

ソフトウェアエージェントにおけるプランニング

Planning for software agents

山田 誠二
YAMADA Seiji

東京工業大学 大学院総合理工学研究科知能システム科学専攻
CISS, IGSSE, Tokyo Institute of Technology
yamada@ymd.dis.titech.ac.jp, <http://www.ymd.dis.titech.ac.jp/~yamada/>

keywords: Software agent, planning, information gathering, life-like agent

1. はじめに

オフィスアプリケーション上でキャラクタが現れてユーザを支援したり、インターネットで情報を収集したりするソフトウェアエージェントが各種開発されている。残念ながら、それらのエージェントの多くは、最初から作り込まれた手続きをその通りに実行するにすぎない。しかし、タスクが多様かつ高度になるにつれて、目標に応じて適切で複雑な行動計画を自動的に生成して実行する機能が重要になってくる。そして、AIにおけるプランニングがそのような機能実現のための要素技術になることは明らかである。本稿では、プランニングの応用されたソフトウェアエージェント研究の事例を紹介し、そこで生じる課題について議論する。

2. インターネットエージェントにおけるプランニング

インターネットで情報収集などのタスクを実行するインターネットエージェントはさまざまなものが研究されている。ここでは、プランニングを行動決定機能とするインターネットエージェントを紹介する。

2.1 SoftBot

SoftBot[Etzioni 94]^{*1} は、ユーザが「インターネット上のある情報を収集したい」という要求がある場合に、その集めるべき情報を目標として入力すると、その情報収集を実現するための実行可能なコマンド系列（例えば、UNIXのコマンド系列）をプランニングする情報収集エージェントである。ワシントン大学の Etzioni らのグループにおいて開発され、UNIX シェルと WWW を使って世界中のインターネット上の情報資源を収集することができる。

```
(forall (?d :in files)
  (if (and (file.type ?d memo.document)
          (subject.of.document ?d "budget")
          (not (string.in.file "draft" ?d)))
      (delivered.to ?d ?obj341)))
```

図 1 SoftBot の扱う目標 [Etzioni 94]

§1 目標の入力

ユーザは、一階述語論理 [馬場口 99] の一部を使って目標を記述し、SoftBot に入力する。この記述言語は、論理積、論理和、全称記号が使える、ある程度複雑な目標も書ける。しかし、一般ユーザが、必ずしも論理的記述に慣れているわけではないので、メニュー形式で項目を選択することにより、簡単に目標を入力できるようになっている。また、メニューによる入力は、SoftBot の目標に、図 1 のような論理式で記述された目標に翻訳される。

§2 命令における曖昧性の解消

ここでは一例として、“A 社の山田さんへ予算メモを送れ” というタスクを考える。このタスクに対し、既存のソフトウェアエージェントのほとんどは、その命令が含む多くの曖昧性をうまく処理できない。

基本的に SoftBot は、与えられた記述とマッチングできるすべての個人や対象をインターネット上で見つけようとする。ユーザが、山田さんの所属を省略した場合、SoftBot はインターネット中の山田さんを探そうとするため、この対象の特定は非常に大きなコストがかかる場合がある。よって、SoftBot は、曖昧性解消を実行する前に、その解消に要するコストを見積もり、それが十分に高いと自力で解消することをあきらめて、ユーザーにさらなる情報を問い合わせるという戦略をとる。

§3 情報収集手続きのプランニング

SoftBot は、コマンド系列の自動生成のために、古典的プランニング [馬場口 99] を用いている。SoftBot で用いられるオペレータの例を図 2 に示す。

SoftBot のプランナーは、プランをノードとした探索空間

*1 <http://www.cs.washington.edu/research/projects/softbots/www/softbots.html>

```

オペレータ名 : (netfind ?person)
条件リスト :
  (currnet.shell csh)
  (isa netfind.server ?server)
  (firsrtname ?person ?firstname)
  (lastname ?person ?lastname)
  (or (person.city ?person ?keyword)
      (person.institution ?person ?keyword))
効果リスト :
  (userid ?person userid)
  (person.machine ?person machine)

```

図 2 netfind のオペレータ [Etzioni 94] (?は変数)

で、半順序プランを探索する最小決定プランニング (least commitment planning) [Weld 94] を用いている。このプランニングアルゴリズムは、目標を副目標 (sub-goal) に分割することができ、また独立に達成できない副目標間の干渉を上手く処理できる。

