

# ポリシー記述を用いたスケジュール自動調整ツール

2M-4

高田 一雄 塚本 昌彦 西尾 章治郎  
大阪大学大学院 工学研究科 情報システム工学専攻

## 1 はじめに

近年、多くのソフトウェアの規模は巨大となり、その構造もより複雑になってきた。そのため、運用を行なう際には、ユーザのもつ選好傾向や利用する環境に応じた運用の制限などを、実際の動作に自動的に反映させる仕組みが要求されている。この点に着目して、筆者らの研究グループではこれまでに、ソフトウェアの管理者や利用者の持つ運用方針をポリシーとして捉え、これを自動的にソフトウェア動作へ反映させるためのポリシー記述言語の枠組みを提案し、インタードメインルーティングプロトコルに適用してシステムの検証を行なってきた [1]。

一方、個人ごとに複雑な運用が要求されるアプリケーションとして、スケジュール管理ツールが挙げられる。同じ時間帯に重複しているスケジュールがいくつも存在し、それらを全て調整しなければならない場合、あるいは、週に一度会議を入れたいが、何曜日の何時から行なうのが最も好ましいかを判断したい場合などに、ユーザのもつ取舍選択の調整方針に従って既存のスケジュールを組み直し、推奨される新たな予定を自動的に提示するような機能は有用であると考えられる。

そこで本稿では、文献 [1] で提案したポリシー記述言語を拡張し、これをスケジュール調整に適用した結果について述べる。

## 2 ポリシー記述言語の概要

文献 [1] で提案した言語では、ユーザのもつポリシーを、ポリシー構文と IS-A 関係に基づいた知識文の 2 つの記述で表現する。

ポリシーは、対象の優先順序を指定する prefer 文と、対象の使用を拒否する prohibit 文という 2 つの構文により記述する。

• prefer <object(1)> to <object(2)>

A Schedule Coordination Tool based on User's Policy Descriptions

Kazuo TAKADA, Masahiko TSUKAMOTO, Shojiro NISHIO  
Department of Information Systems Engineering, Graduate School of Engineering, Osaka University

[with <weight>] [when <condition>]

• prohibit <object> [when <condition>]

一方、知識文は、対象に固有の属性や対象間の関係（グループ関係や階層関係）などを、IS-A 関係式によって記述する。この知識文を予め大量に用意しておくことにより、ユーザが具体的な対象ごとにポリシーを全て列挙する必要を無くす。例えば「会議はなるべく午後の時間帯に行なう」というような抽象的で簡潔なポリシー記述を許すようにしている。

## 3 ポリシー記述言語の拡張

スケジューラでは、いくつかの特有のポリシー表現が必要となる。本研究では、スケジューラ以外のアプリケーションに適用するための汎用性を考慮し、ポリシー記述言語に次のような拡張を行なった。

• 関数処理 Function(X)

今回のシステムでは TimeZone(T1~T2) 関数を用意し、時刻 T1 から T2 までの値を、例えば 9:00, 9:01 などのように、細分化した離散値として返すようにしている。

• 同義語の扱い JOB equals WORK

複数人による知識のメンテナンスを容易にするために、同義語を記述する。

• 知識文の自動追加 \*研究会 is-a 研究会

ユーザが新規スケジュールを入力した場合に、自動的に最低限の知識が追加されるように記述を行なう。

### 3.1 ポリシー記述例

図 1 に、ある学生がこのスケジューラを利用することを想定した場合のポリシー表現の例を示す。

## 4 スケジュール調整の仕組み

全体のシステム構成を図 2 に示す。

ポリシー構文や知識文が更新されている場合に、まずスケジューラはそれらをコンパイルして、以下のような形式の線形の評価関数を生成する [1, 2]。

$$f(x) = k_1 T(x_{object}) + k_2 T(x_{attr1}) + k_3 T(x_{attr2}) + \dots$$

これは対象の優先度を数値で返す関数であり、今回のシステムでは、図 1 で例示したような、スケジュー

**【仕事に関するポリシー】**

- アルバイト よりも ミーティング を 必ず 優先させる  
prefer MEETING to ARBEIT with Certainty
- 依頼人 が 学生 の 仕事 よりも 依頼人 が 教授 の 仕事 を かなり 優先させる  
prefer JOB where EMPLOYER is-a PROFESSOR to JOB where EMPLOYER is-a STUDENT with Intensity

