

TM-0273

プロジェクト管理
エキスパートシステム

松本 均, 市夾宏基
佐藤秀樹(富士通)

February, 1987

©1987, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

プロジェクト管理エキスパートシステム

PROMX: Project Management Expert System

松本 均 市来宏基 佐藤秀樹

Hitoshi MATSUMOTO Hiroki ICHIKI Hideki SATO

富士通株式会社
FUJITSU Limited

1. はじめに

我々は、ビジネス分野に対する人工知能技術の応用として、プロジェクトの計画/管理を知的に支援するシステムの研究・開発を行っている。一般にプロジェクトとは、一定の目標に対して、関係者・関係部門が調整を取りながら、目標を効果的に達成するために行う仕事を指している。こうしたプロジェクトとしては、宇宙開発、プラント建設、出店計画、ハードウェア/ソフトウェア・システム開発等を始めとして、枚挙にいとまがない。近年、プロジェクトは大規模化・複雑化、不確定要因の増大、変更要因の増大の傾向にある。このようにプロジェクトの計画と管理は、益々困難になってきている。しかし、新製品の開発等にみられる様に、プロジェクトを短期間かつ確実に期日までに完了することは、ビジネス上重要な意味をもつようになってきている。

プロジェクト管理者は、分野の専門的知識と共に過去の経験的知識を基に、プロジェクトの計画立案/変更や進捗状況の把握を行っている。しかし、プロジェクトの管理に対する従来の計算機支援は、OR手法（例えば、PERT/CPM等）のパッケージ、あるいは各種データの統計処理/グラフ処理等のデータ処理機能が個別に提供され、管理者が目的に応じてこれらのツールを利用するといった状況である。

一方、近年、人工知能の技術を応用したエキスパート・システムの研究・開発が、特に、エンジニアリング分野、医療分野等を中心として盛んに行われている。我々は、このような背景から、プロジェクトの計画/管理における、

(1) プロジェクトの作業計画立案

(2) 資源の割当て等を考慮したスケジュール作成

(3) プロジェクトの実施設階における進捗管理

といった計画段階から実施設階に至る各種業務を、知的に支援するプロジェクト管理エキスパート・システム（PROMX: PRoject Management expert system）を研究・開発中である。また、PROMXでは、利用者に使い易い高度なインタフェース機能を提供することを目指し、グラフィックスを中心とした知的インタフェースを開発している。

このような知的なプロジェクト管理システムを実現するために、我々はプロジェクトを構成する作業を表すアクティビティの概念モデルを中心とした知識ベース・アプローチを採用する。本稿では、以下、PROMXのシステム概要、および、アクティビティの概念モデル等の知識表現について議論する。また、このモデルを中心とした各種支援機能、および、利用者インタフェースについて説明する。各機能を実現するモジュールは、新世代コンピュータ開発機構が開発した逐次型推論マシンPSI上のプログラミング言語であるESP[1]を用いて、述語論理表現をベースにオブジェクト指向パラダイムのもとに実現している。

2. システム概要

2.1 知識ベース・アプローチ

プロジェクトの計画から実施設階に至るフェーズを統一的に支援するためには、計算機内部にプロジェクトを表現する

ことが必要である。プロジェクトは、一般に、多くの様々な作業から構成される。これらの作業には、総括的に表現される作業から、担当者が実際に遂行する具体的な作業が含まれる。更に、総括的な作業は、より具体的な作業に展開されるといった作業間の構成関係がある。また、作業間には、一般に、先行/後続作業を規定する順序関係が制約として存在する。このような特性をもつ作業（アクティビティと呼ぶ）、および、それらの間の関係を表現する概念モデルを知識ベースの中核とし、この概念モデルから特定プロジェクトを表すアクティビティ・モデルを作成/操作することにより、プロジェクトの計画から管理に至る一連の業務を支援するという知識ベース・アプローチを採用する。図1に示すように、概念モデルはプロジェクトを構成するアクティビティ、および、その関係を表現する。図1の例では、ソフトウェア開発分野に関する作業の一部が表現されている。

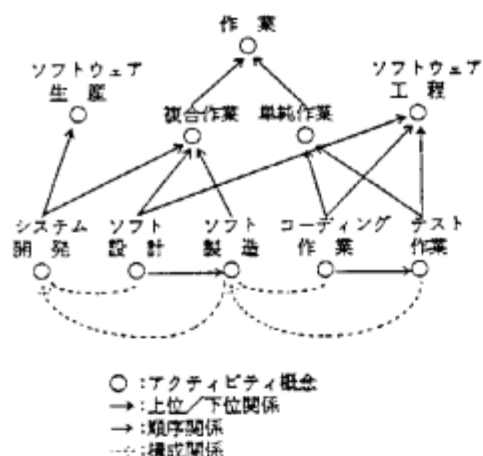


図1. 概念モデルの例

このように概念モデルの基には、対象分野における典型的な作業群が表現される。この際、抽象化階層を利用し、プロジェクト・ドメイン共通の概念とプロジェクトが対象とする分野固有の概念を分離し、種々の分野のプロジェクトに対応することができる構成、即ち、分野移行性が必要である。このような対象分野の典型的な概念モデルから、特定プロジェクトをインスタンス関係として個別に規定し、また、アクティビティ間の各種関係知識を利用して、特定プロジェクトの作業群の構成/順序リンクを表すアクティビティ・モデルを作成/利用する。

2. 2 システム支援機能

我々は、プロジェクトの計画段階として、プラン生成支援機能およびスケジューリング機能を提供する。また、実施段階として、進捗管理を行うモニタリング機能を提供する。以下、各支援機能の概略を示す。

(1) プラン生成支援機能

プロジェクトの目標を実現するために、アクティビティの集合からなるプランの作成を支援する。対象分野における概念モデル、即ち、典型的なアクティビティ知識、から特定プロジェクトのアクティビティ構成をインスタンス・トリー（アクティビティ・モデルと呼ぶ）として、段階的に構築するグラフィック・エディタを提供する。また、各アクティビティの属性編集機能や制約知識に基づくチェック機能を提供する。

(2) スケジューリング機能

プラン生成支援のもとで作成されたアクティビティ・モデルに対して、概念モデルの関係知識を利用することにより、具体的な作業を表すアクティビティ間のネットワークをアクティビティ・モデルに作成する。このアクティビティ・ネットワークを用いて、各アクティビティの開始日/終了日、および、資源の割当てを考慮したスケジュールを作成する機能を提供する。時間の計算は、知識としてPERT等のOR手法を利用する。また、資源の割当ては、経験的評価関数等の知識を用いたヒューリスティック探索法により実現する。

