計画が確定された時点でプロジェクト管理者が計画終了というアクションを起こすと、各作業に対して割り当てられた作業が通知されるとともに、先

行作業のない作業は着手可能となる。ここでは、基本設計作業のステータスが着手可能になり、担当作業者 Pa と Pb に基本設計作業が着手可能状態になったことが通知される。

●当初の予定では、概要設計 1 には作業者 Pa が割り当てられていたが、作業者 Pa は 7 月 6 日から 9 日まで急きょ出張に出かけなければならないこととなった。管理者がそのインタフェースからカレンダーに出張予定を入力すると (図 7 の (a))、モニターはその変更イベントを受け取り、作業者 Pa にその期間に割り当てられている作業があれば、それを通知し、出張を Pa に割り当てるか否かを問う (b)。ここで管理者が Pa に割り当てることを決定するとすると、担当者に通知される (c)。プロジェクト管理者は、レビューに参加予定の作業者 Pc に概要設計作業を割り当て、計画終了アクションを起こした (d)。ところが、作業者 Pc もこの期間は他の作業により、利用可能でないため、モニターは管理者に対して警告メッセージを通知する (e)。そこで作業者 Pd に作業を割り当てることとした。計画終了のアクションを取ると (f)、作業者 Pd に作業が割り当てられ、作業者 Pd に作業が通知される (g)。

●レビューが当初の計画より一日遅れて 7 月 12 日に終了した (終了ステータスが入力される) (図 8 の (a))。モニターはこのステータス変更イベントを受け、管理者に通知する。この時モニターは、作業の終了日時と計画終了日から遅れた日数を算出するとともに、遅れの影響が後続に波及するか否かを管理者に通知する。この場合は遅れの影響が後続作業に波及するため、その旨を通知する (b)。管理者はその影響がどのように波及するかは影響波及解析機能を使って、調べることができる。影響波及解析機能は、作業ネットワークの作業間の関係、また、作業に割り当てられているリソースが持つカレンダー上の制約から、プロジェクト実施中に発生した工程遅延に対して、その影響がプロジェクト全体にどのように波及するかを指摘する [1]。さらに、

遅れが及ぶ作業に対してリソースの置き換えによりリスクが回避できる場合には、代替案の提示を行う。

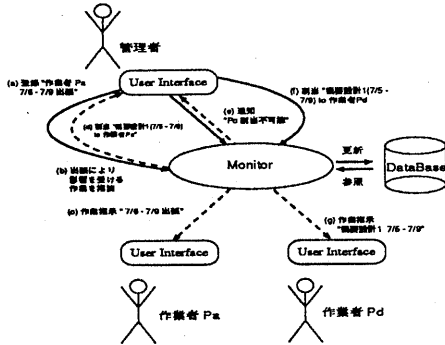


図 7: リソースの制約情報の変動による振舞い

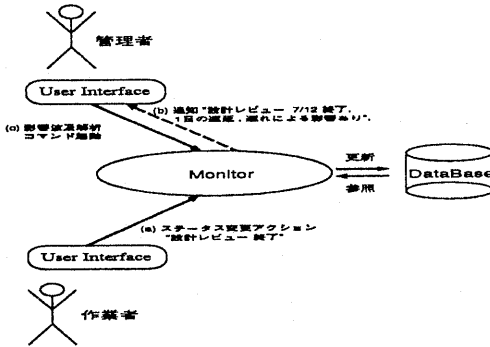


図 8: 作業終了時の振舞い

6 おわりに

本稿では、プロジェクト管理作業を支援するために、プロジェクト管理に関連して行われる作業を分析した。分析に際して、特にプロジェクト管理は管理者と作業者の協調作業によって進められるという視点を主眼に置いた。その結果としてプロジェクト管理においては、管理者と作業者のコミュニケーションが多くあり、それを支援することがひとつの重要な要素であると考え、計算機支援のための枠組を考えたい。

- コミュニケーション支援のためのアーキテクチャとして ECA の枠組が必要であることがわかった。
- 分析に基づき、ひとつの作業に対してプロジェクト管理と言う側面から起動されるアクションとそれにより遷移する状態と状態遷移モデルとして提案した。

- 各アクションに対応する情報の通知先は、これまでに提案したプロセスモデルの作業間の関連、作業とリソースの関連を利用すれば良いことがわかった。

本システムは UNIX ワークステーション上でオブジェクトデータベース Versant[7] と ART-IM[6] を用いて、一部の機能を除いて稼働している。

謝辞

本稿作成にあたり、熱心に議論して下さいました当プロジェクトのワーキング委員会の方に感謝致します。また日本電気株式会社マイコンソフトウェア開発研究所環境システム開発部の小山田正史課長からいただきましたコメントは有益でした。さらにニチメンデータシステム株式会社 大藤 倫昭課長、有原 浩司氏はシステムの実装を手伝って下さいました。記して謝意を表します。

参考文献

- [1] Hazeyama, A. and Komiya, S., A Process Model for Software Process Management, Proceedings of the Fourth International Conference on Software Engineering and Knowledge Engineering (SEKE'92), June 15 - 20 1992, pp.582-589.
- [2] Dowson, M. (Ed.), Proceedings of the First International Conference on the Software Process - Manufacturing complex Systems-, IEEE, October 1991.
- [3] Mi, P. and Scacchi, W., Process Integration in CASE Environments, IEEE Software, March 1992, pp. 45-53.
- [4] Ould, M.A., and Roberts, C., Defining formal Models of the Software Development Process, In Software Engineering Environments, Ellis Horwood Ltd, 1988, pp.13-26.
- [5] Ross, D.T., Structured Analysis(SA) for Communicating Ideas, IEEE Transactions on Software Engineering, Vol.3, No.1, 1977, pp.16-34.
- [6] The ART-IM User's Manual, Inference Corporation, 1990.
- [7] Versant System Reference Manual, Versant Object Technology, January 1992.
- [8] Syntagma, EAST Introduction, Edition 1, 1992.
- [9] Process Weaver General Information Manual, Version PW1.0, Cap Gemini Innovation, 1992.