

IDEF を用いたスケジューリングシステム開発プロセスのモデル化

阿部昭博†‡ 玉井哲雄‡

松下電器産業(株) マルチメディアシステム研究所† 東京大学大学院総合文化研究科‡

近年、生産および物流業務効率化の手段として、スケジューリングシステムの構築を試みる企業が増えている。しかし、実用システム構築のためのプロセスや方法論が明確になっておらず、現場の開発者は経験に頼ったシステム開発を行なっているのが実情である。本稿では、物流および生産業務における4つのスケジューリングシステム開発事例をとりあげて、スケジューリングシステム開発プロセスの分析・モデル化を行い、プロセスの特性について考察する。プロセスモデルの記述は、米国においてCAL Sを推進する際のビジネスプロセスのモデリングで標準的に利用されている手法の一つで、可読性に優れたIDEFを用いる。

Modeling of Scheduling System Development Processes Using IDEF

Akihiro ABE†‡ Tetsuo TAMAI‡

Multimedia Systems Research Laboratory, Matsushita Electric Industrial Co.,Ltd.†
4-5-15 Higashi-Shinagawa, Shinagawa-Ku, Tokyo 140, Japan
E-mail: abe@trl.mei.co.jp
Graduate School of Arts and Sciences, University of Tokyo‡
3-8-1 Komaba, Meguro-ku, Tokyo 153, Japan
E-mail: tamai@graco.c.u-tokyo.ac.jp

In recent years, many companies have considered constructing a scheduling system as a method for improving the efficiency of production and physical distribution businesses. However, the development processes and methodology to construct a practical system are still unclear, so field developers are developing the system, relying on their experience. In this paper, we show the four examples of scheduling system development in the production and physical distribution businesses and discuss their characteristics in regard to the software processes. The description of software processes uses IDEF that is one of the business process modeling technique for the realization of CALS.

1 はじめに

スケジューリングシステムは、複雑な生産あるいは物流の業務環境を理解した上で、大規模かつ種々の制約をもつ組合せ最適化問題（以下、スケジューリング問題）を解く必要があるため、ビジネスアプリケーションの中でも難易度の高いシステムに位置づけられている。しかし、生産および物流業務の複雑化にともない、スケジューリングシステムへの期待は高まっており、既に実用システム構築の事例も数多く報告されている [1][2]。

スケジューリング問題を扱う解法については、古くからOR分野において理論面からの研究が盛んに行われてきた。近年は、OR手法にAI手法なども併用し、実用に耐えうる近似解法の開発とその実問題への適用が進みつつある。

解法研究とは対照的に、スケジューリングシステムの開発プロセスや開発技術の研究は立ち遅れている。そのため、実用システムの構築は、それぞれの現場において段階的・試行錯誤的に行われているのが実状であり、開発現場から課題として指摘されている。筆者自ら携わった多くのシステム開発においても同様で、そのプロセスは不明確であり、管理項目も定義されており、経験的に管理している状態である。

これら課題への取り組みの重要性は最近になって認識されつつあり、FA (Factory Automation) やORの学会で議論され始めている。藤本らは [3]、実用システム事例30件の文献調査を行い、生産スケジューリングシステムを実用化に導くためには、スケジューリング問題の3レベルの難しさを認識した上で段階的に開発を進める必要があることを示唆した。

- 定義の難しさ：
工程が複雑で定式化が難しい
- 求解の難しさ：
制約が複雑で最適化が難しい
- 実現の難しさ：
構成が複雑で実用化が難しい

しかし、現在のところ、ソフトウェア工学に立脚した議論は特に行われていない。

ソフトウェア工学の研究領域では、ソフトウェアの開発プロセスに着目し、開発プロセスの改善を通して生産性、品質向上、開発期間の短縮を図ることを目的としたソフトウェアプロセス研究が着実に成果を結びつつある [4][5]。ソフトウェアプロセス研究は、その研究アプローチから

1. 実際のプロセスの観察、分析、改善 [6][7][8]
2. CMMに代表される規範のプロセスモデルの構築とその適用評価
3. プロセスの形式的記述
4. プロセスの実行環境に関する研究

に分類できると言われている。

本研究では1.のアプローチをとり、実際のスケジューリングシステム開発におけるソフトウェアプロセスをIDEFを用いて分析・モデル化し、その特性を明らかにする。

2 ソフトウェアプロセスの分析方針

本章では、ソフトウェアプロセスの分析における基本的な考え方、取りあげる事例の概要について述べる。

2.1 分析の基本的な考え方

目的 実用スケジューリングシステム開発におけるソフトウェアプロセスの特性を明らかにし、理解・共有化することを目的とする。これは、今後、スケジューリングシステム開発におけるプロセス改善、開発技術を議論するうえでの基盤となる。