(3) モニタリング機能

プロジェクトの実施段階における作業の進捗管理を行う。各アクティビティの進捗データの管理、および、経験的知識に基づいて遅延作業等の問題点を検出、対処案を提示する機能を提供する。問題点の検出/対処のための知識の枠組みとして、問題を種々の観点から階層的に分類する。この問題状況を表す階層トリーを探索することにより、アクティビティやプロジェクトの診断に利用する。更に、プロジェクトの状況に対する問い合わせや必要に応じて再スケジューリングを行う。

また、利用者インタフェースとして、以上述べた各支援機

能の起動やプラン、アクティビティ・ネットワーク、ガント・チャート（開発工程観表）、および、進捗状況グラフの表示／操作を対話的に行うグラフィカル・インタフェース機能を提供する。

2. 3 システム構成

PROMXのシステム構成を図2に示す。前節で述べた各支援機能に対応したモジュール、および、概念モデルや診断、スケジュール評価等の経験的知識からなる知識ベースから構成される。

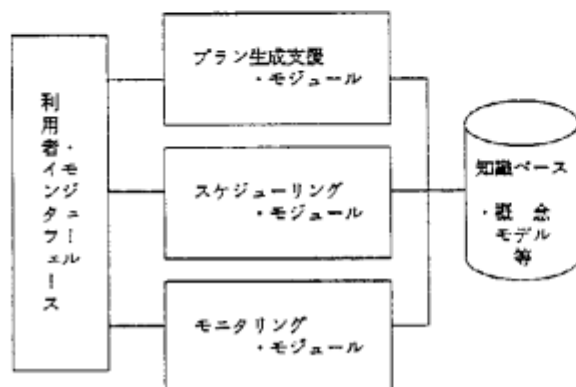


図2. システム構成

3. 対象世界モデル

プロジェクトの計画／管理業務を支援するためには、対象とする特定プロジェクトを計算機上に表現することが必要である。そのために、プロジェクトの分野に共通な概念をモデル化することから始める。

プロジェクトという問題領域に対して、作業を表すアクティビティ概念、アクティビティの開始／終了状態を表すステート概念、および、過去の典型的なプロジェクトを構成するアクティビティ群等からなる概念モデルを考える。概念モデルは、オブジェクト指向パラダイムでいうクラスオブジェクト中心の表現になっている。また、オブジェクト間、即ち、アクティビティ間、ステート間、および、アクティビティとステート間、の各関係知識は述語論理表現を用いる。そして、特定プロジェクトは、概念モデルから生成されるインスタンスオブジェクト、および、その関係として表現される。これをアクティビティ・モデルと呼ぶ。以下、これらのモデル、

および、各種関係知識について述べる。

3. 1 概念モデル

(1) 基本概念

プロジェクト問題領域には、プロジェクトを構成する作業を表すアクティビティ概念と作業の開始／終了時に成立する事実を表すステート概念がある。これらの概念は、それぞれ抽象階層のもとに表現される。例えば、作業に関して言えば、アクティビティは複合アクティビティと単純アクティビティに分類される。複合アクティビティは構成作業をもつ総括的な作業を表し、更に作業の組合せの種類により AND/OR アクティビティとして分類される。単純アクティビティは具体的な作業を表す。このように、アクティビティとステートの概念関係は、クラスオブジェクトの上位／下位関係として表現される。つまり、上位オブジェクトで規定される属性／関係は下位オブジェクトに遺伝される。

さて、プロジェクト業務、特にエンジニアリング・プロジェクト関係においては、何らかの製品（プロダクトと呼ぶ）の開発が行われる。例えば、ソフトウェア開発におけるプロダクトはソフトウェア・システムである。また、ソフトウェア開発においては、設計／製造／試験等といった開発工程（プロセスと呼ぶ）を表す作業がある。図3に示すように、このような2つの観点、プロダクトとプロセス、のもとにアクティビティを分類・整理し、これらをアクティビティ概念のもとに結合する。つまり、プロダクトの階層は対象プロダクトの機能といった観点であり、また、プロセスの階層は開発作業といった観点での分類である。

このように、対象分野の典型的なプロジェクトを構成するアクティビティをプロセス／プロダクトの概念のもとに分類する。我々は、図3に示すように、具体的なプロジェクトとして交換機ソフトウェア・システム開発を対象とした。

これらの個々の概念は、オブジェクト指向パラダイムにおけるクラスオブジェクトとして表現される。

(2) 関係知識

概念間、即ち、アクティビティ間、ステート間、アクティビティとステート間には、上述した上位／下位関係の他に各種関係が存在する。これらの関係には、以下のものがある。

- ③ 全体／部分関係と見なされるアクティビティ間およびステート間の構成関係：例えば、「開発作業は設計、製造、試験作業からなる」「交換機ソフトウェア・システム開発は、システム基本部開発と交換処理部開発からなる」といったアクティビティの構成を表す知識（sub_activity知識と呼ぶ）である。
- ④ アクティビティ間の先行／後続アクティビティを規定する順序関係：例えば、「接続制御試験を行った後、送受信制御試験と課金制御試験を行う」「基本設計を行い、次に機能設計を行い、次に詳細設計を行う」といったアクティビティの順序制約を表す知識（before_after知識と呼ぶ）である。
- ⑤ アクティビティとその開始時／終了時に成立する事実を関係づけるアクティビティとステート間の起動／生成関係：起動関係とは、「設計仕様書が存在すれば、製造作業を行うことができる」といったアクティビティとその実行可能状態との間の関係を表す知識（enable知識と呼ぶ）である。また、生成関係とは、「製造作業により、コーディング・モジュールが生成される」といったアクティビティと実行結果状態との間の関係を表す知識（cause知識と呼ぶ）である。

ぶ）である。

- ④ 状態の因果を表すステート間の因果関係：例えば、「製造完了状態であれば、試験開始状態である」といった生成ステートと起動ステート間の因果を表す知識（cause_enable知識と呼ぶ）である。

これらの4種類のオブジェクト間の関係知識は、述語論理表現によりモデル化される。図4にアクティビティの構成関係の知識表現例を示す。その他の関係知識も同様な表現で記述される。

