

解説

リアクティブプランニング

Reactive Planning

山田 誠二*
Seiji Yamada

* 大阪大学産業科学研究所
ISIR, Osaka University.

1993年7月22日 受理

Keywords: planning, dynamic world, robot.

1. はじめに

身体を持ち物理世界で行動する移動ロボットは、実世界とAIアルゴリズムを結ぶ総合的チャンネルである。そこに我々は、さまざまなセンサやアクチュエータを搭載して、実世界におけるAIの有効性を試すことができる。このように実世界に耐え得ることが望まれるAIアルゴリズムの典型例が、プランニングである。

プランニング(planning)とは、与えられた目的を達成するために、アクチュエータの(物理的)動作を計画することである。よって、プランニングでは、推論結果であるプランが実世界で実行されることが、十分考慮されなければならない。しかし、これまでの研究は、そのような考慮なしに、実世界の内部表現をどのように変換していけば目標に達成できるかという探索問題に終始してきた。そこには、エージェントを取り巻く環境は、変化しない静的世界であるという、現実的とは言い難い仮定がある。そして、それでは動的に変化する環境にうまく対応することは不可能である。このような背景から、単なる探索問題ではなく、実世界における目標の実現を目指して、近年「リアクティブプランニング(reactive planning)」と呼ばれる研究が始まった。本解説では、この分野について、その意義そして具体的な方法論を説明していく。なお、本誌における解説として、すでに[安部 90, 石田 90]があるので、本稿ではこれらとの重複を極力避けた。

2. リアクティブプランニング

2.1 古典的プランニング

STRIPS[Fikes 71]に代表される古典的プランニングでは、環境が計算機上の世界モデル(例えば、述語論理)で記述される。そして、世界モデルを変換する規則であるオペレータの集合と、世界モデルで記述された初期状態と目標状態が、問題として与えられる。古典的プランニングのタスクは、初期状態を目標状態に変換できるようなオペレータ系列を求めることである(図1)。この系列が、プランと呼ばれる。一般に、プランを探索することは、非常に多くの計算量を必要とし、積木の世界においても計算量が指数関数的に爆発する。

このようにして得られたプランは、アクチュエータ

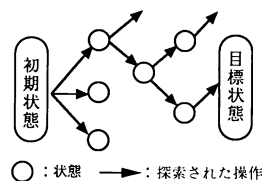


図1 古典的プランニング

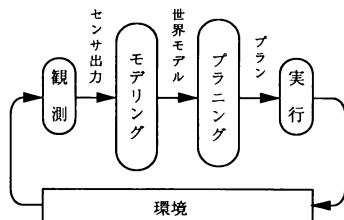


図2 古典的プランナを用いたSMPAモデル

により環境において実行され、実世界において目標状態が実現されるわけである。全体の処理は、センシング → モデリング → プランニング → 実行という手続きであり、SMPAモデル (sense-model-plan-action framework) [Brooks 91] と呼ばれる (図2)。この枠組みの現実的問題点として、環境のモデリングに多大な計算時間を要し、ロボットの動きが非常に緩慢であることが指摘されている [Brooks 91]。

2・2 実世界=動的世界

実世界でエージェントを取り巻く環境は、時々刻々変化する動的世界 (dynamic world) である。その変化は、他のエージェント、物理現象、操作実行の失敗などのさまざまな原因によるもので、プランニング時にそれらをすべて予測することは、現実には不可能である。たとえ、最初の観測による世界モデルをもとに、従来のプランニング手法で目標状態までの完全なプランを作っても、部分的にしか、あるいは全く実行できない事態が頻繁に起こり、環境での目標達成は困難となる。よって、実世界では、このような動的世界に対応したプランニングを行うことが必須となる。こうした背景から、動的世界への対応を目指すリアクティブプランニングの研究が始まった。

2・3 リアクティブプランニング

リアクティブプランニングとは、動的世界に対応できるプランニングである。この観点から、リアクティブプランニングが満たすべき条件を以下にあげる。

(1) **資源の有限性**：環境の変化に追従するには、エージェントの行為の選択と実行が、環境の変化に対し十分に迅速でなければならない。

(2) **不確定性の扱い**：プランや行為の実行の効果などの予測は、必ず不確定性 (uncertainty) を含む。よって、予測が外れた場合にどのように対応するかなどの処理が必要となる。