視点 スケジューリングシステム開発のプロジェクトリーダーの視点で分析・モデル化する。その際、プロセスを実行する開発者・利用者・運用者の関わり、プロセスを実行する際のドメイン制約および開発管理方針の分析を特に重視する。分析の結果は、開発者のみならず利用者、運用者も理解可能な記述方法でモデル化する。

アプローチ 実運用システムあるいは運用可能レベルに到達したスケジューリングシステム4事例(生産2, 物流2)を取り上げ, 実際に行なわれたソフトウェアプロセスをモデル化し, 比較する. 分析の情報源としては, システム開発時に残された議事録, 設計メモ, 改善要望票などを用いる.

分析プロセス 1) 分析方針の設定, 2) 分析事例の整理, 3) 各事例の分析・モデル化, 4) 各モデルの比較, 5) ソフトウェアプロセスの特性把握 という手順で実施する.

モデリング言語 IDEF0[5][9][10]を用いて分析, モデル化を行なう. IDEF0(前身は構造化分析設計手法SADT)は, 米国においてCALSを推進する際のビジネスプロセスのモデリングで標準的に利用されているIDEF手法の一つで, 米国連邦情報処理標準(FIPS: Federal Information Processing Standards)に採用されている. IDEF0をモデリング言語として選定した理由は以下の3点である. 第1にモデルの実行はできないものの可読性が高く, ソフトウェア開発者以外でも十分理解できる. 第2にプロセスとそれに対応する入出力プロダクトの関係のみならず, プロセスに対する制御, 資源の関わりについても記述できる. 第3にIDEF0の適用に熱心なのは生産・物流業務プロセスのモデル化を試みている製造業であり, スケジューリングシステムの主要な適用領域と一致する. これらIDEFおよびIDEF0については3章にて詳しく説明する.

2.2 分析事例の概要

IDEF0による分析に先立ち, 対象とする開発事例の概要を表1のようにまとめた. システムの開発履歴については, プロトタイプおよび運用システムに分けたうえで, ソフトウェアのバージョンに着目して整理した.

- A工場の生産計画システム(事例1)[11]
従来人手で立案していた日程レベルの生

産計画をエキスパートシステム技術を用いて自動立案するシステム.

- B工場の生産計画システム(事例2)
A工場と同様に, 日程レベルの生産計画を自動立案するシステム. A工場向けのシステムをカスタマイズして開発.
- 配車計画システム(事例3)[12]
輸送ハブから複数デポまで放射状に広がる定期トラック路線を対象として, 輸送便への効率的な車両割当てを決定する.
- 輸送シミュレーションシステム(事例4)[12]
全国にまたがる広域物流ネットワーク上の集荷, 輸送, 配送を対象として, 輸配送に関するパラメータを操作した際の輸配送状況の変化をシミュレートする.

3 IDEF0によるソフトウェアプロセスの記述

本章では, IDEF0を用いたソフトウェアプロセスの記述方法について述べる. IDEF(Integrated computer aided manufacturing Definition)は, もともと米国空軍が1機種 of 航空機を発注する際, 発注仕様を標準化するために開発が始まったもので, 現在は, 組織の中で行われている様々な活動や情報の流れを可視化し, CALSを推進するためのモデリング手法として米国で普及しつつある. 手法は開発中を含めて16種類企画されており(表2), その概要と仕様については文献[9][10]で詳しく紹介されている.

IDEF0はダイアグラムと呼ばれる単位(図1~4)でモデルを表現する. ダイアグラムはシステムにおける機能・動作(activity)を一つのボックスで表し, 機能・動作に対する入力(input), 制約条件(control), 出力(output), 資源・設備(mechanism)を4本のアローで表す. 各ボックスは, ダイアグラムを用いて順次詳細化することができ, この階層的な記述により大規模なモデルも比較の見通し良く書ける.