例) 開発は設計／製造／試験から構成される。

```
sub_activity($ 開発, $ 設計) :- (条件);
sub_activity($ 開発, $ 製造) :- (条件);
sub_activity($ 開発, $ 試験) :- (条件);
```

\$XX: クラスオブジェクト

図4. 構成関係の述語論理表現

このように概念モデルとは、図3に示すように、プロジェクトという問題領域における基本概念、および、典型的なアクティビティ群をオブジェクトとして表現し、そのオブジェクト間の関係を述語表現により規定するものである。

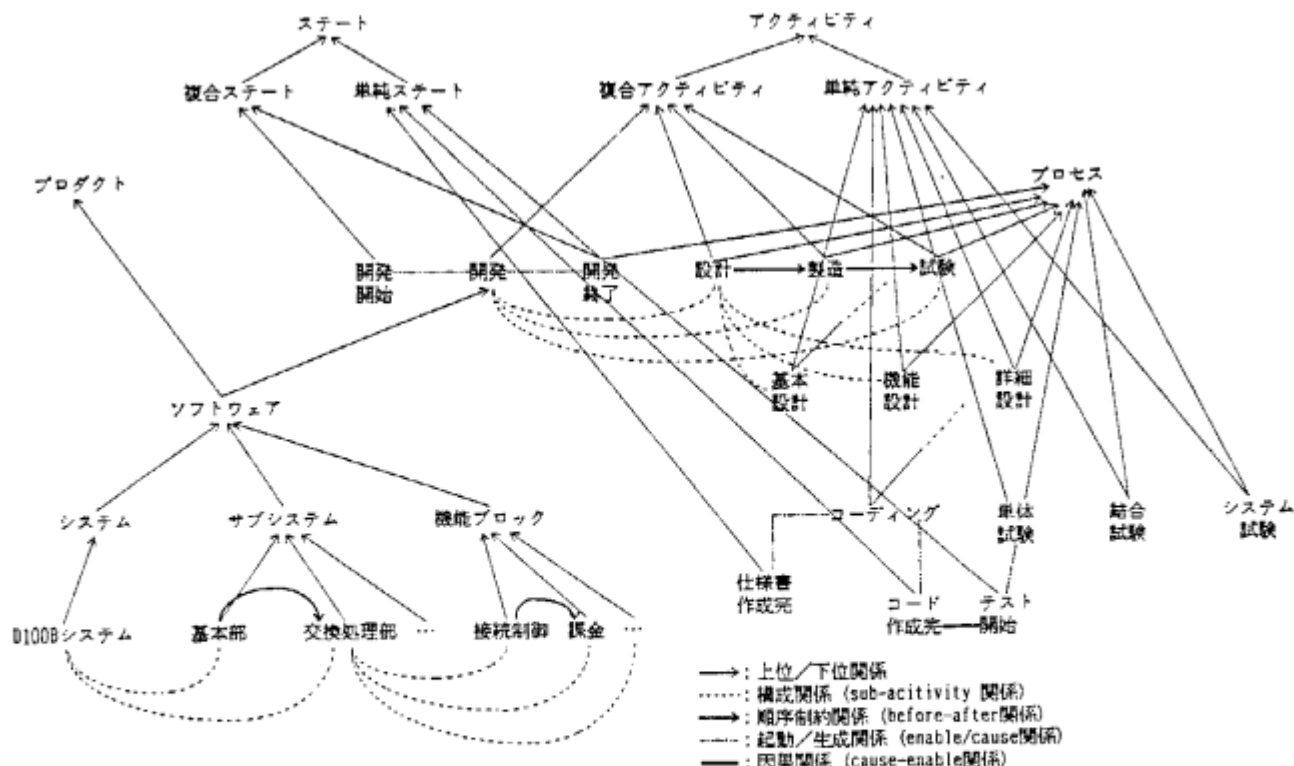


図3. 交換機ソフトウェア・システム開発プロジェクトの概念モデル

3. 2 アクティビティ・モデル

対象とする特定プロジェクトは、前節で述べた概念モデルのインスタンス・ネットワークとして表現される。これをアクティビティ・モデルと呼ぶ。このアクティビティ・モデルは、後述するプラン生成支援機能、スケジューリング機能のもとで生成され、また、各支援機能で利用される。

図5にアクティビティ・モデルの例を示す。特定プロジェクトのアクティビティ・モデルは、プロジェクトを示す最上位アクティビティから順に詳細化されたアクティビティの階層構造をもつ。つまり、詳細なレベルで全てのアクティビティが規定されるというより、むしろ階層の各レベルで規定される。このアクティビティ・モデルは、総括的なアクティビティから順に具体的なアクティビティに展開された、いわゆるWBS (Work Break-down Structure)を形成する。

アクティビティ・モデルの形成の際には、概念モデルの知識である sub_activity 知識が利用され、アクティビティを表すインスタンスオブジェクトの構成に関する双方向リンク (sub_activity_of / has_sub_activity) が張られる。また、アクティビティ間の before_after 知識、ステート間の cause_enable 知識、および、このアクティビティ構成を表す階層構造を利用した推論により、アクティビティの順序に関する双方向リンク (next_activity / previous_activity) が張られる。

また、個々のアクティビティ・インスタンスは、作業期間、利用資源、指定開始日/指定終了日等の制約条件を含む各種属性をもつ。



図5. アクティビティ・モデルの例

4. 支援機能

プロジェクト計画/管理システムには、2. 2節で述べたように、計画段階の業務を支援するプラン生成支援機能およびスケジューリング機能、また、実施段階の業務を支援するモニタリング機能がある。これらの各機能は、前述したアクティビティ・モデルの生成/利用を基本とする。