これらの問題に対処するリアクティブプランニングのシステムが、いくつか提案されている。その具体的な方法論を次章以降で解説していく。本稿では、リアクティブプランニングを、従来のプランニングを全く否定する方法と、それとの統合を目指すものとの二つに分類して説明する。

3. リアクティブプランニングの方法論(1)

—推論の否定—

エージェントが実世界で知的に行動するために、従

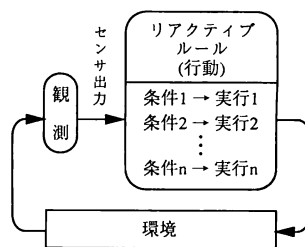


図3 リアクティブプランニングシステム

来の探索を伴うようなプランニングは必要でなく、その場の環境に応じて断片的な行動 (behavior) を行えばよいという考えがある [Agre 87, Agre 90]。また、環境の複雑さがエージェントの行動を複雑にするという主張 [Simon 69] もある。このような背景から生まれたのが、多段の推論 (以降、単に「推論」) を否定するリアクティブプランニングである。

直接実行可能な断片的な行動からなる結論部と、センサ出力から直接判定可能な条件部で記述されたルールは、リアクティブルール、あるいは行動ルール (behavior rule) と呼ばれる。リアクティブルールを用いたリアクティブシステムの構成を図3に示す。観測されたセンサ情報が、複数のリアクティブルールに直接的に与えられ、それぞれのルールが並列にその条件の適用を判定する。そして、条件が満たされたルールのうち、適当なものが選択され実行される。一つのルールが実行されると、また観測を行うというループが、環境の変化に対して十分に短い周期で繰り返される。これが、推論を否定したリアクティブプランニングシステムである。実際には、リアクティブルールの集合の構造化や、ルール間のインタラクションの制御がなされる。センサ情報の扱い方から見ると、古典的プランニングシステムは、モデル構築のために種々のセンサ情報が統合されるセンサフュージョン (sensor fusion) であるのに対し、リアクティブプランニングシステムは、センサ情報をそのまま各リアクティブルールに送るセンサフィッション (sensor fission) である。ここでは、この推論を否定するリアクティブプランニングの例として、包摂アーキテクチャ、状況オートマトンとその記述言語 REX を紹介する。

3・1 包摂アーキテクチャ

従来の移動ロボットのアーキテクチャは、図2のように全体が機能モジュールにより分割されていた。これにかわり、Brooks は非同期にタスクを遂行する行動 (behavior) により分割されたアーキテクチャを提案した (図4)。このアーキテクチャでは、上位レベルが下位レベルの行動を包摂 (subsume) し、制御するた

め、包摂アーキテクチャ (subsumption architecture) と呼ばれる [Brooks 86]. 各レベルのタスクは、センサ出力を直接受け取り並列に処理が進むため、従来の方式よりも、多重目標、多重センサ、頑健性、拡張性の点で優れている。

各レベルは、有限オートマトンを接続して構成される。図5に、その有限オートマトンの例を示す。オートマトンは、入出力、入力バッファ、状態変数、遷移規則を持つ。リセット入力があると、状態はNILになる。また入力にはセンサか他のオートマトンの出力に接続され、出力はアクチュエータか他のオートマトンの入力に接続される。さらに、次に示すようなゲート接続が可能である。

- (1) 抑制 (inhibiter) : 出力に接続され、トリガにより、その出力は一定時間削除される。
- (2) 置換 (suppressor) : 入力に接続され、トリガ

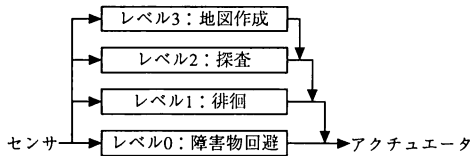


図4 包摂アーキテクチャ [Brooks 86]

