グループウェア Michele への学習機構の適用

1N−8

岡田 豊史　　中内 靖　　安西 祐一郎

慶應義塾大学

1 はじめに

現在、オフィスや研究機関においてWSが導入され、電子メール等が盛んに使われるようになってきている。我々の研究室ではこのような分散環境において、組織の生産性を向上させるために、マルチエージェントモデルに基づくグループウェア Michele(Multi-agent Interface with Communication by Hectic ELEments)[1] を開発している。Michele は、オフィスワークの大半を占める文書依存の定型業務の支援に有効な枠組を提供する。本稿では、Michele に ID4[2] を利用した学習機構を組み込む。そして、Michele における学習の有効性について考察する。

2 協調作業における学習

Michele の応用例として、研究室における物品の購入作業を考える。物品の購入作業は複数のワーカの間での書類の受渡しとしてモデル化できる。以下に物品の購入作業の手続きを示す。

1. 研究員が「物品購入請求書」を秘書に提出する。
2. 秘書はその品物を取扱っている業者を判断し、「見積請求書」を送付する。
3. 業者は「見積書」を作成し、秘書に送付する。
4. 秘書は見積と予算を考慮し購入するかどうか決定する。購入を決定した場合、用度課に「申告書」を提出する。
5. 秘書は用度課から「許可証」が届いたら購入する。

この手続きにおいて、秘書は「見積請求書」の送付先を「物品購入請求書」に書かれている情報から秘書固有の知識を使って決定する。従って、ワーカ間でやり取りされる書類を監視することによって、秘書固有の知識を獲得できると考えられる。これらの知識はプログラマがシステムに組み込むことは可能であるが、非常に繁雑で困難である。もし、システムが秘書固有の知識を自動獲得するようなことができれば以下のような利点があると考えられる。

引き継ぎ作業の円滑化 オフィスにおいては、明文化しづらいような個人固有の知識を用いて作業が行なわれることが多い。ところが、このような知識をシステムが自動的に学習してくれれば、ワーカの交替時に学習によって蓄えられた前任者の知識を利用できるため引き継ぎ作業を円滑に行なうことができる。

入力作業の軽減 書類作成時にワーカが入力すべき項目を、システムが学習した知識をもとに自動的に入力することができるため、ワーカの入力作業を軽減することができる。

手続き記述の軽減 オフィスの手続きを記述するアプリケーション・プログラマは、ワーカが判断するような知識について記述する必要がなくなる。

このような知識獲得システムを Michele に組み込むためには、協調作業における書類の流れを明示的にする必要がある。すなわち、ワーカが判断の材料としている知識と判断される項目をシステムが抽出できるようにする必要がある。そこで我々は、Michele の書類を作成・送付する機能をまとめたマクロ関数を MDL に導入する。このマクロ関数を用いて書類の作成・送付を実行することによりワーカが判断に利用している知識をシステムが獲得することを可能とする。

3 設計

3.1 書類作成のためのマクロ関数

システムが学習するためには、ワーカがどのような情報を判断の材料とするか、また、そのような知識を元に何を判断するかという情報を抽出する必要がある。例えば、物品購入の手続きでは秘書は研究員から購入依頼に対して物品を取扱っている業者を決定する必要がある。この決定を行なう場合、秘書はこの物品の種類・製造元・値段・利用目的・名前といった情報 (Table.1 参照) を利用する。そして、決定された業者の名前は新しく作成される「見積請求書」の送付先のアドレスとして記述される。

Michele のマルチエージェントモデルでは、書類はエージェントとして表現され、書類に記述される項目とその値はエージェントの内部状態であるフィールドとその値として表現される。また、ある書類のエージェントが届いた時、その書類の到着によって新たに起動されるワーカの手続き (「物品購入請求書」を受けとった秘書が「見積請求書」を作成・送付する) は、到着した書類のエージェントのメソッドとして表現される。したがって MDL では、ワーカが判断に利用する情報は到着する書類のエージェントにフィールドとして記述される。また、エージェントの到着により新たに作成・送付されるエージェントの記述は到着したエージェントのメソッド内に記述されている。そして、到着したエージェントの情報を元にワーカが判断し決定した項目は新たに作成されるエージェントのフィールドに格納される。

現在の MDL のシンタックスでは上述したような情報をシステムが抽出することは困難である。そこで我々は、このような情報をシステムが容易に抽出できるようにマクロ関数 FormGenerationMacro を MDL に付加する。FormGenerationMacro は第一引数に示されるエージェントのインスタンスを生成する。そして、生成されたインスタンスのフィールドのうち第二引数以降に記述されたフィールドに対して値を格納し、新たに書類を作成する。FormGenerationMacro を使った「物品購入請求書」の記述例を Fig.1 に示す。ここで、「物品購入請求書」エージェントのうち": prec" キーワードが付加されたフィールドの

Applying Learning Mechanithm to groupware Michele

Toyoshi OKADA, Yasushi NAKAUCHI, Yuichiro ANZAI

Keio Univ.