また、プランが存在すれば必ず見つけるという完全性 (completeness) と、見つけたプランを実行すれば必ず目標を達成できるという健全性 (soundness) の両方の性質を SoftBot のプランナーは満たしている。しかし、残念ながらこれらの理論的な保証は、実際問題で重要である効率については言及しない。完全性は、どのぐらい時間がかかるかはわからないが、必ずプランを見つけないという意味であり、目標の複雑さに対してどの程度の時間でプランを見つかるかについては何も保証しない。よって、SoftBot のプランナーは、バイアス (bias) [馬場口 99] と呼ばれる探索を制御するヒューリスティックを使い探索を制限する。このヒューリスティックは、人間が記述してもよいし、機械学習を用いて自動的に学習することも考えられる。

§ 4 プランニングによる情報資源の統合的利用

宣言的表現であるオペレータにより、SoftBot は、複数で独立したインターネット資源を統合的に利用できる。例えば、図 2 の netfind オペレータでは、その条件リスト中の条件の一つである (person.institution ?person ?keyword) を見ることににより、SoftBot は、netfind を使うためには、その人物の所属機関を知る必要があることがわかる。よって、他のコマンド (grep による検索など) を使って、その情報を検索することをあらたな副目標とする。また、インターネット資源に対する新しいコマンドを SoftBot が利用できるようにするには、そのコマンドのオペレータを記述して、追加するだけでよい。

2.2 Plangent : プランニングモバイルエージェント

Plangent [大須賀 00]*2 は、プランニング機能とモバイル機能の両方を備えたソフトウェアエージェントである。Plangent は、通常のプランニング機能に加えて、再プランニングによる環境変化への適応という特徴がある。

プランニングをほとんど行わない即応プランニング (reactive planning) [馬場口 99] は、環境変化に追従できるという特長をもつ。しかし、古典的プランニングでも、一度作ったプランが環境変化などにより実行できなくなった場合、その変化した環境を初期状態として、もう一度プランニングをやり直すことで、環境変化への対応は可能な場合がある程度可能である。このような再プランニング (replanning) は、手続き全体を固定的に記述する通常のプログラムより、柔軟な行動を実現できる。

一方、モバイルエージェント (mobile agent) は、「計算機ネットワーク上にある計算機から他の計算機に移動する能力をもつプログラム」のことであり、ネットワークでつながっている離れた計算機における遠隔実行、非同期実行などのメリットをもつ。

Plangent は、プランニングを行なうモバイルエージェント、つまりプランニングモバイルエージェントである。プランニングモバイルエージェントとは、目標を達成するための手段を自分で考えながら計算機ネットワーク上を移動して、処理を実行するソフトウェアエージェントの枠組みであり、プランニングエージェントとモバイルエージェントの両方の機能を備えている。プランニングモバイルエージェントの具体的なタスクは、ネットワーク上を移動しながら、情報収集することなどがある。

また、Plangent は、プランニングとして、古典的プランニングを採用している。古典的プランニングは、適切なプランの生成及びその実行が可能であること、Plangent の環境である計算機ネットワークが、即応プランニングを必要とするほどの急激な変化を伴うものではないことなどの理由による。

§ 1 Plangent の働き

Plangent は、ユーザから目標を受け取ると、以下のような手続きでそれを処理する。Plangent の主なタスクは、情報収集である。

- (1) プランニング : ユーザから受け取った目標を達成できるプランを生成する。
- (2) ネットワーク上の移動 : 生成されたプランを基に、処理に必要な情報あるいはサービスがある計算機までネットワークを通して移動する。
- (3) プラン実行 : 移動先の計算機で、そこにある情報やサービスを利用してプランを実行する。
- (4) 再プランニング : プランニング時には予測できなかった環境の変化があった場合に、プランの実行は失敗する。そのような場合、変化後の環境を基にして、プランニングをやり直すことにより対処する。

*2 http://www2.toshiba.co.jp/plangent/index_j.htm

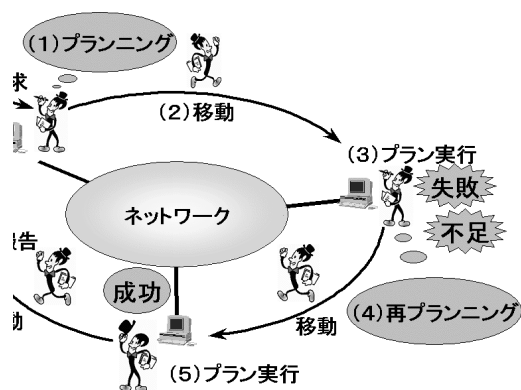


図 3 Plangent の動作 [大須賀 00]