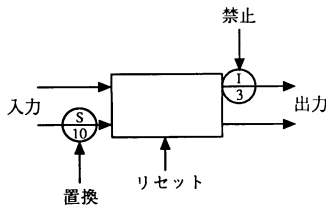


図5 有限オートマトン [Brooks 86]

により、その入力は一時間上位レベルの出力で置き換えられる。

図中の丸の中のIとSが抑制と置換を表し、その下の数字が上記の一定時間である。このような有限オートマトンと、それらの接続を記述する言語が、LISP上で提供されている。構成手順としては、まず、低レベルを有限オートマトンの接続により構成する。そして、次に上位レベルを構成し、その二つのレベル間をゲート接続で結ぶ。この手順を繰り返すことにより、包摂アーキテクチャがインクリメンタルに構成できる。

例えば、レベル0: 障害物の回避、レベル1: 徘徊、レベル2: 探査行動の三つのレベルで構成される移動ロボットの包摂アーキテクチャは、図6のようになる。また、実際に包摂アーキテクチャを用いた移動ロボットが製作されており、作為的な制約のない研究所内でゴミを回収するなどのタスクを実行できた。

移動ロボットの比較的単純なタスクでは、包摂アーキテクチャのどのレベルでも、世界モデルやプランニングを必要としない。このことから、Brooksの「表象なき知能 (intelligence without representation)」というスローガンが生まれる。とにかく、従来の移動ロボットが、世界モデルの構成に多大な時間をかけて牛歩のごとく移動していたことを考えると、低レベルでの処理により、障害物回避などを即座に実行できる包摂アーキテクチャは、まさにリアクティブシステムである。また包摂アーキテクチャは、図3のシステムにおけるリアクティブルール of 構造化と制御を提案したものであると解釈できる。

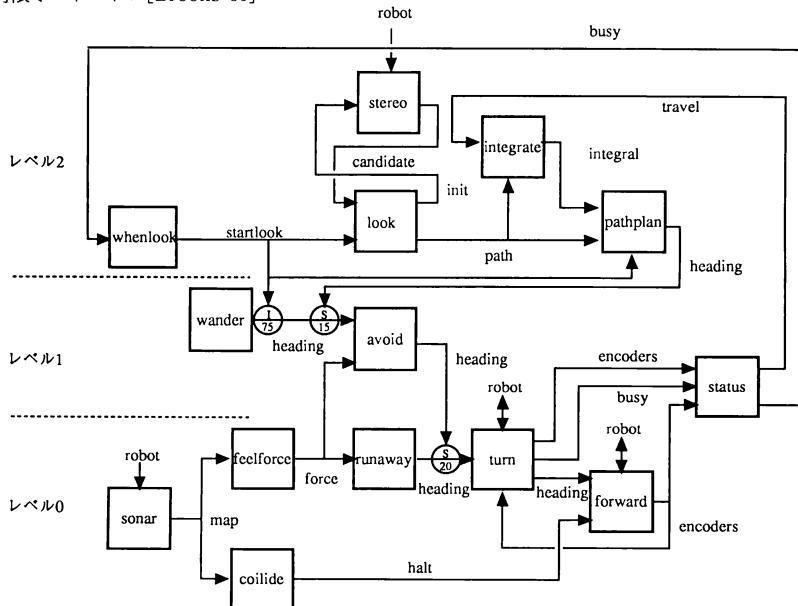


図6 移動ロボットの包摂アーキテクチャ [Brooks 86]

3・2 状況オートマトンと REX

状況オートマトン (situated-automata) [Rosenschein 85] は、エージェントとそれを取り巻く物理的環境に対応したオートマトンのペアによる世界のモデルで、様相論理をその基盤とする。従来の認識論理 (epistemic logic) では、エージェント x が命題 φ を知っているとは、 φ が知識ベースにおいてコード化されているか、あるいは演繹可能であることを意味したが、それでは、知識はある構造化をされていることが前提となり、意味がその構造の解釈に依存するため客観性を失う。状況オートマトンでは、「 x の内部状態が v であるすべての状況において、 φ が真であるとき、 x は内部状態 v で φ を知っている」と再定義され、そのうえで様相論理が組み上げられる。このような理論的枠組みを用いると、例えば、デジタル回路のコンポーネントをエージェントとして捉え、エージェント間の情報の流れを推論することにより、デジタル回路を解析できる。