本研究では, 開発プロセスをactivity, プロダクトをinput/output, 開発プロセスを実行す

表 1: スケジューリングシステム開発事例の概要

	A 工場の 生産計画システム (事例 1)	B 工場の 生産計画システム (事例 2)	配車計画システム (事例 3)	輸送シミュレー ションシステム (事例 4)
システムの 実用レベル	実運用	一定期間実運用後 使われなくなる	実運用	プロトタイプ (性能的には実用 レベル)
プロトタイプ記述 言語・ツール	AI ツール	AI ツール	スケジューリング ツール	OODB, C++
運用システム記述 言語・ツール	AI ツール	AI ツール	C	-
解法のアプローチ	AI 的手法	AI 的手法	OR 的手法	対話的手法
プロトタイプ のバージョン	対象問題明確化 (Ver1) UI 明確化 (Ver2) 解法設計 (Ver3) 解法性能評価 (Ver4)	対象問題明確化 (Ver1) 解法改良 (Ver2) 解法性能評価 (Ver3)	対象問題明確化 (Ver1) 解法設計 (Ver2) 解法性能評価 (Ver3)	対象問題明確化 (Ver1) 解法設計 (Ver2) DB モデル改善 (Ver3)
運用システム のバージョン	運用基本機能 (Ver1) 立案パラメタ機能 (Ver2) 生産変更機能 (Ver3) UI 機能拡張 (Ver4) システム再構築 (Ver5)	運用機能 (Ver1)	運用基本機能 (Ver1) 適用範囲拡大 (Ver2) 計画変更機能 (Ver3)	

表 2: IDEF ファミリー

IDEF0: Function Activity Modeling
IDEF1: Information Modeling
IDEF1X: Data Modeling
IDEF2: Dynamic Modeling
IDEF3: Process Description Capture
IDEF4: Object Oriented Design
IDEF5: Ontology Description Capture
IDEF6: Design Rational Capture
IDEF7: Information System Audit Method
IDEF8: Human-System Interaction Design
IDEF9: Business Constraint Discovery
IDEF10: Information Artifact Modeling
IDEF11: Implementation Architecture Modeling
IDEF12: Organization Design
IDEF13: 3-schema Architecture Design Modeling
IDEF14: Network Design

実際の制約であるドメイン制約と開発管理方針を control, スケジューリングシステム開発に参画する開発者・利用者・運用者を mechanism でそれぞれ表すこととする。図 1 は、これら分析に対する視点を表した最上位のダイアグラムでコンテキストダイアグラムと呼ばれる。

開発プロセスは、フェーズ、サブフェーズ、ステップといった 3 レベルまでを記述の範囲とする。フェーズは、実運用システムを構築するまでのライフサイクルの局面を大まかに区分したものを指す。各フェーズでは、バージョンの異なるソフトウェアプロダクトを開発する場合が

あるが、それに対応するプロセスをサブフェーズと呼ぶ。各サブフェーズは、複数のステップから構成される

A 工場生産計画システムの分析結果を例にダイアグラムの詳細化について述べる。システム化要求を満足した実運用システムを構築するまでのライフサイクルは、3つのフェーズ「システム化要求の明確化」「解法の開発」「運用システムの開発」に区分することができる(図 2)。「システム化要求の明確化」フェーズは、その目的を達成するために 2 回プロトタイプングを実施しており、これをサブフェーズ「対象スケジューリング問題の明確化」「ユーザインタフェースの明確化」としてモデル化している(図 3)。「対象スケジューリング問題の明確化」サブフェーズは、「プロトタイプ開発」「プロトタイプ評価」の 2 つのステップから構成される(図 4)。

開発プロセス(activity)を表すボックス間の接続方法は、a) プロセスの output が後プロセスの input, b) プロセスの output が後プロセスの control, c) プロセスの output が前プロセスの input, d) プロセスの output が前プロセスの control の 4 種類で表現している。プロセスの output が前・後プロセスの mechanism となる接続は、今回のモデリング

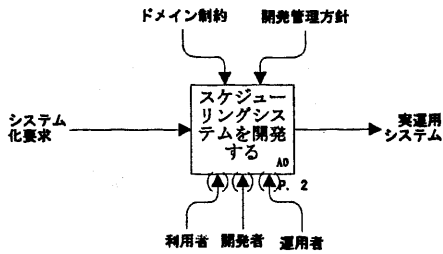


図1: コンテキストダイアグラム

プロトタイピングの開発管理方針
開発ステップの開発管理方針
全体開発計画として決定すべき事項を確認する
開発体制
スケジュール
開発環境
運用システムイメージ
利用イメージ
導入期待効果
評価項目
プロトタイピングの方針を確認する
プロトタイピングの実施範囲
プロトタイピングツール
プロトタイピングのスケジュール
プロトタイピングの開発担当者