(5) 結果の提示：最後に元の計算機に戻って、命令を
発したユーザに結果を提示する。

以上の手続きを図示したものが、図 3 である。

§ 2 Plangent によるエージェント開発

Plangent は、プランニングを実行するために必要な知識とエージェント自身の挙動を記述することにより、プログラミングされる。これらの内容を対象領域に依って変更することで、様々な分野で Plangent のアプリケーションを開発できる。

Plangent は、アクション定義、スクリプト定義、そして、拡張コマンド定義の 3 つの形式により記述される。

i. アクション定義

エージェントがある行為を実行するために必要な条件とその実行の結果生じる環境変化が記述される。古典的プランニングのオペレータに対応する。具体的には、以下のような記述になる。

```
action(アクション名, , ,
      [スクリプト呼出し]
      [事前条件], [事後条件], [ ] )
```

事前条件は、そのアクションが実行するために必要な条件であり、この事前条件が成り立った時に、スクリプト呼出しに記述されているスクリプトが実行される。そして、事後条件は、そのスクリプトが実行されることにより生じる状態を記述している。

ii. スクリプト定義

スクリプト呼出しにより、実行される処理の詳細がスクリプト定義において記述されている。Plangent では、このスクリプト定義の記述のために、PlangentScript という独自のスクリプト言語が用意されている。Plangent は、このスクリプト定義に基づきプランのプランを実行するが、実行中に何らかの問題が生じた場合の対処手続きを記述したり、実行を中断して制御をプランニングに渡すことなども可能である。PlangentScript における組み込みのコマンドの一部を、以下に示す。

- goto : エージェントの移動を実行する。
- newgoal : 新しい目標を生成する。
- subgoal : タスクを副目標に展開する。

また、制御文として、if, while, do-while, for, try-catch などが用意されている。例えば、try-catch により、ある範囲内でスクリプトの実行が失敗した場合の対処手続きを記述できる。

iii. 拡張コマンド定義

組み込みのコマンドでは、実現できないような処理は、拡張コマンド定義において記述する。拡張コマンドは、Java のプログラム形式で記述可能であり、組み込みコマンドと同じように、スクリプト定義から呼び出すことができる。

§ 3 再プランニング

Plangent の特長の一つである再プランニングは、素直にプランニングをやり直す方法である。不測の事態として、プランニング中に利用不可能なアクションが見つかった場合、プランニング中あるいはプランの実行中に、初期状態が誤っていることが判明した場合などに、再プランニングにより対処できる。これらの事態に対し、具体的には、以下の方針で再プランニングが行なわれる。

- プランニング中に利用不可能なアクションが見つかった場合：ある適用可能なアクション定義が使用できなくなれば、別の適用可能なアクションを探して、プランニングをやり直す。
- プランニング/プランの実行中に、初期状態が誤っていることが判明した場合：正しい初期状態を元に、最初からプランニングをやり直す。環境が動的な場合、環境変化によりこのようなケースが頻繁に起こり得る。

なお、Plangent の枠組みで、旅行計画支援エージェント、会議スケジュール調整エージェント、ドライバー情報支援エージェント (InfoMirror) などのさまざまなアプリケーションが開発されている。

3. 擬人化エージェントにおけるプランニング

擬人化エージェント (life-like agent) は、人間や生物の外見をもち、ユーザとインタラクトするエージェントである。擬人化エージェントの実現においては、ユーザがエージェントをいかに人間のように信頼できるようになるかが重要な課題である。そのため、感情モデルをもたせるなどの様々なアプローチが行なわれているが、決まりきったタスク処理だけでなく、さまざまな目標に柔軟に対応できるプランニングの機能も、擬人化エージェントに応用されている。ここでは、そのような擬人化エージェントの研究例として、ユーザへの効果的な情報提示を行なう PPP Persona とそこにおけるプランニングを紹介する。



図 4 Web ブラウザ上の PPP Persona[André 98a]

