

オフィスプロシジャ実行における例外処理の事例知識獲得手法

2J-2

大久保 雅且 石井 裕

NTTヒューマンインタフェース研究所

1. はじめに

オフィスワークの多くは組織的に定められた業務処理手順に従って進められる。業務処理要領や各種帳票処理記入要領に基づく標準的な手順は、複数のオフィスワークにより遂行される一連のアクティビティとして表現でき、office procedureと呼ばれる[1]。Office procedureに関する知識を持つシステムでは、それに基づくアドバイスや自動実行が可能となる[2,3]。

しかしオフィスは本質的にopen systemである[4]。予想外の状況変化によって正常時の業務処理手順がそのまま適用できず、何らかの修正が必要となるケースは現実のオフィスでは極めて多い。このような状況において支援を行なうには、人間の行なったプラン生成・修正の経験そのものを事例知識として収集・獲得し、以後同じ様な状況が起こったときにその知識を役立てる方法が有効である[5,6]。

本稿では、office procedure知識ベースにoffice procedure実行監視制御システムを組込むことにより、人間が行なうプラン生成と修正の事例を知識として半自動的に獲得する手法を提案する。

2. Office Procedure の知識表現と実行監視制御

Office procedureは、分岐や並行処理を含む制御構造の他に各アクティビティを実行するエージェント、関連する多様な文書等の関係構造を表す手続き的な知識(procedural knowledge)である。我々はオフィスワーク記述と分析のためにオフィスモデルOM-1を構築し、その有効性を検証した[7]。さらに、OM-1を基にオブジェクト指向の階層型意味ネットワークを基本とするオフィス知識表現の枠組を開発し、オフィスプロシジャ知識ベースを実現した[2,6]。

本知識ベースではoffice procedureの動的特性を表す概念としてactive(dead) procedure instanceを導入した。一般的な物品購入処理がoffice procedure classであるのに対し、例えば1988年5月に発注して現在進行中のMacII購入処理

は active procedure instance、実行完了した処理は dead procedure instanceとなる(図1参照)。

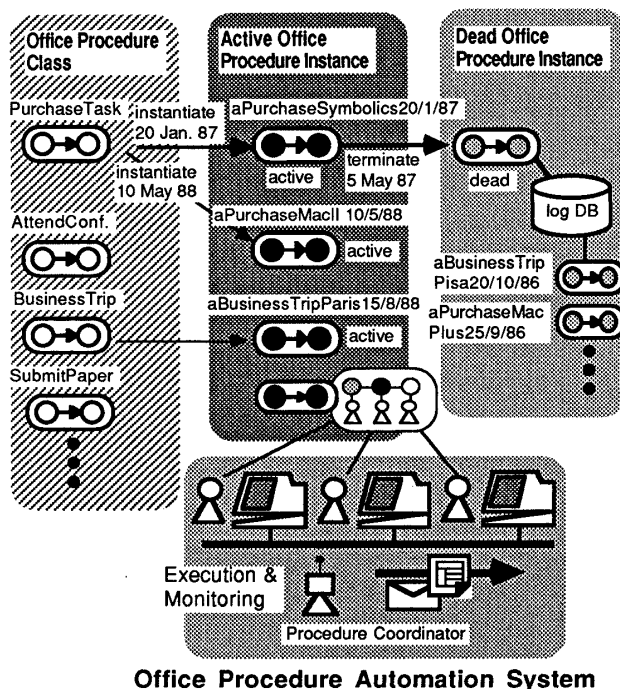


図1 active procedure instance の概念と office procedure 自動化システムによる実行制御と進行監視

Office procedure自動化システムは知識ベース中に定義されたoffice procedureの制御構造に従って、複数のオフィスワーク間にまたがるオフィスワーク全体の進行を分散ネットワーク上の電子メール機能を用いて自動制御するシステムである[3]。通常procedure coordinatorと呼ばれる1つのプロセスがoffice procedure全体の進行を監視制御する。このシステムが監視する対象がactive procedure instanceである(図1参照)。

Office procedure自動化システムは帳票駆動の構造の明確な処理の実行監視制御に適する。我々はこのoffice procedure自動化システムの適用領域をless-structuredな業務にまで拡大するとともにoffice procedure知識ベースシステムと有機的に統合し、オフィスワークの設計からアドバイス、プランニング、実行監視制御まで一貫して行なえることをねらったシステム(KBOIS: Knowledge-Based Office Information System)の構築を進めている[2,6]。

3. 事例の収集と記憶

Open systemとしてのオフィスでは予期しない事象が発生し得る。その予測法や対処法を汎用の手続きや規則として予め用意することはできない。このとき、以前の問題解決事例の想起が問題の予測や解決の有効な手段となる。特に、利用者が問題を特定できない場合や、問題自体は異なっても周囲の状況が類似している場合には、現在置かれている状況をキーとして検索された過去の事例が、現在の問題を把握・理解・解決するための有効なヒントとなり得る。