図5: 開発管理方針に関するチェックリスト

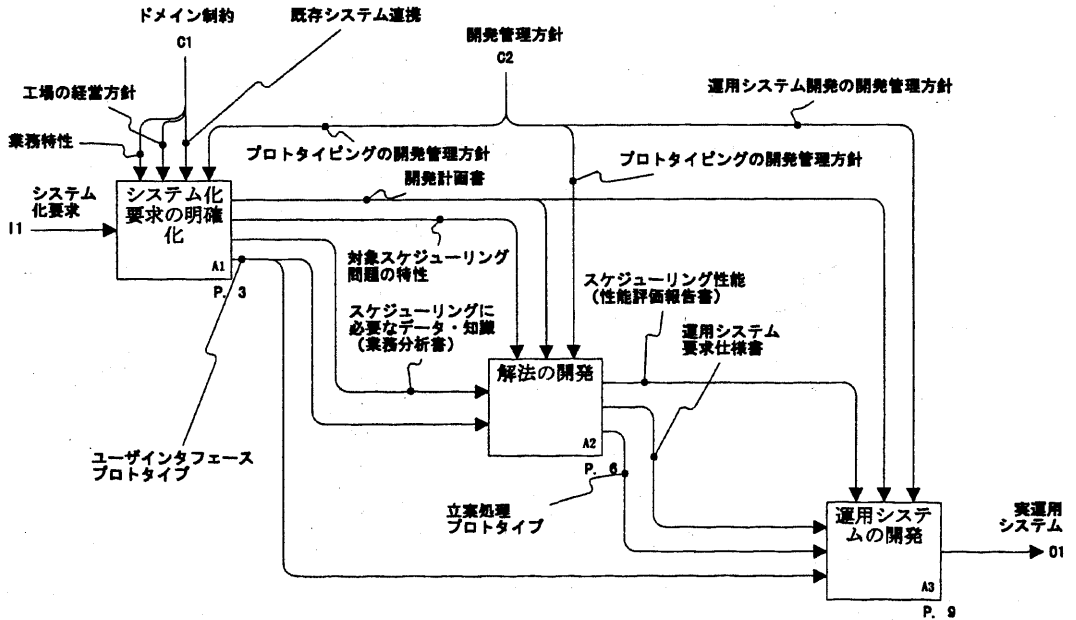


図2: フェーズを表すダイアグラム

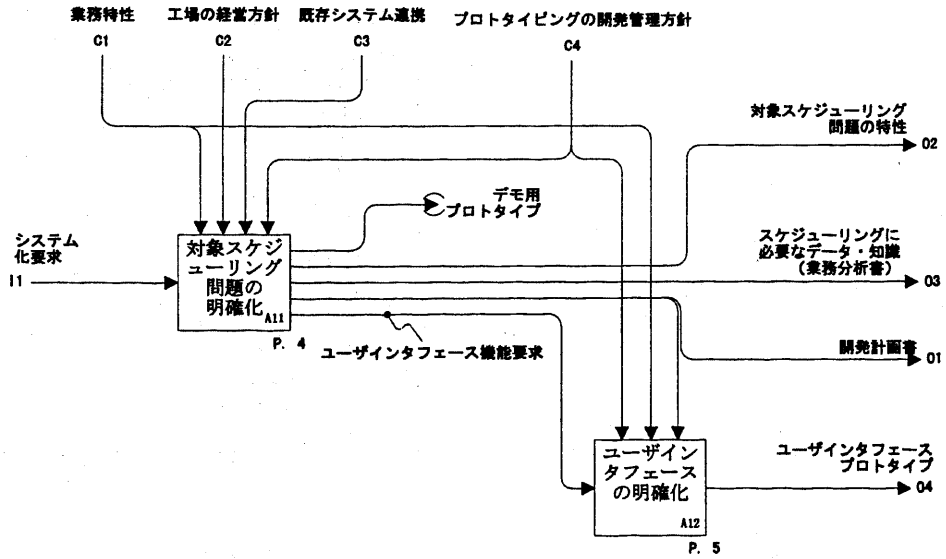


図3: フェーズA 1 (システム要求の明確化) の各サブフェーズを表すダイアグラム

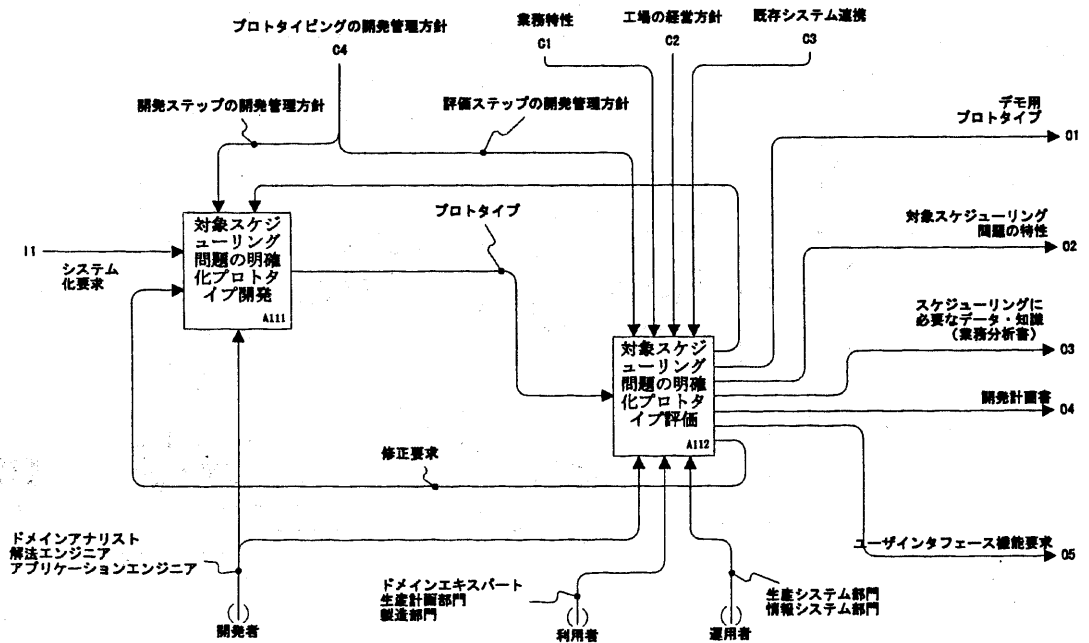


図4: サブフェーズA 1 1 (対象スケジューリング問題の明確化) の各ステップを表すダイアグラム

では存在しない。プロセスの output が自分自身の input や control となる接続は、下位のダイアグラムで表現している。

モデルの可読性の観点から、ダイアグラムの記述について幾つかの工夫を施している。プロセスに対する開発者・利用者・運用者の関わりは、ステップのレベルに対してのみ記述する。これは、IDEF0 のモデリング技法の一つで、特定階層のダイアグラムのみでアローを記述する、トンネル化定義を利用している。また、ドメイン制約および開発管理方針を表す control はチェックリスト表現を用いて補足する (図5)。

4 考察

4つの事例分析結果を比較し、ソフトウェアプロセスの特性について考察する。

