

勤務表設計支援のための事例による事例検索器の制御

1 D-3

○井上慶太郎, 峯崎俊哉, 成嶋弘  
東海大学

1 はじめに

40人程度の従業員が勤務する24時間営業のコンビニエンスストアの勤務表(図1)を設計/調整する支援システム(エキスパート・システムのプロトタイプ)を開発した。勤務表設計問題は、「ある従業員とその勤務時間を別の従業員で代行する。必要ならば勤務時間を複数の時間帯に分割し、それぞれの時間帯を別の従業員で代行する。」というような従業員の配置換え問題(部分問題)に変換され問題解決される等の問題の性質から事例ベース推論(以下、CBRと略す)を適用している。本システムは、事例の修正/修復に他の推論系を併用せず、事例にのみ依存することを考えた。そこで事例をその機能的役割に応じ3種類に分類した。更に、対話型CBRシステムの形式を取ることで、設計者の介入を随所で許し各種事例の自動獲得を実現している。本稿では、このシステムにおける事例による推論制御を考察する。

2 事例の分類と推論機構

2.1 定義

準備として次の項目を定義する。

区分: 従業員が所属している時間帯。

シフト: 勤務時間と従業員の対。

勤務表:

勤務時間を属性名とし、勤務する従業員の識別子の集合を属性値とする属性リスト。

シフトの特徴づけ:

代行する勤務時間帯、従業員の性別、代行する日付での出勤の有無とその勤務時間(出勤していない場合は未定義)の組。

領域知識:

性別、区分、曜日毎のスケジュール、日付毎のスケジュール、等の知識を各従業員について記述している。

事例を機能的な役割に応じ次のように分類する。

基本事例:

勤務された日付、曜日、区分、勤務表の組。

日付	1	2	3	...	31
氏名\曜日	火	水	木	...	木
細野美子		18-23	17-20	...	
村上由紀	17-20		17-20	...	
...	...	...	...	...	...
井上慶太郎		/	17-23	...	17-23
木村智久	20-23		20-23	...	17-23

※・表の成分I-Jは、従業員の勤務時間でI時からJ時まで勤務するという意味。  
・表の成分/は、登録されている区分での勤務が不可能であるという意味。

図1 勤務表の例

修正事例:

曜日、区分、代行されるシフト、代行するシフトの組。

修復事例:

曜日、区分、代行されるシフト、代行するシフトの特徴づけの集合の組。

2.2 勤務表設計支援過程

本システムを使用した勤務表の設計支援過程は図2の通りである。次に図の各処理の概要を説明する。

基本事例の分類:

問題の特徴づけ、これに類似している基本事例を分類する。

基本事例の提示:

分類した基本事例の集合から類似性の高いものから順に設計者に提示し、使用する基本事例を選択する。

修復事例検索・提示:

選択された基本事例に含まれる勤務不可能なシフトと照合する修復事例を検索する。検索されたものから順に設計者に提示し、使用する修復事例を選択する。

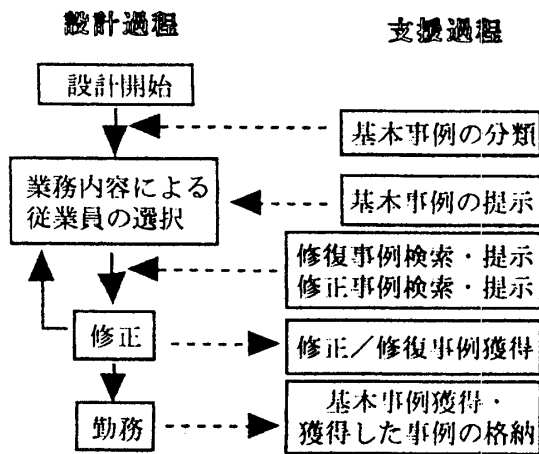


図2 勤務表設計支援過程

修正事例検索・提示：

修復事例の記述に従い、勤務時間の分割を行ない、修正事例を検索する。検索されたものから順に設計者に提示し、使用する修正事例を選択する。

修正/修復事例獲得：

使われた修復/修正事例を得る。

基本事例獲得・格納した事例の格納：

各種事例ベースに獲得した事例を格納する。

2. 3 推論機構

事例の修正/修復の準備として、次の検索を順に行なう。

検索1：基本事例に含まれている調整を必要とするシフトをキーに修復事例を検索する。

検索2：検索された修復事例がもつ代行するシフトの特徴記述とシフトをキーに修正事例を検索する。

このように検索された修正事例を勤務表に作用させる。この際、特徴記述に含まれる勤務時間の分割が、事例の修復案、従業員についての記述が修正案検索キーの役割を持つ。また、推論エンジンは勤務表に勤務不可能なシフトがなくなるまで、上記を繰り返す。(シフトについては完全照合、特徴記述については部分照合を用いて検索を行なう。)

3 評価

3. 1 設計効率について

設計についての時間的コストは、20人分の半月の勤務表を設計する際、熟練した設計者(就業して約5年の設計者)は手作業で約1.5時間を要する。初心者の設計者(就業して数ヶ月の設計者)が設計すると約2時間から3時間を要するところを、システムによる支援環境により初心者の設計者が設計しても約1時間強で済むようになった。

更に次のような条件のもとでの勤務表の調整率

設計方法 \ 条件	条件1	条件2
手作業による設計	20.4%	19.2%
システムの支援による設計	18.8%	17.9%

表1 調整率の比較

(設計されたシフトが調整される割合)を考える。勤務表に含まれる全てのシフトの集合Sとする時、  
条件1：Sから研修中の従業員を含むシフトを除く。

条件2：Sから研修中の従業員を含むシフトを除き更にある月において退社する従業員を含むシフトを除く。

とする。それぞれの条件下での調整率は、表1の通りである。これは、初心の設計者により「手作業による設計」と「システムの支援による設計」をそれぞれ約2ヶ月づつ行なった時の集計である。実験した期間が極めて短いため信頼性にかけるが、実用に期待もてる。また、設計者は設計している日付の勤務表とその近傍(前後1週間程度の日付の勤務表)の勤務表を同時に調整していることが分かった。しかし、インターフェースの簡略化するために、一度に作成できる勤務表の個数を1つに限定しことで、飛躍的な効率化ができなかった。

3. 2 処理の負担について

この実験に用いたシステムが持っている基本事例の個数は約100個、修復/修正事例の個数は約50個である。システムの処理の負担は基本事例の個数に比例して増える。しかしながら、修復・修正事例については個数が増えても、設計者に提示する候補の数が増えるだけでシステムには負担がかからない。また、修正に使用する事例検索の際、事例全体の類似性の評価を避け修正される成分にのみ着目して照合処理をしているため、事例修正/修復の際のシステムの負担を小さくできている。

3. 3 事例の自動獲得について

本システムは、勤務表の質の評価をせず、手当たりしだいに事例を獲得することから、事例ベースの自己組織化は実現できているが、質の悪い事例も獲得してしまう。

4 おわりに

事例のもつ2面性(エピソード的記憶とインスタンス・ルールの側面)を積極的に利用してCBRの制御を試みた。本システムでは、勤務表が満足してはならないような制約(例えば、勤務に当たる従業員の中に少なくともひとりとは男性が含まれる等)を、システムは全く考慮していないため、設計者が責任を持つことになる。しかし、より質の良い勤務表を設計するには、今後システム内に展開する必要がある。