

作業の順序性に基づくノウハウを獲得し利用する ”インテリジェントワークフローシステム”の開発

佐々木 幹郎 御宿 哲也 高野 啓 田中 聡 前田 暉

msasaki@ap.isl.melco.co.jp

三菱電機(株) 情報技術総合研究所
〒247 神奈川県鎌倉市大船 5-1-1

人間の高度な判断を必要とする知的作業において、熟練作業者の持つノウハウを再利用できれば非熟練作業にも同様の作業が行える事が期待できる。しかし、エキスパートシステムでは熟練作業者にノウハウを記述してもらうことが難しく、また、事例ベース推論では問題領域に依存した属性を定義しなければならないためその処理は問題依存せざるを得ない。

本論では熟練作業者が行った作業の順序性に着目し利用する”インテリジェントワークフローシステム”について述べる。本システムは、作業者の問題解決のための作業を処理とその処理結果の組合せで定義し、それら処理間の順序性に関する情報をノウハウとして蓄積し、次の作業を行う際の作業支援に利用する。そのため、応用に依存しないノウハウ獲得が行える。

A system that acquires and uses know-how based on the order of problem solving process – IntelligentWorkflow

Mikio Sasaki Tetsuya Mishuku Akira Takano Satoshi Tanaka Akira Maeda

Information Technology R & D Center, Mitsubishi Electric Corp.
5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan

In some intelligent works that require highly human decision, it is expected that a beginner can do the same work as experts by using experts' know-how. But it is difficult for experts to describe their know-how, and a case-based reasoning system is necessary to define some attributes of the domain problem, so it inevitably depends on those problems.

This article describes “IntelligentWorkflow” that focuses on and uses experts' way of processing. This system defines an problem solving process as a combination of simple procedure and its result, and stores information about the order of procedures as “know-how”. In its knowledge acquisition process, acquisition process is independent from its domain problem.

1 はじめに

近年の計算機システムの能力向上と個々のツールの機能向上に伴い、様々な分野で計算機が用いられてきており、複雑な問題についての問い合わせに対して適切な回答を行う業務など、人間の経験による高度な判断を必要とする知的作業にも計算機システムが適用されはじめています。そして、今後ますますその傾向は強まることが予想されます。

このような人間の経験に基づく判断や処理を必要とする作業において、熟練作業者の持つノウハウを再利用できれば非熟練作業者にも同様の作業ができると期待できるが、このための技術として、エキスパートシステムや、事例ベース推論などの技術が提案されている [1]。

しかし、エキスパートシステムでは、熟練作業者の持つ知識を記述することについての困難さが知られており、また、事例ベース推論を用いたシステムでは、事例表現において問題領域に依存した属性を定義しなければならず、そのため、システムを別の应用到しようとする、新たにどのような属性を用いるかを応用にに応じて定義し直す必要がある [2]。

我々は、熟練作業者が行った操作や作業の順序性に着目し、どんな状況の元にどの作業を行うのが適切かという情報をノウハウとして抽出しておき、非熟練作業者が問題解決を行う際に、このノウハウを利用してシステムが最適な作業についてアドバイスするシステムを検討している。このシステムをインテリジェントワークフロー（以降 IWF）[3] システムと呼んでいる。

IWF システムは次の特徴を持つ。

- 作業内容に特化した属性ではなく熟練作業者が行った作業の順序性に注目し、应用到強く依存しないノウハウ獲得方法を備える
- 熟練作業者が行った作業の履歴からノウハウを得るため、予めノウハウを用意する必要がない
- ノウハウによる完全な自動実行を行わず、人間が作業を行う際、対話的に支援を行う

2 IWF システムのアプローチ

2.1 IWF システムの狙い

人が行う作業を計算機で支援する場合、3つのタイプに应用を分類することができる。

ある目的に対して多くの人が行う作業手順が、同一で固定的なのであれば、その作業は計算機で自動的に行わせることができる。これを定型作業と呼ぶ。

一方、コンテンツ作成や創作活動などは、個人のアイデアや創造性が必要とされることからその作業手順は毎回違いが大きいことが予想され、計算機により

自動化することは困難である。このような作業を非定型作業と呼ぶ。

その都度作業が全く異なるものではないが、熟練した人間が作業中のその時々判断に応じて複数の選択可能な処理の中から最も適切なものを選択して行うような作業もある。このような作業は、定型作業と非定型作業との中間に位置しており、我々は、これを半定型作業と呼んでいる。そして、その時どのような処理を選択したかが、その作業者の持つノウハウとして捉えることができ、例えばサービスセンターでの故障問い合わせに対する回答業務などがこれにあたる。