プロジェクトの計画から実施に至るシステムの処理フロー概要を図6に示す。

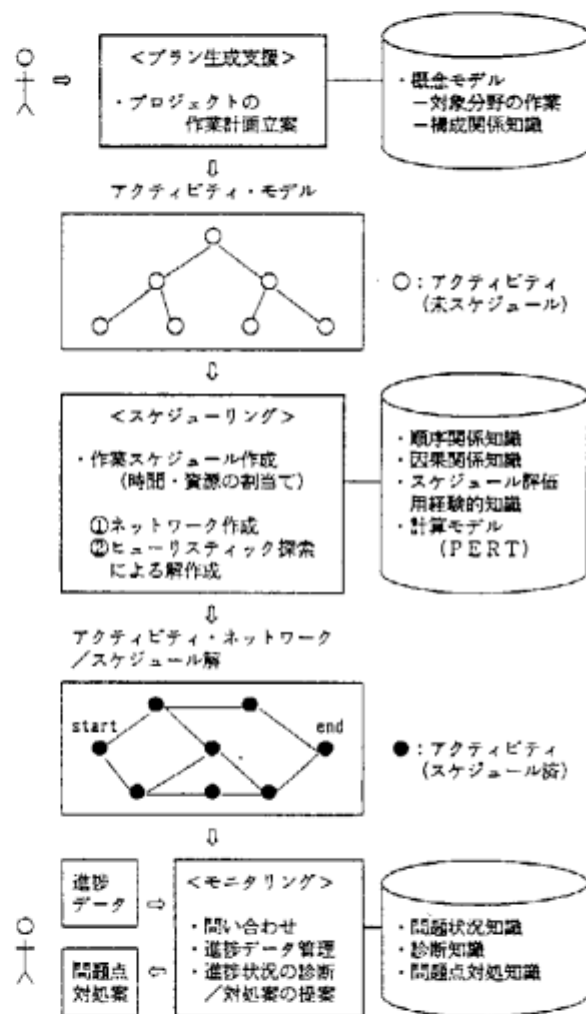


図6. 処理フロー概要

図に示すように、計画段階において、利用者は、プラン生成支援機能のもとで対象プロジェクトのアクティビティ・モデルを対話的に計算機上に作成する。次に、スケジューリング機能のもとでアクティビティ・ネットワークが作成され、この順序制約のもとで時間/資源の割当てを考慮したスケジュール解が作成される。利用者は、システムが生成した複数

のスケジュール解から1つの解を作業スケジュールとして設定する。また、実施段階のモニタリング機能のもとでは、問い合わせ、および、進捗報告データから各アクティビティやプロジェクトの進捗状況を診断し、対処案の提示を行う。

4. 1 プラン生成支援機能

プロジェクトの計画段階は、プロジェクトの目的から始まり、目的達成のために必要となるアクティビティの列挙を行うことから始まる。プロジェクト管理者は、典型的なプロジェクトや過去の実施したプロジェクトのアクティビティ構成を経験的知識として利用している。また、この段階では、アクティビティ間の順序関係は考慮されていない。

プラン生成支援機能では、対象分野に関するアクティビティの概念モデルから特定プロジェクトのアクティビティ・モデルの作成機能、アクティビティ属性の編集機能、および、アクティビティ構成/属性の診断機能を提供する。

利用者は、アクティビティ・モデルの作成/編集にグラフィック・エディタを利用する。

(1) アクティビティ・モデルの作成

我々は、現在、交換機ソフトウェア・システムの開発をプロジェクトの対象分野としてプロトタイプ・システムを開発している。

交換機ソフトウェア・システムの概念モデルは、プログラムの観点からシステム開発、サブシステム開発、機能ブロック開発、更に、プロセスの観点から開発工程といったアクティビティ群にカテゴライズされる。このようにカテゴライズされたアクティビティ群をウィンドウ上にリストメニューとして表示する。利用者は、このメニューから必要なアクティビティを選択し、グラフィカル・ウィンドウ上にアイコンとして表示する。この際、生成メッセージが選択アクティビティ・クラスに送られ、そのインスタンスが生成される。更に、グラフィカル・ウィンドウ上に表示されたアイコンに対して、「展開」の操作を行うことにより、対象アクティビティが構成アクティビティに展開され、アイコンとして表示される。対象アクティビティに対して、展開メッセージが送られる時、対象アクティビティは構成関係知識(sub_activity知識)を利用し、構成アクティビティを生成(インスタンス化)することにより、サブアクティビティとしてリンク関係(has_sub_activity / sub_activity_of)を張る。同時に、各サブアクティビティに展開メッセージを送ることにより、段階的にアクティビティ群のインスタンス・トリーとして、アクティビティ・モデルを作成していく。作成されたアクティビティ・モデルは、グラフィカル・ウィンドウ上にアイコンによるプランとして階層表示される。更に、「削除」「結合」等の操作により、プランの編集を行う。

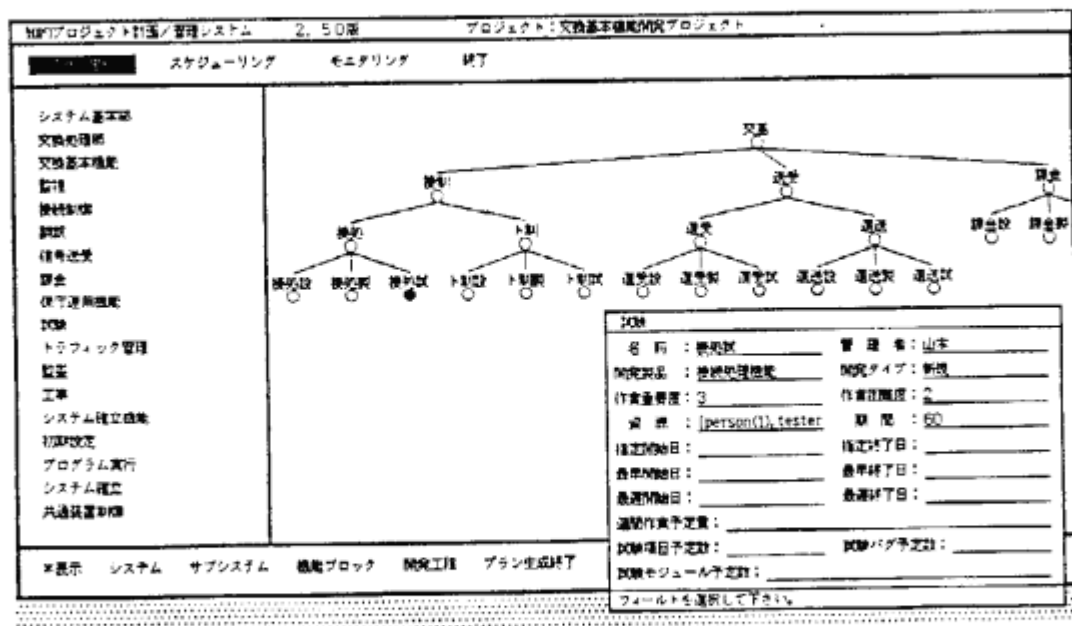


図7. 交換基本機能開発プロジェクトのプラン例

このように、利用者はグラフィカル・インタフェースを利用して、対話的に対象プロジェクトのアクティビティ・モデルを構築していく。

前頁の図7に示されるプランは、交換基本機能開発プロジェクトの画面出力例である。このプロジェクト（図中の交差）は接続制御サブシステム開発（接続）、信号送受サブシステム開発（送受）、課金サブシステム（課金）から構成される。また、いくつかのサブシステムは、機能ブロックの開発工程レベル（例えば、図中の接続設、ト制御等）まで展開された階層構造をもつ。

