

1M-3

ユーザフレンドリな誘導戦略を学習する データエントリシステムの実現*

石井 恵 金田 重郎

NTTコミュニケーション科学研究所

1 はじめに

ユーザがデータベースの更新を行なうためのデータエントリシステムにおいては、(1) 洩れなく、正しく(結果の整合性の保証)、(2) 使い易く(ユーザフレンドリ)、ユーザが必要なデータを入力できなければならない。ここでユーザフレンドリな入力環境とは、ユーザ主導とシステムによるユーザのナビゲーションが調和よく実現されている環境である。具体的には、(a) ユーザによるシステムへの自由な値の通知、すなわち、システムに制約を受けず任意の順序でのデータ投入を許容する入力機能と、(b) ユーザに自然なナビゲーション戦略(データの投入順序)をもつナビゲート機能が実現されていない。更に、ユーザの負担を軽減するため、できるだけ少ないデータ投入数で処理が実現されることが望まれる。

我々は、制約充足の枠組を用いた手法により、ユーザ主導の入力、少ないデータ投入数での処理、整合性の保証を既に実現している[1]。しかし、この従来手法では、ユーザに自然なナビゲーション戦略をもつナビゲート機能は実現されていない。そこで、ユーザフレンドリなデータエントリシステムを実現するために、本稿では、ユーザの操作履歴からの学習により、ユーザにとって自然なナビゲーション戦略を自動的に獲得する手法を提案する。

2 制約充足を用いたデータエントリシステム

2.1 制約表現

データエントリシステムを構築する際、処理結果の整合性を規定するチェック関数、すなわちデータ項目間の制約を記述しなければならない。この際、関係の記述洩れを防ぐようなシンタックスが必要である。そのため、我々は項目間の関係を規定する制約を、その関係が許すデータ状態を、データ状態相互が排他的になるように列挙して表現するシンタックスを提案している[1]。各制約では、宣言されているデータ状態のどれか1つと、処理結果が整合しなければならない。

図1は扶養関係の制約「年収130万円未満の家族がある社員は、扶養申請書を提出し、3000円/月の扶養手当が支給される。」を提案シンタックスで記述した例である。制約は *constraint* で定義され、制約が許すデータ状態は *element* で定義され、制約素と呼ばれる。処理結果は、3つのうちのどれか1つの制約素と整合することにより、正しい処理結果とみなされる。

2.2 制約充足とデータベースの更新処理

文献[1]では、新たに入力されたデータの値と整合するデータ状態を、与えられた制約が許すデータ状態の宣言の中から決定することにより、データベース更新処理を制約充足の枠組で実現する。各制約を充足する際、制約を充足するデータの値を一意に決定できない場合がある。その場合、本

```
constraint( 扶養判断制約 , (a1, a2, a3) ).
element(a1, ( pattern(家族, 存在, 有),
                pattern(家族, 収入, X),
                test(X <= 1300000),
                pattern(扶養申請書, 存在, 有),
                pattern(社員, 扶養手当, 3000) ) ).
element(a2, ( pattern(家族, 存在, 有),
                pattern(家族, 収入, X),
                test(X > 1300000),
                pattern(扶養申請書, 存在, 無),
                pattern(社員, 扶養手当, 0) ) ).
element(a3, ( pattern(家族, 存在, 無),
                pattern(扶養申請書, 存在, 無),
                pattern(社員, 扶養手当, 0) ) ).
```

図1: 制約の例

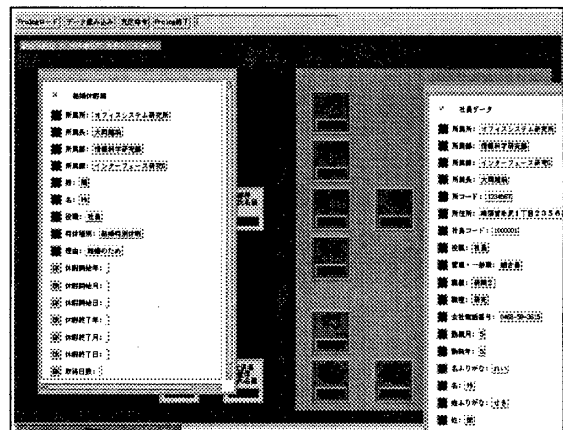


図2: データエントリシステムのGUI画面

枠組ではユーザに質問を行なう。どのデータ項目を質問するかは、制約構造を用いた、質問数を削減するためのヒューリスティックにより動的に決定される。図2は、[1]を用いて実現したデータエントリシステムの例である。

