

Supporting Combined Human and Machine Planning: An Interface for Planning by Analogical Reasoning

Michael T. Cox, Manuela M. Veloso: Supporting Combined Human and Machine Planning: An Interface for Planning by Analogical Reasoning, Case-Based Reasoning Research and Development: 2nd International Conference on Case-Based Reasoning, pp.531-540 (1997).

現実的で複雑なプランニングの状況では、人間の・自動化されたプランナーがお互いに望むプランを構築するために相互作用する、主導権が混交したプランニングのフレームワークが必要である。理想的には、この共同の協調には、人間や計算機が一人で生成できるものより、良いプランに到達できる可能性がある。人間のプランナーはよく、過去の経験に頼り、過去のプランニング事例を検索して適用する、事例に基づくプランニングのアプローチを取る。Prodigy/Analogyで行われる、生成的プランニングと事例に基づくプランニングが組み合わされた類推によるプランニングは、主導権が混交した統合を研究するための最適なフレームワークを与える。しかし、人間のユーザにこのプランニングのループに従事させることは、様々な新しい研究課題を生成する。我々が発見した、主導権が混交するプランニングシステムを生成する挑戦は、プランニングパラダイムは人間と計算機のプランニングにおいて異なる、プランとそのプロセスの可視化は複雑だが必要である、人間のユーザはプランニングの領域と技術の双方に関して経験の範囲を越えて及ぶ、という3つのカテゴリになる。本稿は、Prodigy/Analogyでの類推による、プランニングのプロセスに人間を組み込みためのインタフェースの設計時の、3つの問題へのアプローチを示す。このインタフェースは、ユーザに生成的プランニングと事例に基づくプランニングの両方に従うことを許し、ユーザに情報の提示のコントロールを許すことによる、ユーザの経験における不一致に近接する。

1 Introduction

□ 主導権が混交するプランニング (mixed-initiative planning)

- 計算機システムと人間のプランニングのインタラクション
 - システム・人間単体以上のパフォーマンス (理想的には) [Ferguson 1996, Oates 1994]
 - 二者間の不均衡の問題
- 本アプローチの3大課題
 - 人間とシステムのプランニングパラダイムの相違点
 - プランの可視化
 - 人間ユーザの異なったプランニングの領域・技術
- PRODIGY プランニングシステム
 - 主導権が混交するインタフェース設計時の問題点への対処

□ 3大課題

- 人間・システムのプランニングの統合
 - AI: モデルを仮定・検索によるプラン生成
 - 人間: 事例に基づく [Hammond 1989]
 - : 人間は経験のないことをうまくできない [Riesbeck 1989]
 - : 類似経験を使用した新しい解の形成 [Kolodner 1993, Seifert 1994]→ Prodigy/Analogy: 生成的アルゴリズム・事例に基づくアルゴリズムの併用
- プランの可視化
 - 行動の系列
 - : 各行動とゴール・状態との関係が複雑
 - 目標木表現 (goal tree representation)
 - : プラン構造の発見・具体化を支援
 - : 行動の選択理由を表現
- ユーザごとの経験の違い
 - プランニング領域の知識とスキルの違い・プランニング技術の違い
 - ブラックボックス (⇔ ガラスボックス [Veloso 1995])
 - : 技術的・実装上の情報を隠蔽
 - : 選択的コントロール機構 [Cox 1997], 初心者からエキスパートを支援

□ 主導権が混交するインタフェース

- 生成的・類推的プランニングの切り替え

- 事例ライブラリへの事例格納・検索（手動・自動）
- 目標木によるプラン提示
- 提示情報のレベルのカスタマイズ

□論文の構成

- 2 節：PRODIGY の生成的・類推モードの制御アルゴリズム
- 3 節：PRODIGY の主導権が混交するインタフェース
- 4 節：ユーザの熟達度に合わせた反応（を簡単に）
- 5 節：まとめ

2 PRODIGY: A Hybrid of Generative and Case-Based Planning

□ PRODIGY：自動化プランナー

- 生成的な状態空間プランニング・事例に基づくプランニングの結合
- Prodigy4.0
 - オペレータ空間の検索
- Prodigy/Analogy
 - 事例ライブラリの検索

2.1 The Generative Planning Algorithm

- Prodigy4.0[Veloso 1995]
 - 状態空間を持つプランニングシステム
:手段目的解析・後向き推論検索
- 領域理論の構成（対象領域の問題・知識の表現？）
 - オブジェクトとオペレータの階層
 - オブジェクトの状態を変更する推論ルール
- 計画問題表現
 - 初期状態（オブジェクトと述語表現）
 - ゴール表現の集合
 - プランニングの決定
- 制御ルール
 - 問題解決エピソード（事例）／領域一般ヒューリスティックス