(2) アクティビティ属性の編集

プランを構成する各アクティビティに対して、図7に示すような属性入力／参照フォームを利用して、作業期間、利用資源等の各種属性を対話的に編集する。これらの属性は、アクティビティ・モデル中の対応するインスタンス・オブジェクトのスロット情報として設定される。これらの情報は、次節で述べるスケジューリング機能において、制約条件として利用される。

(3) アクティビティ構成／属性の診断

作成されたプランに対する診断機能として、各種構成上の経験的知識を利用する。例えば、アクティビティの構成上の制約として、「構成アクティビティの作業期間はその親アクティビティの作業期間より短い」「構成上にループが発生してはならない」「接続制御機能の開発は信号送受機能の開発が必要である」といった経験的知識をもとに診断を行う。

このようにして、構成上矛盾のないアクティビティ・モデルの作成を支援する。

4. 2 スケジューリング機能

プロジェクトの計画段階において、作成されたプランのアクティビティ群に対して、開始日時／終了日時、および、各アクティビティの遂行に必要な資源の割当てを行った作業スケジュールを作成する必要がある。

従来、スケジューリングのための技術として、PERT／CPM手法〔2〕が利用されている。PERT／CPM手法では、スケジューリングの対象となるアクティビティをネッ

トワークでモデル化し、このネットワーク上での経路計算により時間割当てを行う。しかし、このネットワークの入力や各アクティビティの要求する資源の割当ては、経験的知識を基に人手により行っている。

PROMXのスケジューリング機能では、前節で述べたプラン、即ち、アクティビティ・モデルを入力とする。そして、対象分野の概念モデルの順序関係知識、因果関係知識等を利用したネットワークの自動生成機能、更に、PERT手法や経験的な評価知識を用いて、時間や資源割当てを行うスケジューリング機能を提供する。

(1) アクティビティ・ネットワークの作成

スケジューリング機能の第1ステップとして、アクティビティ・モデル上の具体的な作業を表す最下位アクティビティ間に順序関係を表すネットワークを作成する。

- ① まず、各アクティビティの起動／生成スタート間に、概念モデルの因果関係知識（cause_enable知識）からリンクを張り、相当するスタートに関連するアクティビティ間に順序関係を規定する。

例えば、図8に示すアクティビティ・モデルに対して、

```
:cause_enable( $ 設計完了, $ 製造開始 );  
:cause_enable( $ 製造完了, $ 試験開始 );
```

といった開発工程の状態に関する因果知識から、接続設計アクティビティと接続製造アクティビティ間等に順序関係を規定する。

- ② 次に、概念モデルのアクティビティ間の順序制約を表す知識（before_after知識）から、アクティビティ・モデル上の対応するアクティビティ間に順序関係を規定する。

例えば、次の順序制約があれば、

```
:before_after( $ 接続設計, $ 送受設計 );
```

接続アクティビティと送受アクティビティ間に設計に関して、順序関係が規定される。

- ③ 最後に、アクティビティ・モデル上の上位で規定されたアクティビティの順序関係を下位のアクティビティに対して伝播させる。

例えば、②で規定された接続アクティビティと送受アクティビティの順序関係は、それぞれの構成関係をもとに推論され、図8に示すような最下位アクティビティ間の順序

関係が規定される。

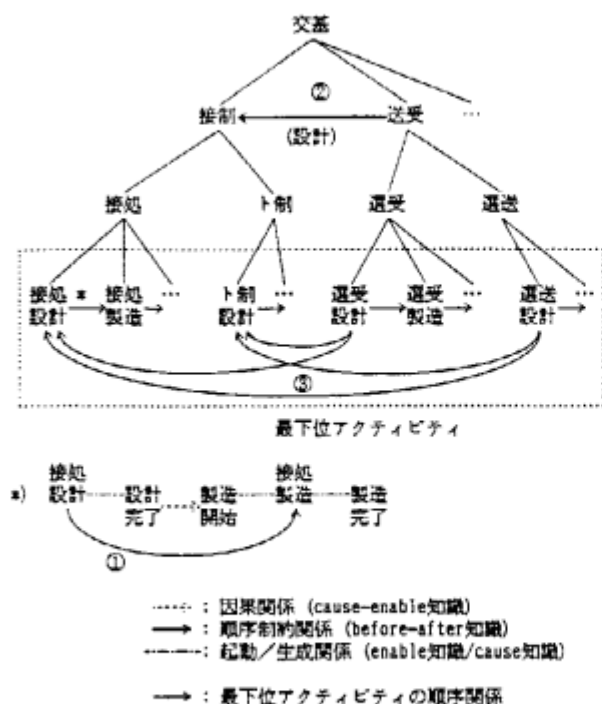


図8. アクティビティ順序関係作成手順

このように、最下位アクティビティ間の順序関係を規定したアクティビティ・ネットワークが作成される。この順序関係は、各アクティビティのスロット (next_activity / previous_activity) に双方向リンクとして設定される。図9に図7に示した交換基本機能開発プロジェクトのプランに対するアクティビティ・ネットワークの画面表示の例を示す。

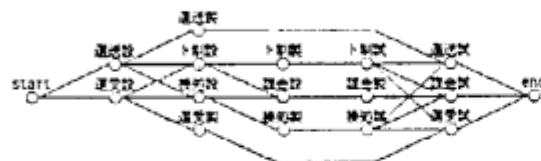


図9. アクティビティ・ネットワークの表示例

(2) 時間/資源スケジュールの作成

作成されたアクティビティ・ネットワーク上の各アクティビティに対して、時間および要求資源の割当てを行う。

我々は、スケジューリング・アルゴリズムとして、ネットワーク表現されたアクティビティの順序制約をもとに、経験的知識を利用したヒューリスティック探索手法を採用する。

探索トリーの各ノードは、スケジューリングの途中状態、および、その時点の資源状態を表現する。ノードの各種属性を図10に示す。

- 評価値：途中解に評価関数を用いた結果の値
- 資源割当て済みアクティビティ集合：途中解において資源が既に割当てられたアクティビティ群
- 資源割当て候補アクティビティ集合：次のスケジューリング段階で資源割当てが可能状態にあるアクティビティ群
- 資源未割当てアクティビティ集合：資源割当てが行われていないアクティビティ群。資源割当て候補アクティビティ群は含まれない。
- 資源状態：途中解における資源管理状態