3 履歴からのナビゲーション戦略獲得

ユーザフレンドリなデータエントリを実現するためには、ユーザに自然なナビゲーション戦略をもつナビゲート機能、すなわち、ユーザにとって自然な順序での質問生成が課題となる。例えば、家族のデータを入力する際は、姓→名→住所→生年月日といった順序で、入力を促されれば、ユーザは自然に感じるはずである。

前記手法のように制約構造を用いた動的な質問生成では、ユーザに自然なナビゲーション戦略でのデータ入力は困難である。なぜなら、全ての処理に対して、人間にとって自然となる入力順序を生成する関数を予め定義することは困難であるからである。また、陽に入力順序のシーケンスを予め記述

*Navigation Strategy Learning for User Friendly Data-entry Systems
Megumi ISHII and Shigeo KANEDA
NTT Communication Science Laboratories

するアプローチも、同様に困難である。

我々は、ユーザにとって自然なシーケンスは、ユーザが実際に利用するシステムのユーザインタフェース上でしか獲得できないと考える。なぜなら、実際のシステム上でデータを入れる状況下にユーザを置き、実際にユーザに利用させることにより、ユーザにとっての自然さを獲得できると考えるからである。

そこで、我々はユーザの操作履歴から、データ入力ナビゲーション戦略を獲得するアプローチをとる。この際、以前獲得した履歴と同一、すなわち、データ項目の値とデータ項目の入力順序が全て過去の履歴と同一であれば、過去の履歴をナビゲーション戦略としてそのまま適用することが考えられる。しかし、データ項目の値と入力順序が全て同一の処理に対してのみ、獲得ナビゲーション戦略を適用したのでは実用的でない。

そこで、我々には与えられた制約の充足状況を汎化基準として利用する。なぜなら、各制約において、同じ制約素と整合する制約充足状態同士は、局面的には、意味的に同じ状態とみなせるからである。ただし、制約の充足状態のみでは、制約素とデータ状態との整合状態の変化を伴わないデータのみの場合、同一の状況とみなしてしまおう。そこで、制約素の整合状態の変化を伴わない入力シーケンスを1つのナビゲーション戦略として獲得し、制約の充足状態が変化しない場合は、その状態に対応するナビゲーション戦略にそってナビゲーションを行なう。

データ状態の充足状況への汎化は、各制約素に対して与えられたデータ状態に対する真偽値(「真」、「未定」、「偽」)を定義し、制約素の真偽値の状態へ汎化することにより行なう。例えば、図1において、ユニットの状態が「家族・存在=有」、「家族・収入=700000」、「扶養申請書・存在=有」、「社員・扶養手当=3000」、とした場合、ユニット状態(家族・存在, 家族・収入, 扶養申請書・存在, 社員・扶養手当)=(有, 700000, 有, 3000)は、制約素真偽値状態(a1, a2, a3)=(真, 偽, 偽)に汎化される。

本アプローチの大きな利点は、予めユーザにとって自然なナビゲーション戦略を用意せずとも、ユーザのシステム利用時に、ナビゲーション戦略をプログラミングレスでシステムに組み込むことができる点である。

4 評価

本提案手法を、実用アプリケーションである総務業務エキスパートシステムKOA [2]のタスクに適用した。ユーザはKOAの質問に答えることにより必要な帳票を作成する。

4.1 自然なナビゲーション戦略の獲得

図3に、本方式によって獲得されたナビゲーション戦略によって行なわれた処理履歴(図3(A))と、2.2の手法によって行なわれた処理履歴(図3(B))を示す。後者は、配偶者に関するユニットを聞いた後、社員に関するユニットを尋ね再度、配偶者に関するユニットの質問に戻っており、自然な流れとなっていない(図3(B))。それに対し、本方式では、ユニットのあるまとまりをもって質問しており、より自然なナビゲーションが行なわれている(図3(A))。