値が秘書が「見積り請求書」エージェントを作成するために必要な知識となる。また、FormGenerationMacroによって作成される書類の項目のうち、これらの知識を元に判断されるであろう項目に":learn" キーワードが付加される。

```
(DEFAGENT 物品購入請求書
  (FIELD kind:  :prec)
  (FIELD quantity:)
  (FIELD maker: :prec)
        :
  (METHOD 見積請求書の作成
        :
    (FormGenerationMacro 見積請求書
      (FIELD supplier: (Ask MI
        :comment "購入先を選んで下さい。"
        :menu    "COOP,Nezuram,XEROX")   :learn)
      (FIELD budget:   (Ask MI
        :comment "予算の種類を選んで下さい。"
        :menu    "equipment,books,misc") :learn)
    )
        :
  ))
```

Fig.1 「物品購入請求書」の記述例

このようにFormGenerationMacroを用いて書類を作成することにより新しい書類を作成するために必要な情報を抽出することができる。抽出される情報はTable.1に示すように、複数の属性と値の組とそのクラスとして表現される。そこで、我々はこのような知識を学習するための学習アルゴリズムとして帰納的学習アルゴリズムのうち教師有りでインクリメンタルであるID4を利用する。学習の結果生成される判別木は新しく作成されるエージェントのうち": learn" キーワードが付加されたフィールドの数だけ生成される。

属性	種類 (kind)	furniture
	製造元 (maker)	itoki
	値段 (cost)	middle
	利用目的 (use)	firtures
	名前 (item)	desk
クラス	販売店 (class)	nezuram

Table.1 Sample Data

3.2 FormGenerationMacro 実行のアルゴリズム

以下に、FormGenerationMacroの実行アルゴリズムについて説明する。

1. マクロ関数が呼ばれると、第一引数に示されるエージェントのインスタンス（書類）を生成する。
2. 生成された書類のフィールドのうち":learn" キーワードのついたフィールドに、既に学習した判別木を元にその値を求めて格納する。
3. 書類をユーザに提示し、学習によって決定された項目が正しいかどうかを確認してもらう。
4. 学習結果が間違っていた場合、ユーザに正しい値を入力してもらう。そして、正しい値を例として判別木を学習させる。

4 実装及び評価

FormGenerationMacroをMDLに組み込み、システムが知識を獲得するようにした。また、ID4により生成される判別木は個々のユーザ環境で永続的に保持されるようにファイルとして記録すれるようにした。

我々の研究室での実際の秘書の仕事からサンプルデータを50個を作成し、FormGenerationMacroを用いた書類作成を用いて評価をとった。Fig.2に50個のデータをインクリメンタルに与えた時の正当率を示す。ID4では、情報理論を適用して最適な判別木を生成しているため、暗記学習に比べて正当率が高くなる。約30個のデータを学習した時点で90％程度の正当率を示しており、実用に耐え得ると思われる。

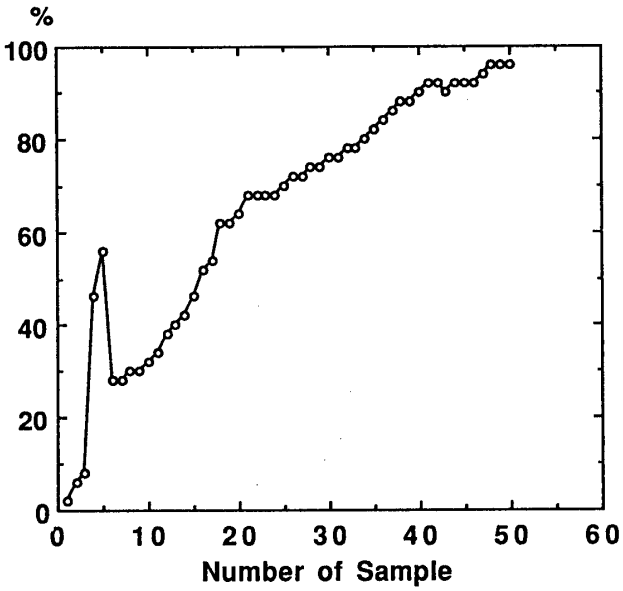

Fig.2 学習による正当率

5 おわりに

本稿ではマルチエージェントモデルに基づくグループウェアMicheleに学習機構を組み込み、その有効性について検討した。そして、グループウェアにおける学習機構の有効性を示す結果を得た。

参考文献

[1] Nakauchi, Y., Itoh, Y., Sato, M., and Anzai, Y., : Michele: A Multi-Agent Interface Architecture for Distributed Open Environments,*TOOLS'91*,(1991).

[2] J., C., Schlimmer, and D., Fisher, : A Case Study of Incremental Concept Induction ,*AAAI'86*,(1986).