図10. ノードの属性情報

スケジューリング解は、資源割当て済みアクティビティが全く存在しない開始ノードをルートとし、あるアクティビティに対して、資源割当てを行った途中解にあたるノードを子ノードとする探索トリーを順次展開していくことにより求められる。

ヒューリスティック探索によるスケジューリング・アルゴリズムの概略を図11に示す。本アルゴリズムでは、探索トリーにおけるリーフ・ノードをもつOPENリストと呼ぶデータ構造を用いる。

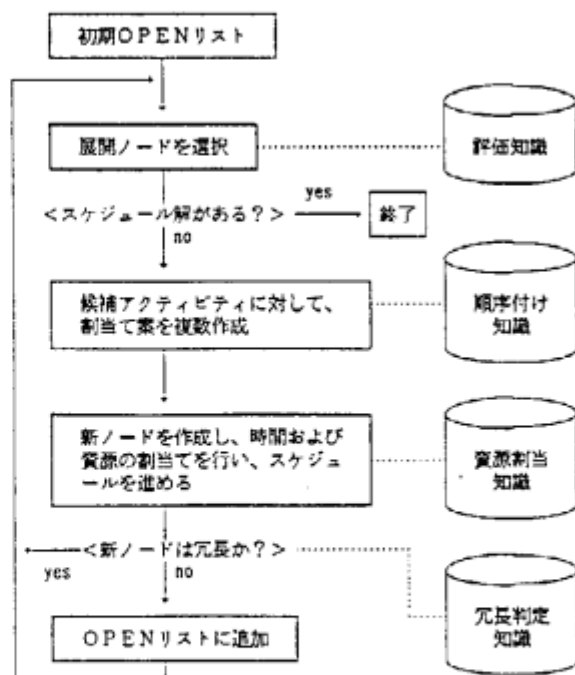


図11. スケジューリング・アルゴリズムの概略

① 展開ノードとして、OPENリスト中の評価値の一番良いノードを1個取り出す。そのノードに於て、全アクティビティが資源割当て済みならば、そのノードを解とする。

評価知識は、理想的な時間スケジュール（PERTによる値）と比較した遅延時間や割当て終了率等を用いる。

② 選択ノードの資源割当て候補アクティビティ集合に対して、どのアクティビティを候補にするかという複数の割当て案を作る。この各割当て案の順序付け知識として、重要度の高い順、要求資源の多い順、残り期間の長い順といった対象分野の経験的知識を利用することを検討している。プロトタイプ・システムでは、順序付けを考慮せず全ての割当て案を対象としている。

③ 各割当て案に対応して、新ノードを作成する。各ノードのもとで、以下の処理を行う。

(a) 対象アクティビティに要求資源を割り当て（資源割当て知識）、その結果、アクティビティの実行期間を決定する。資源割当てが不可能な場合、このノードをデッドノードとする。

(b) 次に、新しく候補となるアクティビティを調べる。つまり、資源未割当てアクティビティ集合のなかで、全ての先行アクティビティが資源割当て済みとなったアクティビティを新候補として、資源割当て候補アクティビティ集合に追加する。

(c) 資源割当て済みアクティビティ集合以外のアクティビティに対して、PERT手法により時間スケジュールを再計算し、各値を更新する。

(d) 各新ノードの冗長性をチェックし、冗長ノードをデッドノードとする。冗長性判定には、「対象ノードが同じレベル（ルートからの深さ）にあるノードに対して、資源割当て済みアクティビティの種類および決定時間が等しいならば冗長である」という知識を用いる。

⑤ デッドノードを除いた新ノードをOPENリストに追加する。

以上の操作を繰り返すことにより、複数のスケジュール解を作成することができる。利用者は、複数のスケジュール解のガントチャートを表示／参照することにより、1つの解を作業スケジュールとして設定する。図12に、図7に示した交換基本機能開発プロジェクトのプランに対するスケジュール結果、および、ガントチャートの出力例を示す。

4. 3 モニタリング機能

プロジェクトの実施段階では、プロジェクト管理者は、作業の各担当者からの進捗報告をもとにプロジェクトの進捗状況を管理する。この際、問題点の早期発見、また、それを解決する対策を早急に施すことがプロジェクトを成功させるための重要な仕事となる。

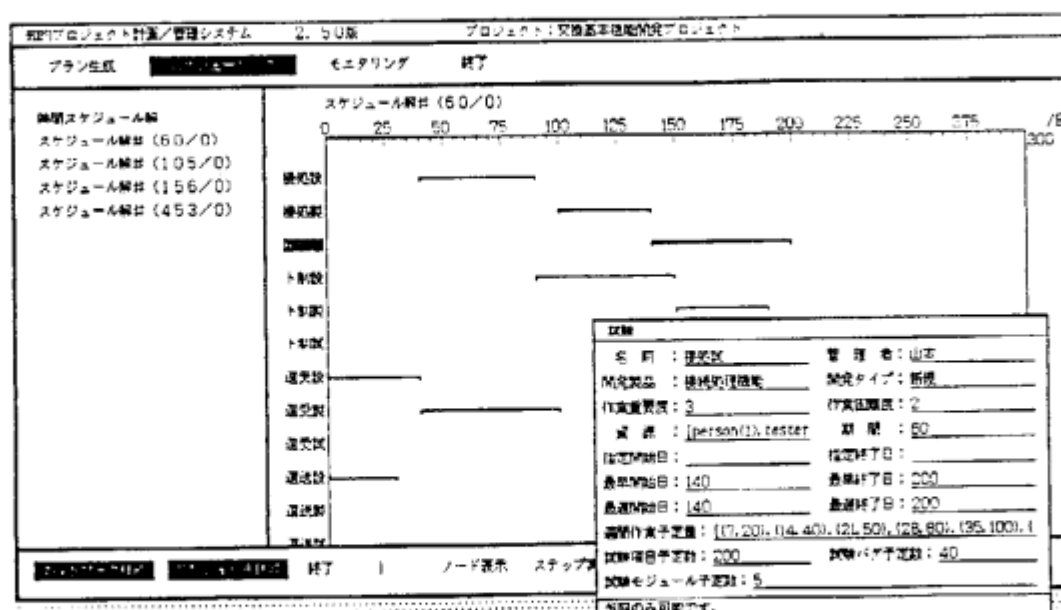


図12. スケジュール結果出力例

従来、プロジェクト管理の計算機支援は、データ統計処理パッケージやグラフ表示等の図形パッケージを提供するといった状況である。つまり、進捗状況に関する判断、問題点の検出／対処といった本来のプロジェクト管理業務は、経験的知識による面が大きく、人手に任されている。