3.1 PPP Persona : プレゼンテーションするエージェント

PPP Persona (Personalized Plan-based Presenter) [André 98b, André 98a]^{*3} は、ウインドウや Web ブラウザの上に現れて、Web の情報源やデータベースから情報を収集して再構成し、それを人が人にコミュニケーションするようにユーザに提示する擬人化エージェントである。PPP Persona の研究では、ユーザと音声などで双方向にコミュニケーションするのではなく、ユーザへいかに自然に効果的に情報を提示するかというヒューマンインタフェースに主眼がおかれている。

図 4 は、Web ブラウザに現れた PPP Persona である。休暇を過ごす場所を探しているユーザに、適当なコテージを紹介している。この図は、特にコテージのテラスについて説明をしているシーンであり、説明対象を指示しながら音声合成を用いた発話による説明を行なう。さらに、この図をよく見ると、PPP Persona は、説明対象の方を向いている。このように、顔が対象部品の方を向くように頭を動かして、ユーザの注視をうながすことができる。

§1 プレゼンテーションスクリプトのプランニング

PPP Persona は、基本的にプレゼンテーションスクリプトという手続きを基に、ユーザに情報提示を行なう。このプレゼンテーションスクリプトを手で作成することには限界があるため、その自動生成をプランニングに

より実現する。このプランニングでは、プレゼンテーション戦略と呼ばれる設計知識を記述したオペレータを用いる。具体的な例を以下に示す。

プレゼンテーション戦略

ヘッダ : (Introduce Per User ?object ?window)

適用条件 : ((posy = 1))

(Bel Persona (ISA ?object Physical-Object))

下位行動 :

((A1 (Make-Window Per User ?object ?window)

(A2 (S-Show-Window Per User ?window ?object))

(A3 (Elaborate-Parts Per User ?window ?window))

(A4 (S-Wait Persona User)))

定性制約 : ((A1 meets A2)(A3 starts A2)

(A3 meets A4)(A4 finishes A2))

定量制約 : ((10 ≤ DurationA2) (2 ≤ DurationA4 ≤ 2))

開始 : A1

終了 : A2

プレゼンテーション戦略では、通常のエージェントの内容である適用条件と、その戦略の達成に必要な下位行動、そして定量的な時間制約と Allen 流 [Allen 83] の時制論理 (temporal logic) で記述された定性的な時間的制約が記述されている。ただし、各行動の実行時刻は事前にはきっちりと決められることはなく、行動の実行時に初めて決まるようになっている。上の例でも、A2 の最低期間と A4 の期間が 2 であるという制約のみになっている。

このようなプレゼンテーション戦略をもとに、PPP Persona では、階層的プランニング (hierarchical planning) と時制推論システムを組合わせて、基本的には後ろ向き探索でプレゼンテーションスクリプトを自動生成する。その手続きの概要を以下に示す。

- (1) すべての副目標が、基本行為に展開されたていれば終了。ただし、ユーザとのインタラクションを考慮して、一部の副目標は展開されずに残り、マウスクリック可能なアイコンとしてプレゼンテーション時に表示される。
- (2) プレゼンテーション目標とマッチするプレゼンテーション戦略を一つ見つけて、その下位行動を新たな副目標とする。
- (3) それぞれの副目標について、適用されたプレゼンテーション戦略のすべての定性/定量制約からなる局所的な時制制約ネットワークをつくる。
- (4) 副目標が満たされないか、時制制約ネットワークが矛盾する場合は、別のマッチングするプレゼンテーション戦略を探すため、(2) へ。そうでなければ、プレゼンテーション目標を副目標で更新して、(2) へ。

*3 <http://www.dfki.de/~andre/>

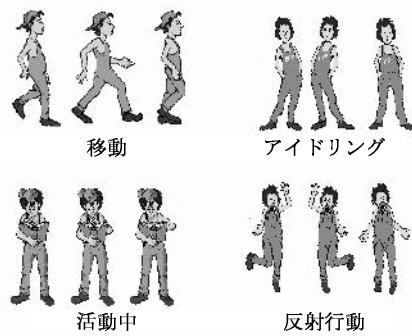


図 5 PPP Persona の自己行動 [André 98b]