この状況オートマトンを実装するためのプログラミング言語が、REX である [Rosenschein 86]。REX は、Lisp-like な言語で、記憶エレメント (storage) とそれらの間の遷移規則などの制約を記述する。REX コンパイラは、入力制約を満たすデジタル回路の記述を生成して、その回路により一定時間内に計算が行われる。例えば、1, 2 または 3 を入力されたときのみ、1 を出力するような REX プログラムからは、図 7 のような回路が生成される。実際に、REX は SRI の移動ロボットで使用された。

GAPPS [Kaelbling 88] は、REX をより宣言的にした言語で、目標と目標リダクションルール (goal reduction rule) により、並列性を持つ行動を記述できる。GAPPS コンパイラは、与えられたゴールにたどり着くためのリアクティブルールの集合を出力する。これら一連の研究は、[Kaelbling 91] にまとめられている。

なお、この種のリアクティブプランニングシステムとして、ほかにも PRS [Gorgeff 89]、Pengi [Agre 87] があがるが、これらについては、[安部 90, 石田 90] を参

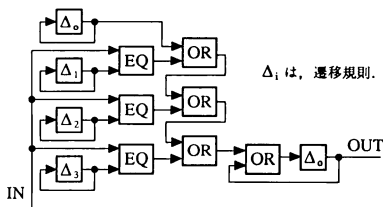


図 7 REX により生成された回路 [Rosenschein 86]

照されたい。

4. リアクティブプランニングの方法論 (2)
—古典的プランニングとの統合—

本章では、推論を否定するのではなく、従来のプランニングとリアクティブプランニングとの統合を図ろうとする研究に注目する。

4・1 プランニングによるリアクティブルールの生成
古典的プランニングあるいはそれに類似した手法を用いて、リアクティブルールを事前に生成する研究を紹介する。

[1] ユニバーサルプラン

Schoppers は、「プランニングとは、起こり得る状況に対する行動の目標指向的選択である」と主張する [Schoppers 87]。与えられた目標状態から、オペレータを用いて後ろ向きに探索を行うことにより、操作の実行によって起こり得るすべての状態をあらかじめ生成し、その状態と目標状態を条件部としたリアクティブルールを自動生成する。これが、ユニバーサルプラン (universal plan) であり、決定木で表現される。図 8 に、目標状態が $on(a, b) \wedge top$ の場合のユニバーサルプランを示す。図中の下線部は行為を表す。このようなユニバーサルプランをいったん生成しておけば、後は環境の状況が、決定木上のどの状態かを探索し、そこに書かれているオペレータの実行を繰り返すことにより、動的世界に対応でき、再プランニングも不要である。

ユニバーサルプランは、起こり得る状況のグループ化したコンパクトな表現になっている。しかし、潜在

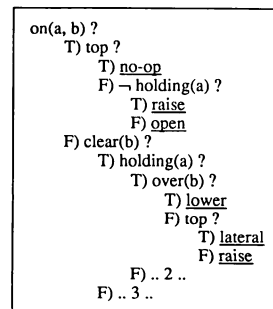


図 8 ユニバーサルプラン [Schoppers 87]

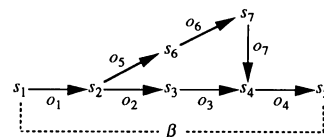


図 9 時制射影 [Drummond 90]

的に、センサ出力のチャンネル数の指数オーダで状態数が増加するため、ユニバーサルプランのサイズの爆発が問題となり、*AI Magazine* 誌上で議論がされた [Ginsberg 89]。