PROMXのモニタリング機能は、アクティビティ情報等の問い合わせ機能、進捗データの管理機能、および、経験的知識を用いた進捗状況の診断／問題点の対処案の提示機能を提供する。これらの機能の実現は、アクティビティ・モデル、即ち、各構成アクティビティや各種関係を利用する。

(1) 問い合わせ機能

問い合わせ機能は、プロジェクトの進捗状況、開発製品の開発状況、資源の利用状況等の種々の観点からプロジェクト管理者が状況を把握することを支援する。

現在、プロジェクトの構成作業であるアクティビティに関する問い合わせを中心に検討している。アクティビティに関する問い合わせは、

- ① 個々のアクティビティの情報（属性／関係）に関する問い合わせ： 例えば、「接続処理設計アクティビティの予定開始／終了日時は？」「接続制御開発アクティビティを構成するアクティビティは？」等
- ② ある条件を満たすアクティビティを求める問い合わせ： 例えば、「クリティカル・パス上のアクティビティ

ィは？」「1週間以上遅れているアクティビティは？」

「現在、進行中のアクティビティは？」等

に分類される。

前者の問い合わせに対しては、アクティビティ・モデルからアクティビティの階層／ネットワークおよび開発工程（ガントチャート）をグラフィック表示する機能や各種属性をフォーム形式のもとに表示する機能で実現している。後者の問い合わせに対しては、アクティビティ・モデルが表す各種関係を利用して推論することが必要である。我々は、E S P言語[1]のメソッド述語により実現している。

(2) 進捗データの管理機能

フォーム・インタフェースのもとで報告される作業進捗データは、内部データ形式に変換され、アクティビティ・モデル中の対応アクティビティのスロットに設定される。また、図13に示すように、個々のアクティビティの進捗状況をグラフ表示する機能を提供する。この際、実行中のアクティビティに対しては、過去のデータからアクティビティの予想終了日を同時に表示する。図13の例では、課金設計アクティビティの設計書に関する週単位の予定量、現在までの実績量、および、アクティビティの予想終了日をグラフ表示している。

更に、データ管理機能として、インスタンス・ネットワークとして表現されているアクティビティ・モデルを二次記憶上に格納／復元する機能を実現している。

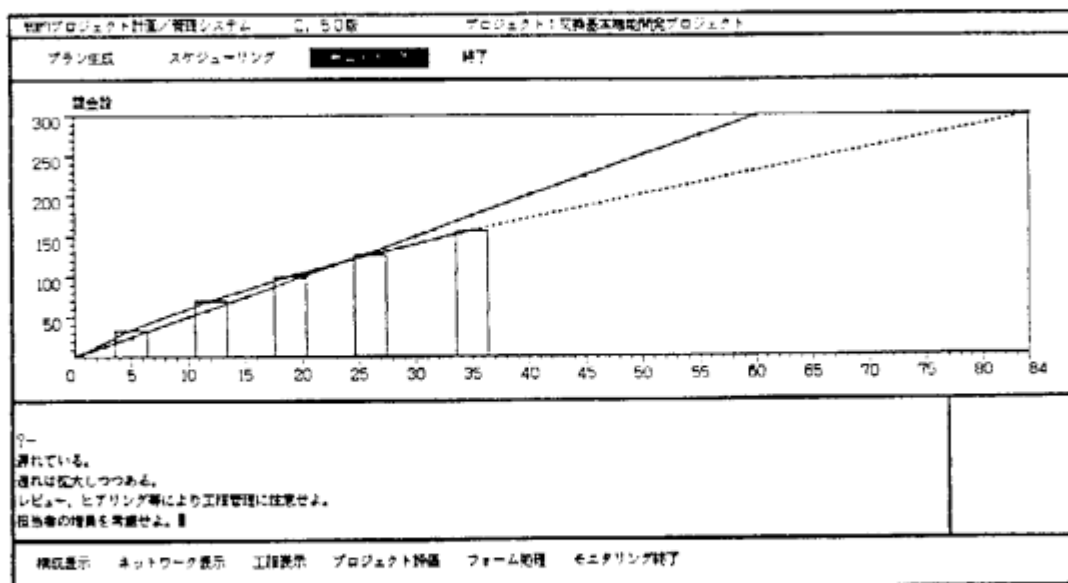


図13. 進捗状況のグラフ表示例

③ 進捗状況の診断／対処機能

プロジェクト管理者は、プロジェクトの状態を様々な観点からの経験的知識をもとに管理している。つまり、プロジェクトの問題発生を様々な角度から早期に検出しようとする。ソフトウェア開発プロジェクトにおける問題状況は、図14に示すような「工程上」「日程管理上」「品質上」といった観点で分類することができる。