システムライフサイクルとしての特徴 実用化の判断を行うまでは、システム化要求の明確化フェーズ、解法開発フェーズが存在し、それぞれのフェーズで数回のプロトタイプングを実施し、スケジューリング問題のもつ難しさを段階的に解決している。運用システムの仕様定義は、どのような解法のアプローチを採用するかに依存する部分が大きい。実用化の目処が立つまでは明確にならないことが多い。実用化決定後の運用システム開発フェーズでは、業務環境の変化に追従しながら運用にシステムを馴染ませてゆくため、スパイラル型開発が必要となる。

開発管理上の課題 課題は大きく4つ存在すると思われる。第1に、システム化要求の明確化フェーズ、解法開発フェーズにおいて、それぞれプロトタイプングの性格が異なるが、それを十分認識できていない。すなわち、仕様探索型と実験型の違いである。加えていずれのプロトタイプングにおいても、実施の目的、範囲、評価方法等について、プロトタイプの評価者である利用者と共有化できないまま開発を行っている。その結果、仕様追加や無駄な修正を引き起こし、プロトタイプングの作業工数の

増大を招いてしまっている。第2に、業務知識・ドメイン制約の理解が不十分なまま開発管理を実施しているため、作業の手戻りや不適切な評価を招くケースが見受けられる。第3に、運用システム開発において、運用をつづけながらシステムを拡張する際にソフトウェア構成管理、ドキュメント管理のための支援環境を組み込んだ開発管理が行われていない。第4に、利用者・運用者のみならず開発者自身がスケジューリングシステム開発の特徴を理解していないため、実運用までのプロセス全体を見通すことができていない。