§ 2 ペルソナエンジンによるプレゼンテーションの実行

次に、生成されたプレゼンテーションスクリプトを入力として、実際の PPP Persona の行動、つまり実際のプレゼンテーションをペルソナエンジンが実行する。このとき、PPP Persona の行動は、プレゼンテーションスクリプトのみで決定されるのではなく、エージェントの自己行動 (self-behavior) により決定されるものがある。自己行動とは、プレゼンテーションスクリプトとは独立した、エージェントの状態やその変化を表出する行動を意味する。PPP Persona では、図 5 に示される“移動”、“アイドリング”、“活動中”、“反射行動”の 4 つの自己行動が用意されている。

i. 行動の記述

PPP Persona の行動は、古典的プランニングと同様のオペレータにより記述されている。オペレータには、直接的に画像系列に関連づけられている基本行為と、複数の基本行為から構成される複合行為からなる。例として、以下に基本行為 `bottomupjumping` を示す。このオペレータは、両手が標準位置にあり、PPP Persona がアイコン化されておらず、ユーザの方を向いて指示棒なしで立っている場合に適用可能になり、「ジェスチャ: 42」に割り当てられた画像系列が表示され、その結果として「効果: $((posy = 1))$ 」で状態が更新されることを意味する。

基本行為: `bottomupjumping`
 条件: $((leftarm\ standard)\ (rightarm\ standard)\ (icon\ noicon)\ (bodydir\ front)\ (bodypos\ stand)\ (stick\ off))$
 効果: $((posy = 1))$
 ジェスチャ: 42

ii. 行動のコンパイル

実時間でエージェントを行動させるためには、ランタイムで行動を生成していたのでは間に合わない。よって、事前に、プレゼンテーションスクリプトからすべての考えられる初期状態にいたるまで、後ろ向きにプランニン

グを行ない、状態遷移をすべて展開して、有限状態オートマトンを生成するコンパイラが開発された。そして、プレゼンテーションの実行時には、その有限状態オートマトンをたどることにより、高速にエージェントの行動を実行できる。

この有限状態オートマトンから、行動モニタ (behavior monitor) が構成される。行動モニタは、複雑なプレゼンテーションタスクを PPP Persona の画像系列に分解し、基本行為の適合条件の判定、アニメーションの実行、そして状態の更新を行なう。さらに、必要に応じて自己行動を随時実行することにより、エージェントの信頼感を増大させることが可能である。

4. ソフトウェアエージェントにおけるプランニングの課題

ここでは、ソフトウェアエージェントにおけるプランニングの課題を議論する。

4.1 状態の不確実性

ソフトウェアエージェントはユーザとインタラクションをもつものが多く、そのとき環境の状態は、ユーザの感情、認知的負荷などからなるユーザの内部状態を含んだものになる。また、当然ながら、環境の情報から状態を同定することは、目標の同定や再プランニングなどにおいて非常に重要であるため、プラン認識の分野で研究されている。しかし、一般に人間の内部状態は不安定で不確実なため同定が難しく、ソフトウェアエージェントによるユーザの内部状態の同定も簡単にはいかない。今後、コンピュータビジョンによる表情認識などをさらに発展させたユーザの内部状態の同定手法の開発が、プランニングそのものの研究と共に重要な課題と考えられる。

4.2 オペレータの自動生成

状態の同定の困難さに加えて、プランニングの実用化を妨げているのが、オペレータ生成の問題である。通常、プランニングの枠組みでは、想定されるタスクに対して必要十分なオペレータが設計者により与えられていると仮定している。しかし、実際の応用では、オペレータを事前に誤りなく適切な粒度で記述することは、設計者にとって大きな負担となり、そのような仮定は成り立ちにくい。この問題は、Web のような動的な環境で行動するソフトウェアエージェントではより深刻な問題なる可能性がある。なぜなら、使えるオペレータが時間的に、あるいは Web サイトなどの空間的に変化するかも知れず、そのことがすべてのオペレータを事前に用意することを困難にするからである。

この問題を解決する方法の一つは、オペレータをエージェントが自動生成することである。オペレータの自動生成については、記号のオペレータ生成 [Wang 95]、口

ボットの実世界でのセンサ信号からオペレータを生成する研究 [Schmill 98], Web ページからオペレータを生成する研究 [山田 99] などいくつかの研究があるが, まだまだ不十分であり今後の発展が期待される.

4.3 実行可能なプランの任意時間プランニング

擬人化エージェントがユーザに信頼されるための条件として, 人間と同様の応答が挙げられる. 人間の場合, 難しいタスクであってもある時間内で適当なプランの実行を開始する. このような応答をエージェントで実現するには, 完全で最適なプランニングを目指すよりも, 即応プランニングや任意の時間で実行可能なプランを出力できる任意時間プランニング (anytime planning) の方法論が重要である.