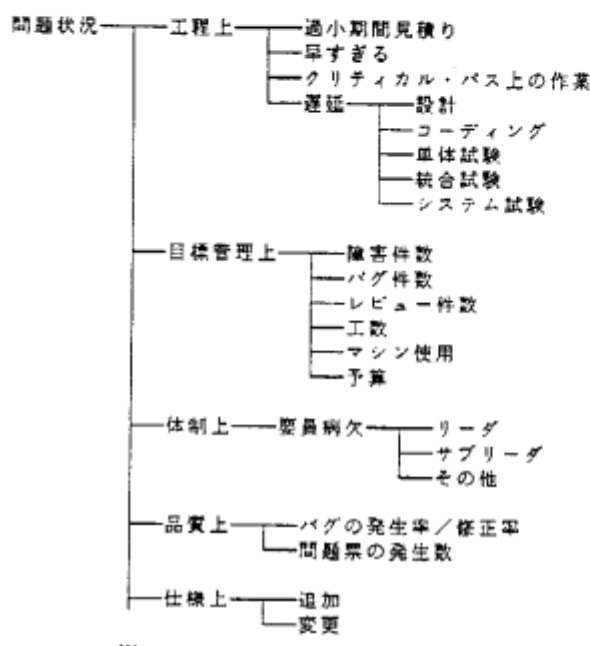


図14. 問題状況の分類

我々は、図14に示すような問題状況を、階層的に分類したツリー上のノードとしてモデル化し、計算機上にオブジェクトおよびその上位／下位関係として表現する。また、診断／対処に関する経験的知識は、この階層的に分類された問題状況オブジェクトのもとにルールとして表現される。

そして、進捗状況の診断プロセスを階層上のより具体的な問題状況を探す分類型問題解決プロセスと捉える。つまり、推論メカニズムは、上位の抽象的な問題状況オブジェクトから、下位の具体的な問題状況オブジェクトへトップダウンに遂行される。この際、対象のアクティビティの属性や関連オブジェクトをデータとし、各問題状況に適合するか否かを診断等のルールを利用して調べる。

図13に、課金設計アクティビティの進捗状況のグラフ表示と共に、システムが答えた診断／対処案に関するメッセージがウィンドウ上に表示されている。

4. 4 利用者インタフェース

PROMXの利用者インタフェースは、計算機上のアクティビティ・モデルとのインタフェースを与えるものである。各支援機能の節で示したように、インスタンス・ネットワークとして表現される対象プロジェクトのアクティビティ・モデルを図形（アイコン）により、表示／操作するグラフィカル・インタフェース機能を提供する。利用者は、アクティビティ・モデルを以下に示す様々な観点から、

- ① 構成を表すプラン
- ② 順序関係を表すアクティビティ・ネットワーク
- ③ 開発工程を表すガント・チャート
- ④ アクティビティ属性を表す入力／参照フォーム

表示／操作することができる。図15に利用者とアクティビティ・モデルのインタフェースの概念図を示す。

また、図形をアイコン・オブジェクトとして操作する基本機能やグラフィカル・ウィンドウ上での縮小、拡大、スクロール、印刷等の基本機能を提供している。

グラフィカル・インタフェースの設計指針や設計仕様は、本稿での主旨ではないので別の機会に報告したい。

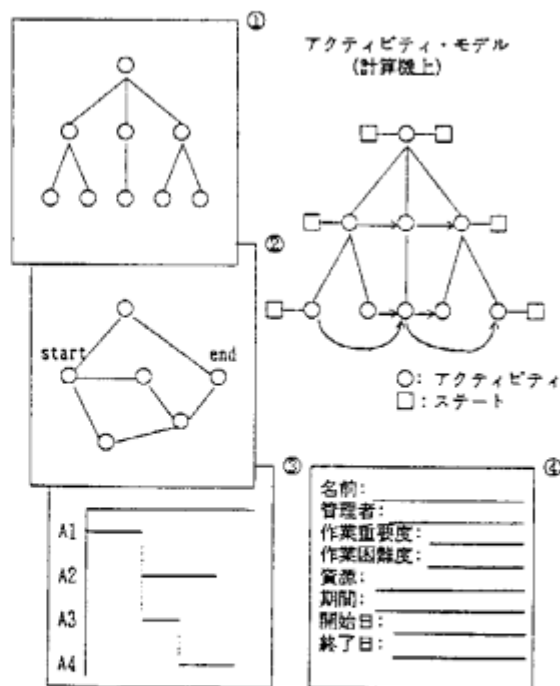


図15. アクティビティ・モデルとのインタフェース

5. おわりに

本稿では、プロジェクトの計画段階から実施段階に至る業務を知的に支援するプロジェクト管理エキスパート・システム(PROMX)について述べた。我々は、対象世界の概念モデルから特定プロジェクトのアクティビティ・モデルを計算機上に構築し、このモデルを中核とする知識ベース・アプローチを採用した。そして、プロジェクトの計画作成、スケジュール作成、進捗管理といった各種業務をこのモデルを中心に実現する方式、即ち

- ① プラン生成支援機能において、グラフィック・エディタを用いたアクティビティ・モデルの作成
 - ② スケジューリング機能において、アクティビティ・モデルに対するネットワーク/スケジュールの作成
 - ③ モニタリング機能の問い合わせ、進捗データ管理等において、アクティビティ・モデルの利用
- を提案した。

現在、各モジュールのプロトタイプ・システムが完成したところである。このプロトタイプ・システムをもとに、システムの各種支援機能の評価を行う予定である。例えば、スケジューリング機能で利用する評価知識は、経験的な計算モデル(評価関数)として断片的に規定しているが、「時間」「資源」等の様々な観点からの断片的な評価知識をどのように結合し、モデル化するかは今後の課題である。また、モニタリング機能における進捗状況の診断/対処に対して、問題状況の利用法に関する評価、考察を行っていきたい。

更に、システムを他分野のプロジェクトに適用することにより、システムの評価/改良、および、プロジェクト管理というドメインにおける分野移行性の検証を行う必要があると考える。

謝辞

本研究は、新世代コンピュータ技術開発機構(ICOT)の委託研究として行われた。本研究に対して御支援頂くICOTの岩下安男室長に深く感謝致します。また、日頃御指導頂く当社の林達也部門長付、平塚芳隆部長代、牧之内顕文部長代、および林弘部長代に感謝致します。最後に、有意義な御助言を頂いた坪井二郎氏に深く感謝致します。

【参考文献】

- [1] T. Chikayama, ESP Reference Manual, ICOT TR-044 (1984)
- [2] 加藤昭吉著「PERTの知識」日本経済新聞社 (1968)
- [3] 日本電信電話公社「D10 形自動交換機 [II] ソフトウェア」電気通信共済会 (1979)
- [4] 松本他「知的プロジェクト計画・管理システムー開発アプローチー」情報第33回全国大会 7Y-2 (1986)
- [5] 市来他「知的プロジェクト計画・管理システムー知識ベースに基づくスケジューリング方式ー」情報第33回全国大会 7Y-3 (1986)
- [6] H. Sato, Project Management Expert System, Proceedings of ACM CSC 87 (Feb. 1987)