- | |
|---|
| <ol style="list-style-type: none"> 1. 初期化 2. 目標状態が満足されていれば終了 3. どの目標が未解決か決定 4. 現在の状態において予条件を満たすオペレータが選択されているか決定 5. ゴールをサブゴール化するかオペレータを適用
サブゴール化なら 6, 適用なら 7 6. 未解決の目標・達成可能なオペレータを選択, 3 へ 7. 適用可能なオペレータによって状態を変更, 2 へ |
|---|

Fig.1 Prodigy4.0 のプランニングアルゴリズムのトップレベルビュー

- プランニングアルゴリズム（→ Fig.1）
 - ゴール・オペレータの選択, オペレータの具体化の系列
 - オペレータの現在の状態への「適用」・未解決のゴールの「サブゴール化」
→ ヘッドプラン（プラン実行のシミュレーション）とテールプラン（ゴール指向推論）の組み合わせ
- 事例（プランニングのエピソード）
 - 決定の正当化を解法とともに事例ライブラリに（ユーザが？）保存
 - 問題のゴール・初期状態の部分集合により特徴づけ

2.2 2.2 The Derivational Analogy(CBR) Planning Algorithm

- Prodigy/Analogy
 - プラン生成・プランニングエピソードの解釈と蓄積・過去のプランの検索と再利用
 - 注釈を付加してプランを蓄積
 - 過去の決定プロセスに基づく新しい解の導出
(過去の解の適用ではない)

1. 初期化
2. 事例の選択
3. 過去の事例から関連するオペレータを獲得
4. 過去の失敗があれば現在の検索パスから失敗を枝切り
5. 現在の状態とのマッチングによりオペレータの syntactic な適用可能性をチェック
6. 過去の理由が保持されるか決定することで semantic な適用可能性をチェック
7. 選択が妥当でなければ、適切なアクションを選択
他のプランニングが必要ならば事例のガイドを中止
追加事例を検索
領域理論を用いて再度プランニング
必要でない過去のプランニングの事例のガイドを進める
他のガイド事例を選択することで注意の焦点を変更

Fig.2 事例に基づくプランニングアルゴリズムのトップレベルビュー

- 誘導類推のアルゴリズム (→ Fig.2)
 - 現在の問題をカバーする事例 (問題解決エピソード) を1つ以上検索
 - ガイドの情報源として使用
- 汎用的リプレイ機構
 - 過去の事例における正当化構造を現在の問題文脈で解釈
- 解の生成
 - ガイドする事例の集合から解を生成
 - 類似事例からのガイドをマージする方略

3 Visualization of the Plan and the Planning Process

- 目標：アルゴリズム実行の可視化
- Tcl/Tk + Motif
 - :Tool Command Language/Tool Kit
 - Tcl は C 言語に組み込むコマンド言語として開発されたもの。Tk は GUI 拡張パッケージ
 - UI・プランナーは独立プロセスで通信
 - PRODIGY は Lisp・CUI インタフェース
- GUI (→ Fig.3)
 - 上：コントローラ・主表示，下：2 事例を提示
 - Load：データ読み込み，Replay：プランナー実行，Step：逐次実行
 - 実行途中でインタラクション可能
 - 状態表示
 - :テールプラン (左)，ヘッドプラン (右)
- ゴール木表示 (Fig3. 左)
 - 問題領域：打ち上げロケット
 - 2つのトップゴールの解決で終了
 - :オブジェクト 1・2 が locb に配置
 - :オペレータ unload-rocket を 2 回実行して解決
 - 2 事例を併用
 - :**：使用されたステップ
- 生成的・事例に基づくプランニングモードの切り替え
 - Planning Mode

- 手動検索 (→ Fig.4)
- 木のノードからの情報提示 (→ Fig.5)
 - 木ノードからオペレータ・ゴール選択の正当化を表示
 - オペレータの定義を表示
- ノードの解釈情報 (→ Fig.6)
 - 人間・システムによる解釈
 - :ある決定の失敗理由の例 (Fig.6)
 - 失敗を検索パスの枝切りに使用

4 Expertise in Planning Versus Expertise in Tecnology

- システムの対象
 - プランニングシステムに詳しいユーザ向けに最適化
 - 内部メカニズム・データ表現・プラン構造を詳細表示
 - 初心者への促進の検討中
- プランニング形成知識のないユーザ支援 [Veloso 1998]
 - ForMAT 兵力配置プランナー
 - 事例からの適用をアドバイス
 - Prodigy/Analogy を意識しないリモート利用
- 初心者モード
 - 自然言語形式の出力