何をもちて人間にとって自然なナビゲーションと感じるかはアプリケーションによって異なる。しかし、本方式では、自然さを定義しなくても、思いのままにユーザに利用させることにより、自然なナビゲーションが実現できる。

4.2 汎化レベルと獲得されるナビゲーション戦略

汎化レベルが低いと汎用性のないナビゲーション戦略となり、また、汎化レベルが高いとある状況において適用可能なナビゲーション戦略が増え、適切なナビゲーション戦略が選択されなくなる。KOAのタスクに適用した際、制約の充足状態が同じでも、異なる順序でナビゲートされた方がよい場合があった。これは、ある状態において、過去どの程度ユニットの情報を取得したという点が反映されていないために

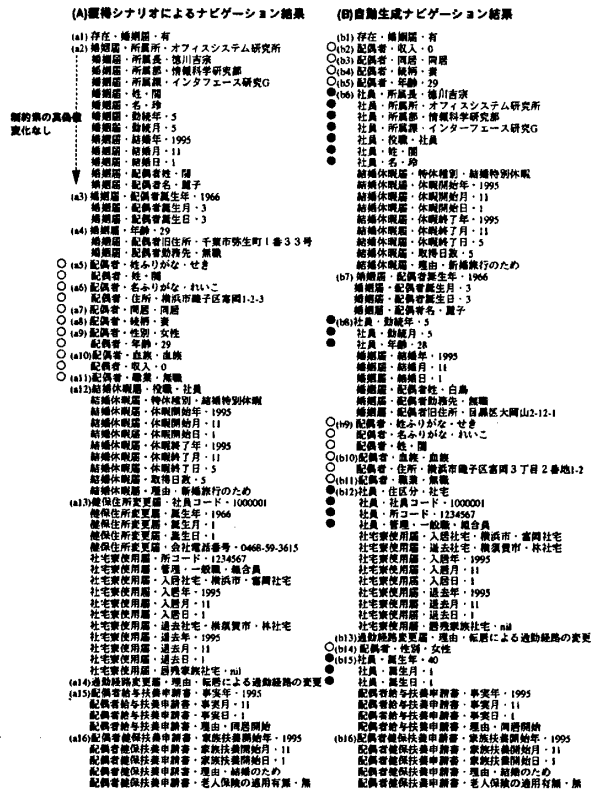


図3: ナビゲーション履歴

おこった。KOAに関していえば、取得されたユニットの情報考慮に入れた汎化レベルが必要であると考えられる。

上記問題は、ある時点において、値が入力されたユニットと同じで、制約充足状況が同じユニットの状態相互は、同じ状況にあるという思想にもとづく汎化により解決できる。この思想は、ある状態において、それ以前に獲得された特徴的な情報が同一なら、同じ状況とみなすことを意味し、自然である。このように、汎化に制約を用いる本手法は、汎化に意味をもたせることが可能であり、アプリケーションに適した汎化基準を与えることができる。

5 まとめ

本稿では、ユーザフレンドリなデータエントリシステムの構築手法について提案した。本手法の特徴は、ユーザの操作履歴を制約の充足状態にもとづき汎化し、ナビゲーション戦略として利用することにある。その結果、陽に記述することが困難であった、ユーザにとって自然に感じられるナビゲーションを実現できる。また、学習機能によりそのシステムが使われる環境に適したナビゲーションが可能となる。

謝辞

本研究を進めるにあたり、活発な議論及び、インプリメントに協力頂いた豊橋技術科学大学 鍋島英知氏に深謝します。

参考文献

[1] M.ISHII, Y.SASAKI and S.KANEDA: Interactive Constraint Satisfaction for Office Systems, Proc. of the Tenth Conference on Artificial Intelligence for Applications, pp. 116-124 (1994).
 [2] 中野・小島・金田: 容易に知識修正ができるオフィス業務エキスパートシステム, 電子情報通信学会, 人工知能と知識処理研究会, AI91-80, pp. 49-56 (1992).