ドメイン制約の特徴 「コスト削減のみならず、マーケットや顧客ニーズにできるだけフレキシブルにかつスピーディーに対応する」といった工場経営、企業経営の方針は、スケジューリングシステムの導入効果に直結しているため、実用システム構築において最も優先度の高い制約となっている。スケジューリングシステムは基幹情報システムと生産・物流システムの間位置し、かつその実現が難しかったため最後までシステム化されずに残っている場合が多い。そのため、既存システムとの連携は不可欠となっている。また、スケジューリング問題の難しさは対象業務の特性に大きく依存する。たとえば同じ会社の同じプラスチック部品成形工程でも、工場環境、製造する製品が違えば、異なったスケジューリングの解法が必要となる。

要員の参加 プロトタイプから運用システム開発まで、利用者・運用者・開発者の協力体制が不可欠である。特に、業務特性を考慮した解法の開発においては、利用者の参加が重要である。スケジューリングシステムは、基幹情報システムと生産システム、物流システムの間位置づけられるため、それを運用する部門は事例ごとに異なっているものの、運用で成功した例はどれも運用部門の強いリーダーシップが存在している。運用システムのスパイラル開発

は、運用者主導となるため、開発者は保守的な役割が強くなってゆくことを理解しておく必要がある。

5 おわりに

IDEF0で記述した4事例のモデルから標準プロセスを導くとするならば、

1. 仕様探査型プロトタイプングを実施するシステム要求明確化フェーズ
2. 実験型プロトタイプングを実施する解法開発フェーズ
3. スパイラル型開発を実施する運用システム開発フェーズ

を段階的に実施するプロセスが望ましいと思われる。

今後は、標準プロセス定義および開発技術の両面から、スケジューリングシステム開発の在り方について考察を進める予定である。標準プロセスの定義にあたっては、今回のIDEF0を使った記述に近い形を想定しているが、スケジューリングシステム開発に携わる利用者、開発者、運用者の意見を反映しながら、IDEF3の利用なども検討する予定である。

謝辞

本研究についてドメイン分析の観点から有益な助言を頂いた情報処理学会ソフトウェア工学研究会ドメイン分析WGのメンバー各位に深謝する。

参考文献

- [1] 宝崎隆祐, 藤井進: 物流システムにおけるスケジューリング, システム / 制御 / 情報, Vol.37, No.6, pp.344-349, 1993.
- [2] 黒田充: 生産スケジューリング研究の課題と展望, 生産スケジューリングシンポジウム'94 論文集, pp.1-13, 1994.
- [3] 藤本英雄他: スケジューリング問題・解法の複雑度とそのデータベース化, 生産スケジューリングシンポジウム'97 論文集, pp.233-242, 1997.
- [4] 落水浩一郎: ソフトウェアプロセスに関する研究の概要, 情報処理, Vol.36, No.5, pp.379-391, 1995.
- [5] Fuggetta, A. and Wolf, A.: Software Process, John Wiley & Sons, 1996. (岸田孝一監修: ソフトウェアプロセスのトレンド, 海文堂, 1997.)
- [6] Curtis, B., Krasner, H., and Iscoe, N.: A Field Study of the Software Development Process for Large Systems, Comm. ACM, Vol.31, No.31, pp.1268-1287, 1988.
- [7] Tamai, T. and Torimitsu, Y.: Software Lifetime and Its Evolution Process Over Generations, Proceedings of the Conference on Software Maintenance, pp.63-69, 1992.
- [8] Tamai, T. and Itou, A.: Requirements and Design Change in Large-Scale Software Development: Analysis from the Viewpoint of Process Backtracking, Proceedings of the 15th International Conference on Software Engineering, IEEE, pp.167-176, 1993.
- [9] 石橋: IDEF: BPA のための手法, オペレーションズリサーチ, Vol.42, No.4, pp.201-209, 1997.
- [10] Federal Information Processing Standards: IDEF0, FIPS183, <http://www.idef.com/>
- [11] Abe, A., Nagai, A., Ishiguro, M., Maeda, T. and Nakano, G.: I-CAPE: An Expert System for Production Scheduling in Plastic Molding Factory, Proceedings of the Japan-U.S.A. Symposium of Flexible Automation '92, pp. 1465-1472, 1992.
- [12] 阿部昭博, 小橋一夫: 輸配送計画システムにおけるドメインモデル構築とその評価, ソフトウェア開発のためのドメイン分析・モデリング技術シンポジウム論文集, pp. 21-30, 1996.