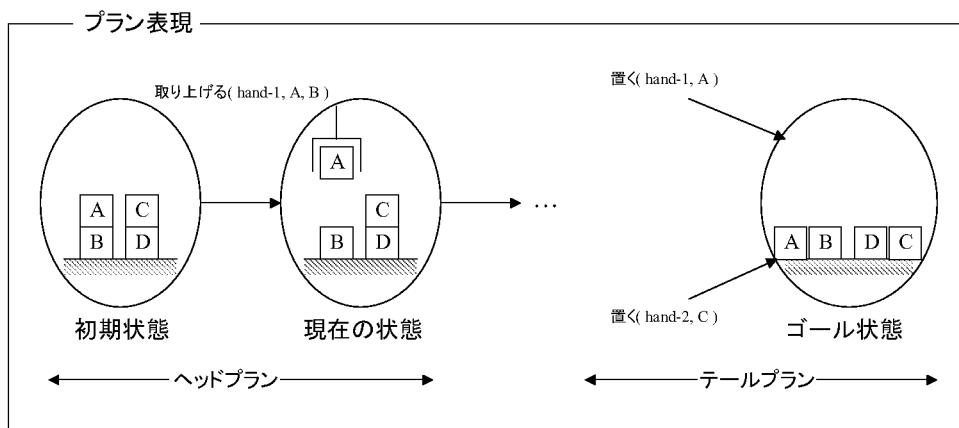
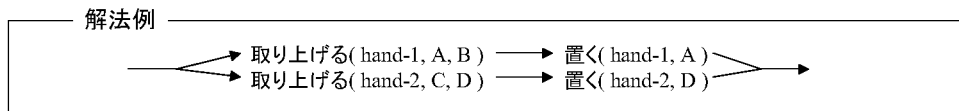
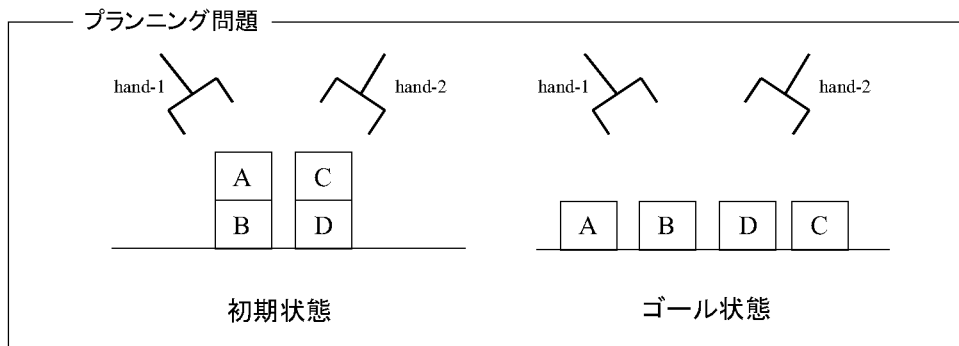
5 Conclusion

- 主導権が混交するユーザインタフェース
 - 生成的・事例に基づくモードの切り替え
 - 事例の保存と検索・再利用
 - ゴール木構造表示
- 目標・展望
 - 統合システム上での人間・システムのインタラクション促進
 - ユーザのしているもの・求めるもの・行うことへの焦点
 - :インタフェース・目標・タスク

私見的注とコメント

- プランニング問題
 オブジェクトの状態に関する初期状態とゴール状態が与えられ、オペレータによってオブジェクトの状態を変更しながら、初期状態をゴール状態に変更する問題、といったところか。

オペレータ			
取り上げる(hand, x, y)	載せる(hand, x, y)	持ち上げる(hand, x)	置く(hand, x)
予条件	予条件	予条件	予条件
上にある(x, y)	手の中にある(x)	床にある(x)	手の中にある(x)
上に何も無い(x)	上に何も無い(y)	上に何も無い(x)	追加
何も持っていない(hand)	追加	何も持っていない(hand)	床にある(x)
追加	上にある(x, y)	追加	上に何も無い(x)
手の中にある(x)	上に何も無い(x)	手の中にある(x)	何も持っていない(hand)
上に何も無い(y)	何も持っていない(hand)	削除	削除
削除	削除	床にある(x)	手の中にある(x)
上にある(x, y)	手の中にある(x)	上に何も無い(x)	
上に何も無い(x)	上に何も無い(y)	何も持っていない(hand)	
何も持っていない(hand)			



プランニングタスクのベタな例 ([Fink 1994] より)

- システムとユーザのインタラクション
通常、自動化システムは実行可能な処理のみを行い、常識知識やプロダクト評価等、計算機にはどうしても困難であることは「人間のユーザに補ってもらおう」という言い方をする。しかし、この論文では積極的に、「人間とシステムとのインタラクション」は「人間だけで・システムだけ」より（理想的には）優れたパフォーマンスを示す、と述べている。
- 人間と AI システムの違い
人間は過去の経験を利用し（高質・低量？）、AI システムは網羅的探索を実行する（低質・多量？）、とある。人間とシステムの特技や特性がうまくインタラクションする良い、ということか？
- 誘導類推
PRODIGY は、事例の利用において、解ではなく、事例の解法ステップを利用している。（事例を「エピソード」と呼ぶのは、このような背景から？）
CBR では通常、事例の解（過去の問題解決のプロダクト）を直接書き換え、現在の問題の解を得る。これに対し、誘導類推は事例の解法導出過程（プロセス）を拝借し、その解法を現在の問題にあてはめて解を再導出する。
- 自分のシステムでは、プロダクトからプロセスを取り出してエピソード化している、と言えるか？