〔2〕 状況制御ルールの生成

Drummond は、時制射影 (temporal projection) により、リアクティブルール的一种である状況制御ルール (situated control rule) [Drummond 89] を自動生成する方法を提案した [Drummond 90]。まず、ある時間区間において満たされるべき行動制約 (behavioral constraints) で表現された目標が与えられる。また、実行時間および実行が成功する確率が記述されたオペレータと、行動制約の反順序関係である行動制約戦略も与えられる。そして、初期状態から前向きにオペレータを適用して、状態遷移図を展開して時制射影を行い、目標を満たすプランが探索され、まず一つのプランが得られる。

図 9 に、時制射影の例を示す。 s と o はそれぞれ状態と適用されたオペレータである。この例では、 β という行動制約が与えられ、状態 s_1 から、まずプラン $[o_1, o_2, o_3, o_4]$ が得られる。後は時間の許す限り、別のプラン ($[o_1, o_5, o_6, o_7, o_4]$) が探索される。そして、プラン上の s と o それぞれについて、IF s & β THEN o という形式の状況制御ルールが生成される。

一つのプランから得られたルールが実行できない状況でも、代わりに別のプランのルールを使える場合があるので、時間をかけるほど、多くのプランが生成されて頑健性 (robustness) が増す。つまり、処理時間に比例して質の高い出力が得られることになり、このことは任意の時間でアルゴリズムを停止しても、その時間において最適な解が得られることを意味する。このような特性を持つアルゴリズムは、任意時間アルゴリズム (anytime algorithm) [Dean 88] と呼ばれる。

〔3〕 リアクティブルールの学習

Mitchell らは、説明に基づく学習: EBL [Mitchell 86] によるリアクティブルールの学習を提案している [Mitchell 90]。Theo-Agent と呼ばれる実際の移動ロボット (Hero 2000) 上で実装され、プランが生成されるごとに、そのプランをマクロ化してリアクティブルールが学習される。その結果、観測から行動選択、実行までの反応時間は、学習前後で、数十秒から 1 秒以下に高速化される。ただし、実験で用いられた三つの目標は、物体に近づくなどの非常に単純なもので、学習されたリアクティブルールも少数であり、現実的な領域でどれほどの能力があるのかは、不明である。また、類似した研究として、[Gervasio 89] がある。

4・2 即応性と熟考性の統合

リアクティブルールからなるシステムでは、環境の変化に対する即応性 (reactivity) は高いが、その行動の最適性は乏しい。逆に、プランを作る (熟考する) と、最適性は高くなるが、即応性が低下する。このような現象は、即応性と熟考性 (deliberation) のトレードオフと呼ばれ、リアクティブプランニングの目標の一つが、これらの統合である [Shapiro 92]。以下に、このテーマに関する研究を紹介する。

〔1〕 IRMA

Pollack らは、動的世界のシミュレーションであるタイルワールド (TileWorld) において、即応性と熟考性のトレードオフの実験を行った [Pollack 90]。タイルワールドとは、2次元格子上に、エージェントと複数個のタイル、障害物、そして穴を配置した計算機上の実験環境である。エージェントは、タイルを穴の位置まで運んでいき、それを埋めると、穴に割り当てられている得点が得られる。タイルと穴が出現・消失することにより、動的世界となっている。

この実験で用いられたエージェントは、Bratman らにより提案された IRMA (Intelligent Resource-Bounded Machine Architecture) [Bratman 88] を具体化したものである。その構成を図 10 に示す。全体のプロセスの目的は、意図構造 (intention structure) を環境に応じて維持することである。意図構造とは、現在のエージェントの意図を表すもので、具体化されたプランの集合であり、そのプランに沿って実行が行われる。まず、認知プロセスは、環境の変化を観測し、新しく現れた穴などを目標 (オプションと呼ばれる) として、フィルタリング機構に渡す。また、従来のプランナである目標-手段推論で生成されたプランもオプションとして、フィルタリング機構に渡される。そして、現在の行動との整合性や目標の価値のしきい値評価に基づき、フィルタリング機構により選択されたオプションが、熟考プロセスに渡される。さらに、そこで穴の得点や単位労力当りの得点率により、オプションの詳細な評価 (熟考) が行われる。熟考プロセスによ