IWF システムはこのような半定型業務の支援を対象としており、熟練作業者の行う作業の順序性に着目することでノウハウを獲得し、非熟練作業者に対する作業支援を行う。

2.2 機能構成

図 1 に IWF システムの概略的な機能構成を示す。

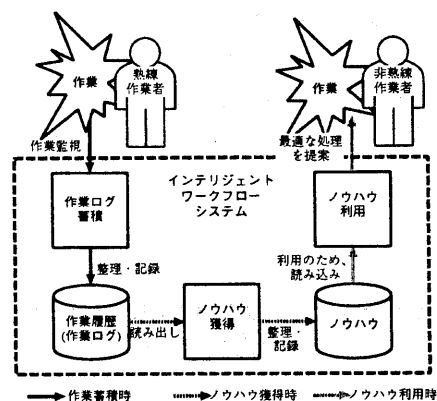


図 1: システムの概念

熟練作業者が行った作業履歴には、その作業者の持つノウハウが含まれていると捉え、まず、熟練作業者が行った作業を作業ログ蓄積により監視し、その作業履歴を蓄積する（図中、作業蓄積時の流れ）。

次に、ノウハウ獲得により蓄積された情報から、作業を行うのに有用な情報を抽出し、ノウハウとして蓄積する（図中、ノウハウ獲得時の流れ）。

そして、ノウハウ利用により、非熟練作業者が作業を行う際にノウハウを用いて最適な処理を提案することで作業支援を行う（図中、ノウハウ利用時の流れ）。

2.3 手順部品と作業ログ

IWF システムでは一つの作業履歴を、単純な処理とその処理結果をペアにし、そのペアを処理した順番に組合せたものとする。この作業履歴情報を作業ログと呼んでいる。また、作業ログは作業開始前の「初期状態」に関する情報も持つ。

一つ一つの処理は、マニュアルに従って機械的に作業を行えるような単純なものである。これを手順部品と呼ぶ。手順部品は作業の中で行う処理を定義したものであり、それぞれ ID により区別される。利用者は必要な手順部品を選択して実行し、その結果どうなったかを処理結果として記録する。処理結果は手順部品を一つ処理した時、作業対象がどうなったか、状況がどう変化したかといった情報で、これは作業者が次のような処理を行うのか選択の際に影響を与える条件となる情報であると考えられる。

図 2 にパソコン故障点検作業についての作業ログの例を示す。

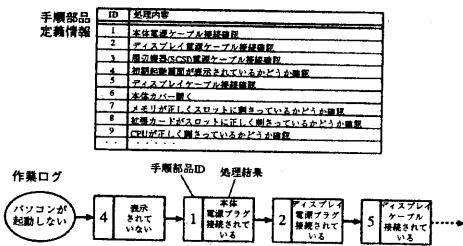


図 2: 手順部品と作業ログの一例

効率的な作業とは、作業ログがより少ないステップで作業の目的を達成した場合の作業と考えられる。本文中では、作業ログを次の形で表す。

状態 X : 1(1a) → 2(2b) → 3(3a)....

これは、「作業開始前の初期状態が「状態 X」であるところに手順部品 1 で定義された処理を行い、その結果の状況が 1a となった。そして、その次に手順部品 2 で定義された処理を行い、その処理結果が 2b であった....」という一連の作業の作業ログを示している。

作業ログは、IWF システムでは図 1 中の「作業ログ蓄積」において蓄積されるが、これは応用ソフトウェアでの作業監視や、利用者による直接入力など応用によりさまざまな蓄積方法が考えられる。

3 整理された情報 - ノウハウ -

3.1 ノウハウの定義

IWF システムでは手順の整理を行い、獲得される情報を「ノウハウ」と呼んでいる。

ノウハウとは、利用者が作業のため手順部品を利用する際に、システムが最適な手順部品を提案するためのノウハウである。これは問題解決の過程で、適用可能な処理(手順部品)が複数あることがわかっている時、それらのうちどれを選択するのが適しているかという情報である(図 3)。