2章で述べたoffice procedure自動化システムの活用により、本システムではタスクの進行状況の追跡とその記録を可能としている。これによりoffice procedure中の各アクティビティに関して、

- ・ 開始時刻と終了時刻
- ・ 各帳票や文書にどのような情報が記入・参照され、いつ、誰の手から誰の手へわたったか
- ・ そのアクティビティが誰により遂行されたか

等の情報を自動的に記録することができる。

プラン生成時や実行時に修正が必要となった場合には、上記に加えて問題が発生した状況を特定できるような命題や条件等を明確にし、記録する。具体的には以下の手順によって状況の特定化が行なわれる。

- (1) システムの進行状況監視機能により該当するactive office procedure instanceの現状態(例えば各アクティビティの進行状況や各帳票類の所在等)を特定化する。
- (2) どのような問題が発生したか(例えば期日を過ぎても納品されない等)を明確にする。
- (3) あるアクティビティの実行が停止している場合には、なぜそのアクティビティが実行できないのか(どの制約条件あるいは前提条件が満たされていないのか、どのような事象の発生によるものなのか等)を調べる。
- (4) 初期プランと該当するprocedure instanceの実行経過記録とを比較し、あるいは問題を発見したアクティビティから制御や帳票のフローを逆昇ることにより、問題の発生原因となったアクティビティを探る。
- (5) 原因を特定化する。

状況の特定化を完全に自動的に行なうことは(1)を除いては困難であり、利用者による指示や判断が必要となる。システムは状況の特定化が容易に行えるように、office procedureの状態の図示と、特定化すべき項目の候補メニューの提示による支援を行う。

予想し得る問題や原因の候補は予めシステムに用意できる。これらの候補は、上記手順中、前の段階までで明

らかになった部分的な状況によって順次絞り込まれる。予想外の問題や原因に関しては、それが過去に発生したものであれば、その事例に基づいて候補の追加が行われる。このようにして得られた発生問題やその原因の候補をメニューとして提示する。利用者はメニューから選択するか、メニュー内にない場合には新たに入力することによって、状況の特定化を進める。新たに入力された問題や原因は、以後同じ様な状況が発生した場合には、前述のようにメニューの一部として提示される。

各事例は収集時に記録された状況からインデクスを生成し、修正プランと共に記憶される。事例の記憶法は、エピソード記憶のためのMOPs[8]をモデルとする。

標準処理手順は知識ベース中に既に組織的に分類されて格納されている。雛型として使用したoffice procedure classがMOPsにおける一般化エピソード(generalized episode)に対応する。また、問題が顕在化した時点でのoffice procedure instanceの状態、発生した問題やその原因を元にしてインデクスが生成され、実行されたプラン(対処法)が事例として記憶される。このうち、システムが予め用意したメニューに含まれる問題や原因に関しては、構造化されたインデクスとして用意しておく。利用者が新たに加えた場合には新しくインデクスが生成される。

4. おわりに

Office procedure自動実行システムと組み合わせ、その自動実行監視機構と自動記録機能を利用することによって、(半)自動的に事例を収集・記憶する方法を提案した。本稿で示したように、標準の手続きに含まれない例外的事象の発生時には、利用者のわずかな労力によってシステム自身の知識を増加させ、有用性を高め得る。さらに、複数の人間が本システムを利用することにより、事例(経験)の共有が可能となる。今後、プロトタイプシステムの作成と評価を通じて本手法の有効性を確認する。

参考文献:

- [1] Ellis, C.A., Information Control Nets: A Mathematical Model of Office Information Flow, in 1979 Conf. on Simulation, Measurement and Modeling of Computer Systems, 225-239
- [2] Ishii, H. and Kubota, K., Office Procedure Knowledge Base for Organizational Office Work Support, IFIP WG8.4 Working Conf. on Office Information Systems, Linz, Aug. 1988 (to appear)
- [3] Kreifelts, T. et al., DOMINO: A System for the Specification and Automation of Cooperative Office processes, in EUROMICRO 84 (1984) 33-41
- [4] Hewitt, C., Offices are Open Systems, ACM Trans. Office Information Systems 4 (3) (1986) 271-287
- [5] Kolodner, J.L. et al., A Process Model of Case-based Reasoning in Problem Solving, in Proc. IJCAI-85 (1985)
- [6] 大久保, 石井, オフィスプロシジャ修正事例に基づくプランニング技法, AI学会SIG-HICG-8801-4, 1988年6月
- [7] 石井, オフィスモデルQMIによるオフィスワーク記述と分析, 信学技法OS86-24, 1986年9月
- [8] 石井, 大久保, オフィスプロシジャの知識表現と分散処理環境における実行制御システム, 信学技法OS88, 1988年7月
- [9] Schank, R.C., Dynamic memory: A theory of learning in computers and people (New York: Cambridge University Press 1982)