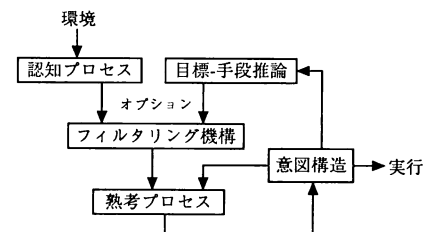


図 10 TileWorld エージェント [Pollack 90]

り選択されたオプションから、意図構造が作られる。ここでは、熟考とは、目標-解析推論ではなく、オプションの選択と捉えられる。

このシステムで簡単な実験が行われた。まず、環境変化の速度を変えて、エージェントと人間の得点を調べた。その結果、環境変化が速くなると、人間の応答が追従できなくなった。次に、フィルタリング機構のしきい値と環境変化の速度の関係を調べた結果、環境変化が遅いときは、フィルタリングしないほうがよいことがわかった。

〔2〕 プランニングと実行のインタリーブ

部分的プランニングと実行を交互に繰り返すインタリーブプランニング[McDermott 78]も、即応性と熟考性の統合を実現できる。ただし、いつプランニングを停止して、実行に切り換えるのかという問題がある。

筆者は、プラン中のすべてのオペレータが実行可能な確率をプランの成功確率と定義づけ、それを用いてプランニングと実行の切替えタイミングを決定する方法を提案した[山田 91]。まず、STRIPS-like なプランからベイジアンネットワークが生成され、そのうえで環境の変化の程度を表す入力確率を用いて、プランの成功確率が計算される。環境の観測は、プランニングと並行に行われ、プランナは前向きビームサーチでプランを展開していき、その成功確率が実行のしきい値より下回ると、実行が開始される。このインタリーブプランニングは、実行のしきい値を変えることによって、熟考性と即応性の強さの比を調節できることが特徴である。

また、簡単な実験が、タイルワールドで行われ、タイルの配置や得点の分布が偏っている場合にインタリーブの効果ができること、つまり、熟考性と即応性の統合が有効であることが報告されている[磯田 93]。

上記二つの研究で実験に用いられたタイルワールドあるいはそれを単純化したもの[Kinny 91]が、動的世界の一つの標準的実験環境になりつつある。今後このような標準問題の設定が、各研究の比較評価のためにも重要であろう[大沢 93]。

〔3〕 行動ネットワーク

Maes によって提案された行動ネットワーク (behavior network)[Maes 90, Maes 91]は、行動をノードとし、行動間の因果関係をアークとする有向グラフである(図 11)。行動は、STRIPS-like なオペレータで記述され、ある行動の追加リスト中のリラテルと同じものが、他の行動の条件リスト中にあるとき、その間に因果関係のアークが張られる。

各ノードは、並列に環境と目標をモニタして、それ

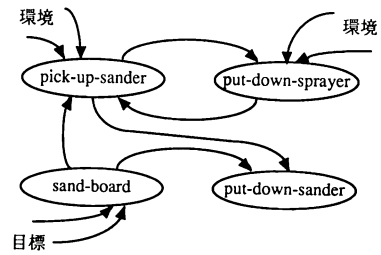


図 11 行動ネットワーク[Maes 90]

ぞれ条件リストと追加リストとの照合がとられ、前向き・後ろ向きのエネルギーを受け取る。そして、アークを通して行動間で前向き・後ろ向きのエネルギー伝搬を繰り返す。ネットワーク全体が平衡状態に達したとき、環境において条件リストがすべて真のノードの行動が、そのエネルギーの強さに比例した確率で実行される。このようなエネルギー伝搬により、再プランニングや頑健性を実現できることが実験的に示されている。

行動ネットワークにおいて、前向き・後ろ向きのエネルギー伝搬の比をパラメータで調整できる。前向きを強めた場合は、合目的性はあまりないが、環境に適用可能な行動が即座に実行され、後ろ向きを強めた場合は、合目的性のある行動が遅れて実行される。この意味で、行動ネットワークも、熟考性と即応性が統合されており、さらにその調節も可能であるといえる。ただし、現実的にはパラメータ調整が難しい。