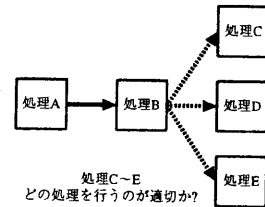


図 3: 複数の手順部品候補が考えられる場合

この手順部品選択のための基準を次のように考える。

作業者は、問題解決のために一種の目的意識 - 例えば、故障問い合わせに対する回答業務であれば、「電源があやしいのでそこを調べよう」といった故障原因の究明や、故障部分の修理処置など - を持ち、それによって一連の作業を行う。この作業者の持つ目的を「仮説」と呼ぶことにする。

そして、作業の途中で仮説に反するような処理結果が得られると、新しく別の仮説を設定し直して作業を進めてゆく(図 4)。

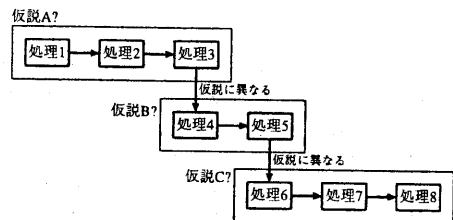


図 4: 仮説に基づく作業の流れ

従って、手順部品の選択には次の 2 通りの考え方がある。

1. 仮説に従って、前の処理と関連のある手順部品を選択

- 仮説の変更により、新しい仮説に属する手順部品を選択

IWF システムではそれぞれ「仮説内ノウハウ」「仮説間ノウハウ」と呼び次のように定義する。

仮説間ノウハウ 「ある手順部品よりもある手順部品が優先され行われた」という情報

仮説内ノウハウ 「ある手順部品がある手順部品と連続して行われた」という情報

これらのノウハウを手順部品選択の指標とする。
ノウハウの獲得は次のように行われる。

まず、作業ログからは、作業の途中の状況に応じた手順部品間の優先関係に関する情報を抽出し、仮説間ノウハウとして記録する。これは、全ての作業ログに対して行われ、手順部品間の基本的な優先関係が得られる。

さらに、全ての作業ログに着目して、複数の作業ログ内に共通した手順部品列があれば、その部分の仮説間ノウハウを手順部品のつながりを表すように統合する。これが仮説内ノウハウである。

それぞれのノウハウについて記述する。

3.1.1 仮説間ノウハウ

一つの作業ログから得られる、作業途中の様々な状況における手順部品間の優先関係を定義した情報を仮説間ノウハウと呼んでいる。仮説間ノウハウは次の情報を持つ。

- ノウハウの利用条件 (状況)
- 手順部品間の優先づけ

仮説間ノウハウを次のように表す。

```
if 状態 A then
  x > y
```

このノウハウの利用条件は、状態 A であり、その時手順部品 x は手順部品 y よりも作業に適していることを示している。

ノウハウの利用条件である「状態 A」は、その状況を記述できる表現であれば特に規定しないが、現在の IWF システムでは、初期状態とそこから利用した手順部品とその処理結果で示すようにしている。

3.1.2 仮説内ノウハウ

作業ログが複数蓄積されると、それらの作業ログの中で続けて用いられる手順部品が検出可能となる。

「手順部品 x の次は手順部品 y」といった、よく用いられる手順部品の連続性に関する情報を、IWF

システムでは仮説内ノウハウと呼んでいる。複数の作業ログに共通に現れる連続した手順部品群を検出し、仮説内ノウハウとして記憶する。

仮説内ノウハウを次のように表す。

```
if 条件 A then
  x(Rx) → y(Ry) > z
```

これは、熟練作業者の作業において、条件 A という状態で、手順部品 x と手順部品 y が続けて行われることが多く作業ログ群の中に見られる場合に、獲得される仮説内ノウハウである。Rx, Ry はそれぞれ手順部品 x, 手順部品 y を行った際の処理結果を示している。