即応プランニング [山田 93] は, 実現できる行動が反射的行動の域を越えていないため, 服従アーキテクチャ (subsumption architecture) のようなプランニングによる熟考 (deliberation) と反射行動の両方を使い分けることが必要になる. 実際, 前述の PPP Persona においては, プレゼンテーションスクリプトと自己行動の併用という形で, この使い分けが実現されていると考えられる. また, 任意時間プランニング [Boddy 89] では, 実際に環境において実行可能 (executable) なプランを生成することが肝要である. 例えば, 後向きプランニングが途中で中断されるとその不完全なプランは初期状態に適用できないため, そのプランの実行は難しい. 一方, 単なる前向きプランニングでも実行可能なプランを生成できるが, 実行可能な不完全プランを生成できる任意時間プランニングに適したオペレータ記述まで考慮に入れた研究が今後望まれる.

5. ま と め

本稿では, ソフトウェアエージェントにおけるプランニングを概観するために, プランニングが応用されているインターネットエージェントと擬人化エージェントの研究事例を紹介した. そして, 最後に, ソフトウェアエージェントにおけるプランニングの研究課題について考察した.

プランニングするソフトウェアエージェントは, プランニングを行動決定のメカニズムとして用いた, 環境の観測, 行動決定, 実行からなる総合的なシステムである. エージェントが適切な行動をとれるためには, プランニングの性能だけの性能向上を目指すよりも, エージェントというシステム全体が上手く行動できるように, オペレータ生成や状態の同定までも視野にいれた研究が今後進むべき方向と考える.

◇ 参 考 文 献 ◇

- [Allen 83] Allen, J. F.: Maintaining Knowledge about Temporal Intervals, *Communications of the ACM*, Vol. 26, No. 11, pp. 832-843 (1983).
- [André 98a] André, E., Rist, T., and Müller, J.: Integrating Reactive and Scripted Behaviors in a Life-Like Presentation Agent, in *Proceedings of the Second International Conference on Autonomous Agent*, pp. 261-268 (1998).
- [André 98b] André, E., Rist, T., and Müller, J.: WebPersona: A Life-Like Presentation Agent for the World-Wide Web, *Knowledge-Based Systems*, Vol. 11, No. 1, pp. 25-36 (1998).
- [馬場口 99] 馬場口, 山田: 人工知能の基礎, 昭晃堂 (1999).
- [Boddy 89] Boddy, M. and Dean, T.: Solving Time-Dependent Planning Problems, in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 979-984 (1989).
- [Etzioni 94] Etzioni, O. and Weld, D.: A SoftBot-Based Interface to the Internet, *Communication of the ACM*, Vol. 37, No. 7, pp. 72-76 (1994).
- [大須賀 00] 大須賀昭彦: プランニングモバイルエージェント, 長尾確 (編), エージェントテクノロジー最前線, 第 5 章, pp. 84-115, 共立出版 (2000).
- [Schmill 98] Schmill, M. D., Rosenstein, M. T., Cohen, P. R., and Utgoff, P.: Learning What Is Relevant to the Effects of Actions for a Mobile Robot, in *Proceedings of the Second International Conference on Autonomous Agent*, pp. 247-253 (1998).
- [Wang 95] Wang, X.: Learning by Observation and Practice: An Incremental Approach for Planning Operator Acquisition, in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 549-557 (1995).
- [Weld 94] Weld, D.: An Introduction to Least Commitment Planning, *AI magazine*, Vol. 15, No. 4, pp. 27-61 (1994).
- [山田 93] 山田誠二: リアクティブプランニング, 人工知能学会誌, Vol. 8, No. 6, pp. 729-735 (1993).
- [山田 99] 山田, 大澤: WWW における概念理解のためのナビゲーションプランニング, 人工知能学会誌, Vol. 14, No. 6 (1999).

[担当委員: × ×]

19YY 年 MM 月 DD 日 受理

—— 著 者 紹 介 ——

山田 誠二 (正会員)

1984 年大阪大学基礎工学部卒業. 1989 年同大学院博士課程修了. 博士 (工学). 同年大阪大学基礎工学部助手. 1991 年同大学産業科学研究科講師. 1996 年東京工業大学大学院総合理工学研究科助教授, 現在に在る. 工学博士. 人工知能, 特に, Web での情報検索, ロボット学習に興味をもつ. 情報処理学会, 日本ロボット学会, 電子情報通信学会, AAI, IEEE, ACM 各会員.