**【スケジュールを入れる時間帯に関するポリシー】**

- 時間帯 (17:00~) よりも 時間帯 (9:00~17:00) を なるべく 優先させる  
prefer TimeZone(9:00~17:00) to TimeZone(17:00~) with Possibility
- 仕事 が ミーティング の 場合, 休日 や 時間帯 (20:00~) は 絶対 に しない  
prohibit WEEKDAY and TimeZone(20:00~) when JOB is-a MEETING

**【仕事や人物に関する知識】**

- T 氏 は N 研究室 の 学生  
T is-a N-LAB.STUDENT
- T 氏 の 所属 チーム は 知識処理 チーム  
T.TEAM is-a KP-TEAM
- 知識処理 チーム の ミーティング は N 研究室 の 行事  
KP-TEAM.MEETING is-a N-LAB.WORK
- ミーティング の 議長 は 教授  
MEETING.CHAIRMAN is-a PROFESSOR

図1: ポリシー記述例

ルとして新たに追加する仕事自身に関わるポリシーと、追加を行なう時間帯に関するポリシーとを反映した2つの評価関数が生成される。

スケジュールの調整は、スケジュールを追加する範囲をユーザが大まかに指定し、システム側がその時間帯をある決まった間隔（実装では5分ずつ）で分割してそれぞれの優先度の大小を比較することによって行なう。例えば、30分間の新たなスケジュールを追加する場合、システムは追加範囲を5分ずつに分割し、それぞれの時間帯のうち、前後合わせて6つの区間（ $30 \div 5$ ）において足し合わせた優先度が最大となる区間をスケジュールの候補とする。

### 5 実装

スケジューラの実装には、プログラミング言語としてJavaを用いた。実行画面の一例を図3に示す。

いくつかのスケジュールが入力されている状態で、ユーザがそれらの調整を指示すると、システムは重複しているスケジュールを全て検出し、それらを組み直

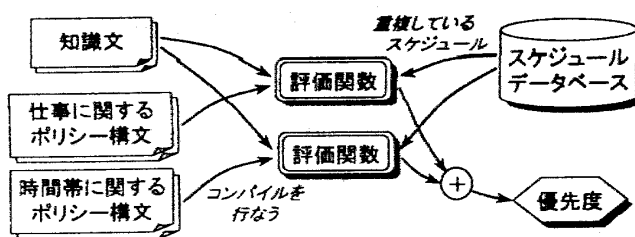


図2: システム構成

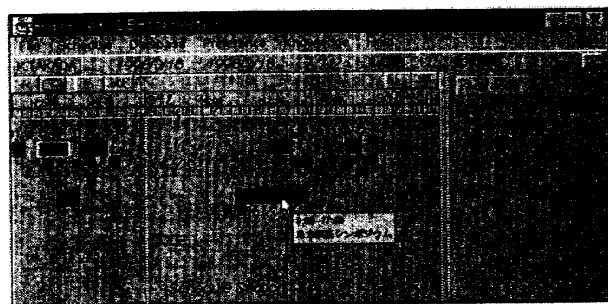


図3: スケジューラの実行画面

した新たな予定をユーザに対して提示する。ユーザがそれに満足しなければ、次に優先度の高い候補を表示させたり、あるいは手動で部分的に修正を行ったりすることが可能となっている。

このスケジューラを複数人に試用してもらったところ、概ね満足できるスケジュール候補が導出されるとの回答を得た。このスケジューラは、スケジュール調整のためのエキスパートシステムではないため、ユーザの望む最適なスケジュール配置を常に提示できるわけではないが、いくつかの候補を自動的に算出してユーザのスケジュール作成を助ける機能に関しては、良い評価が得られた。

### 6 おわりに

今回実現したシステムでは、運用のために最低限必要とされる知識文やポリシー構文を、システム設計者が予め見極めて提供しておく必要があり、円滑な運用を実際に始めるまでに手間がかかることが問題となっており、今後の検討が必要である。また、運用開始後の知識のメンテナンスが煩雑であるとの指摘も受けており、この点についても改めて検討を行なう必要がある。

### 参考文献

- [1] 高田一雄, 塚本昌彦, 西尾章治郎, “ポリシー記述言語を用いたソフトウェア運用の実現,” 情報処理学会第53回全国大会講演論文集 (1), pp.237-238, Sep. 1996.
- [2] 高田一雄, 塚本昌彦, 西尾章治郎, “ソフトウェア自動運用のためのポリシー記述言語,” 人工知能学会誌第13巻3号, May. 1998 (掲載予定).