利用条件は、仮説間ノウハウと同様である。
次に、ノウハウの獲得方式について述べる。

4 ノウハウ獲得

ノウハウ獲得のための処理の流れを図5に示す。

この処理は図1の「ノウハウ獲得」において、新しい作業ログが追加されるごとに行われる。

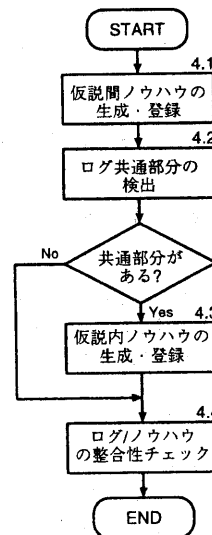


図5: ノウハウ獲得処理の流れ

新しいログが追加されると、まずそのログに含まれる手順部品間の優先関係を仮説間ノウハウとして取り出す。次に新しいログと、既に登録されている他の作業ログとの照合を行い、互いに一致する手順部品列を検出する。

一致する手順部品列がある場合は、そこから連続した手順部品に関する情報を得て、仮説内ノウハウを

作る。

4.1 仮説間ノウハウの生成・登録

仮説間ノウハウは、新しく熟練作業者が作業を行うごとにその作業ログから得る。例えば、次の作業ログ

状態 X : 1(1a) → 2(2b) → 3(3c)

が得られた場合、作業ログに含まれる「どの手順部品を先に行ったか」という情報を仮説間の優先順位に基づく適合性と捉え、以下の仮説間ノウハウ (1) ~ (4) を得ることができる。

```
if 初期状態 = 状態 X then
  1 > 2          . . . (1)
  1 > 3          . . . (2)
```

```
if 初期状態 = 状態 X and
  1 の結果 = 1a then
  2 > 3          . . . (3)
```

```
if 初期状態 = 状態 X and
  1 の結果 = 1a and
  2 の結果 = 2b then
  3 > φ          . . . (4)
```

最初の仮説間ノウハウ (1), (2) は「状態 X において、手順部品 1 は、手順部品 2 や手順部品 3 よりも作業するのに適している」という仮説間ノウハウである。

既に同一の仮説間ノウハウがシステムに登録されている場合は、その仮説間ノウハウは新しく支援される作業ログが得られたことで補強される。

4.2 ログ共通部分の検出

複数の作業ログの中に共通に見られる一連の作業順序を、仮説に基づく「連続性」として捉え、その部分を仮説内ノウハウとして生成する。

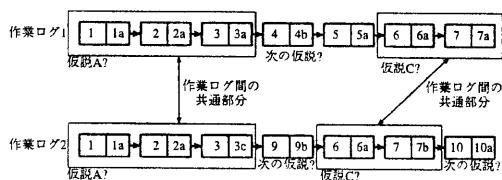


図 6: 複数作業ログからの共通部分検出

仮説内ノウハウの生成のために、蓄積された作業ログと比較し、共通部分を検出する。

例えば、二つの作業ログ

状態 X : 1(1a) → 2(2a) → 3(3a) → 4(4a) → 5(5a)

状態 X : 1(1a) → 2(2a) → 3(3b) → 7(7a) → 8(8a)

がある場合、共通部分

状態 X : 1(1a) → 2(2a) → 3

を検出する。この共通部分から仮説内ノウハウが生成される。

4.3 仮説内ノウハウの生成・登録

前節で検出されたログ共通部分から、連続性のノウハウとして仮説内ノウハウを生成し、記憶する。

先ほどの、共通部分

状態 X : 1(1a) → 2(2a) → 3

からは、次の仮説内ノウハウ (5), (6) が得られる。

```
if 初期状態 = 状態 X then
  1(1a) → 2(2a) → 3(3a) > 4 . . . (5)
  1(1a) → 2(2a) → 3(3a) > 7 . . . (6)
```

これは、状態 X の時、手順部品 1, 手順部品 2, 手順部品 3 が手順部品 4 や手順部品 7 よりも先に連続して処理することが適しているというノウハウである。

これらの仮説内ノウハウは、既に同じものがシステムにある場合はその頻度を増加することで補強され、無い場合は新しいノウハウとして登録される。

4.4 作業ログ/ノウハウの整合性チェック

複数の異なる作業ログからノウハウを獲得すると、それらの作業ログやノウハウの間で矛盾が発生する可能性がある。

これは、複数のログの間に連続性が認められ、ノウハウとして得られたが、その連続性に反するような手順部品の使い方が新しく追加された作業ログ内に見つかった場合などが考えられる。

例えば、既に以下の仮説内ノウハウ

```
if 初期状態 = 状態 X then
  1(1a) → 2(2a) → 3(3a) > 4 . . . (5)
  1(1a) → 2(2a) → 3(3b) > 7 . . . (6)
```

と、元になった作業ログ

状態 X : 1(1a) → 2(2a) → 3(3a) → 4(4a) → 5(5a)

状態 X : 1(1a) → 2(2a) → 3(3b) → 7(7b) → 8(8a)

が存在していた状態で、新しい作業ログ

状態 X: 1(1a) → 2(2a) → 5(5b) → 9(9a)

が追加され、次の仮説間ノウハウが得られた場合を考える。

```
if 初期状態 = 状態 X and
  1の結果 = 1a and
  2の結果 = 2a then
  5 > 9      . . . (7)
```

この仮説間ノウハウ (7) は上記仮説内ノウハウ (5), (6) に反する。これにより、仮説内ノウハウ (5), (6) は縮小され、次の仮説内ノウハウになる。

```
if 初期状態 = 状態 X then
  1(1a) → 2(2a) > 3      . . . (8)
```

ノウハウ (5), (6) の後半部分 2 → 3 が上記 仮説間ノウハウ (7) と矛盾し無効になるため、前半部分 1 → 2 だけが仮説内ノウハウとして残っている。

このように、ノウハウは互いに整合性をチェックしながら獲得され、分解される場合もある。

5 ノウハウの利用

ノウハウの利用について記述する。

これらの処理は図1中の「ノウハウ利用」において行われる。

ノウハウは、定義にも示した通り、そのノウハウがどのような状況で使われるべきかを示した条件を持っている。利用の際は、まず、現在行っている作業の状態—すなわちどのような手順部品を利用してその結果どのような状態になったかという情報—と、ノウハウの持つ条件情報を比較し、一致するノウハウを得る。

検索条件に一致するノウハウから、最も優先されるべき手順部品を得るため、頻度などでそれぞれのノウハウを評価・順位づけする。そしてそのノウハウの持つ仮説の先頭の手順部品を適切な手順部品として提案する。複数のノウハウから一つの手順部品が提案される場合は、その手順部品はより優先度が高くなる。このように、複数の手順部品がその優先順により並び換えられて提案される。

利用者により手順部品が選択され、その仮説内ノウハウを利用すると決まると、次の事項が発生しない限り、その仮説内ノウハウ中の手順部品の順番に従って利用すべき手順部品を提案する。

- 仮説内ノウハウの最後まで手順部品を利用し、仮説内で次の手順部品がない

現在の作業の状況

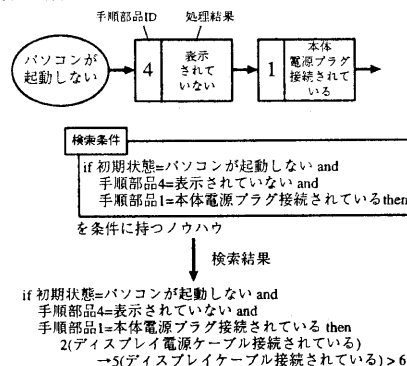


図7: 作業状況と適合するノウハウの検索

- 手順部品利用の結果がノウハウに登録されたものと異なる
- ノウハウに基づき提案した手順部品と異なるものを、利用者が選択し、処理した

上記の事柄が発生した場合は、再び現状の状態を条件部分を持つノウハウの検索を行い、どのようなノウハウが利用可能かを調べ、該当するノウハウがあればそれに基づいて手順部品の提案を行う。

作業手順の各状況に応じて、利用可能なノウハウを検索し、得られたノウハウを利用して、そこに最適な処理(手順部品)を提案することで、作業者の支援を行う。

6 試作システム

6.1 適用対象

現在我々は、IWF 試作システムの開発を行っている。

本試作システムは応用対象を、サービスセンターのような場所で、お客からの故障の問い合わせに対して適切なアドバイスを行う業務(問い合わせ業務)のための支援システムと想定している。

問い合わせ支援作業の業務は次のようなものである。

1. 顧客からの故障に関する電話が来る
2. サービスセンター作業者は顧客から故障の状況を聞く
3. 顧客との応答内容から、故障原因の確定や、復旧のための処理を顧客(実際の作業員)に指示する
4. 作業員側で処理を行う
5. 処理結果内容をサービスセンターへ報告する
6. 2.に戻って、故障が直るまで繰り返し

4. 少数の作業ログからもノウハウ獲得が可能
IWF システム は少ない作業ログからもノウハウ獲得を行い、支援を行うことができる。ただし、制約のところでも述べた通り、その作業が熟練作業者の行った効果的な作業であることが前提となる。

7.3 関連技術

事例ベース推論に関する研究では、様々な事例の記述方式や、検索の高速化などの技術が提案されており、設計・診断などさまざまな応用に適用されている [1]。これらのシステムでは、事例記述について属性を定義し、その値を類似判定に用いるため問題に処理が強く依存し、別の応用への適用が困難である。IWF システム では、問題依存する属性を避け、順序に着目して作業からノウハウを取り出すというアプローチをとっている。

また、単一の事例のみを修正して適切な解を得ることは限界があり、複数の事例を利用する必要性が指摘されている [2]。IWF システムは過去の複数の作業ログから必要な情報をノウハウとして抽出しておいて、そのノウハウに基づいて必要な処理を提案する。

この他、より具体的に作業ログを蓄積・編集・再利用する製品として、パソコン上のソフトウェアの操作など GUI の操作を記録してスクリプトとして実行するソフトウェア製品がある [5]。これらは、現状では効果的な手順を利用者が作成してゆく必要があり、複数の作業ログからノウハウを抽出する機能を持つものは見当たらないが、作業ログの記述および抽出の技法として注目したい。

7.4 今後の課題

先に述べた制約や問題点に対する今後の課題について述べる。

7.4.1 評価・検証

今後、まず、試作システムにより得られたノウハウや、ノウハウを利用したシステムによる提案が、適切なものかどうか評価・検証を行う。本方式は汎用的なノウハウ獲得を目指したものであるため、できるだけ多くの問題領域について評価する。

7.4.2 作業ログの属性に基づくノウハウの獲得

現状の IWF システムでは熟練作業者の行った作業に着目するが、その作業について順序性を持たない応用についてはノウハウ獲得が行えない。

処置を含まない診断などの応用では、人間の行う作業の順序性にあまり意味がなく、作業対象の状態な

どの属性値が重要視される。このような応用に対して、作業ログに含まれる処理に属性を与え、その属性をもとに優先度を抽出する方式 [4] を組み込み、問題解決の過程を抽象化して知識として蓄積できるようにするような拡張方式の検討も行う。

7.4.3 作業ログ/ノウハウの整理・高速化

IWF システムにおいて問題解決に必要な情報はあらかじめノウハウとして抽出されるが、作業ログが増加するとノウハウの量も増加してゆくと、ノウハウの整理や高速化の検討が必要である。

特に、現在得ているノウハウは作業の各局面に応じて最適な次の処理を提案するための情報であるが、ある程度ノウハウを蓄積して、数ステップ先の作業手順を提示できるようなノウハウを生成する検討を行う予定である。

8 まとめ

作業者の作業履歴に注目し、作業履歴から順序性を抽出してノウハウ情報として整理する方法についての検討と試作システムを紹介した。

この方式は、熟練作業者のノウハウを予め記述するのではなく、作業を積み重ねて行くことでノウハウの蓄積が行える。また、獲得方式自体は様々な応用に対しても対応可能であるように、応用に特化した情報を利用するのではなく、作業の順序性に着目し利用した。今後は試作システムを元に評価を行ってゆく予定である。

参考文献

- [1] 奥田, 山崎: “事例ベース推論とその応用例”, 情報処理学会誌 Vol.31 No.2, pp.244-254(1990).
- [2] 小林: “事例ベース推論の現状と展望”, 人工知能学会誌 Vol.7 No4, pp.559-556(1992).
- [3] A. Takano, T. Mishuku, M. Sasaki, S. Tanaka: “Intelligent workflow – a Method For Efficient Trouble Shooting Services by Acquiring and Re-using Procedural Knowledge –”, CAIMS '96.
- [4] T. Mishuku, T. Tsuchida, S. Tanaka: “A Co-operative Scheduling System –CoPlanner– by Interactive Knowledge Acquisition”, DKSM '96, pp.243-262.
- [5] “新製品レポート ScriPlayer-PRO 32”, 日経バイト, March 1996.