5. ま と め

本稿では、リアクティブプランニングをその方法論を中心に解説した。包摂アーキテクチャなどの研究は、ロボットをもっと速く動かそうという物理的要求に端を発しており、ロボットという物理的なチャンネルを通して得られた貴重な方向性であると考えられる。古典的プランニングを研究してきた古典的 AI 研究者は、この刺激に対して反応すべきであり、その一つの回答が従来のプランニングとの統合にあるといえる。これを機会に、ロボット研究者と AI 研究者間での建設的交流が発展することを切に希望する。

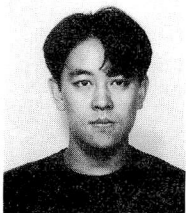
謝 辞

本解説を執筆する機会を与えてくださった三菱電機(株)の滝 寛和氏、草稿に関してコメントをいただいた大阪大学基礎工学部の山口智浩助手、石黒 浩助手、そして文献調査を協力していただいた磯田佳徳氏、矢倉洋之氏に感謝いたします。

◇ 参 考 文 献 ◇

- [安部 90] 安部憲広：プランニング，人工知能学会誌，Vol.5, No. 6, pp.737-747 (1990).
- [Agre 87] Agre, P. E. and Chapman, D.: Pengi: A Implementation of a Theory of Activity, *AAAI-87*, pp.268-272 (1987).
- [Agre 90] Agre, P. E. and Chapman, D.: What are plans for?, *Designing Autonomous Agents*, Bradford-MIT (1990).
- [Bratman 88] Bratman, M. B., Israel, D. J. and Pollack, M. E.: Plans and resource-bounded practical reasoning, *Comput. Intell.*, No. 4, Vol. 4 (1988).
- [Brooks 86] Brooks, R. A.: A Robust Layered Control System for a Mobile Robot, *IEEE Robotics and Automation*, Vol. 2, No. 1, pp.14-23 (1986).
- [Brooks 91] Brooks, R. A.: Intelligence Without Reason, *IJCAI-91*, pp. 569-595 (1991).
- [Dean 88] Dean, T. and Boddy, M.: An Analysis of Time-Dependent Planning, *AAAI-88*, pp. 49-54 (1988).
- [Drummond 89] Drummond, M.: Situated Control Rules, *Conf. on Principles of Knowledge Representation and Reasoning* (1989).
- [Drummond 90] Drummond, M. and Bresina, J.: Anytime Synthetic Projection: Maximizing the Probability of Goal Satisfaction, *AAAI-90*, pp. 138-144 (1990).
- [Fikes 71] Fikes, R. E. and Nilsson, N. J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artif. Intell.*, Vol. 2, pp. 189-208 (1971).
- [Gervasio 89] Gervasio, M. T. and DeJong, G. F.: Explanation-Based Learning of Reactive Operators, *Int. Workshop on Machine Learning*, pp. 252-254 (1989).
- [Ginsberg 89] Ginsberg, M. L.: Universal Planning: An (Almost) Universally Bad Idea, *AI Magazine*, Vol. 10, No. 4, pp. 40-44 (1989).
- [Gorgeff 89] Gorgeff, M. P. and Ingrand, F. F.: Decision-Making in an Embedded Reasoning System, *IJCAI-89*, pp. 972-978 (1989).
- [石田 90] 石田 亨：知識表現と動的世界—最近のプランニング研究から—，人工知能学会誌，Vol.5, No.2, pp.146-153(1990).
- [磯田 93] 磯田，山田，豊田：インタリーブ・プランニングとその実験的評価，情処学会人工知能研，93-AI-86-7 (1993).
- [Kaelbling 88] Kaelbling, L. P.: Goals as Parallel Program Specifications, *AAAI-88*, pp.60-65 (1988).
- [Kaelbling 91] Kaelbling, L. P. and Rosenschein, S.: Action and Planning in Embedded Agents, *Designing Autonomous Agents*, MIT Press (1990).
- [Kinny 91] Kinny, D. and Georgeff, M.: Commitment and effectiveness of situated agents, *IJCAI-91*, pp. 82-88 (1991).
- [Maes 90] Maes, P.: Situated Agents can Have Goals, *Designing Autonomous Agents*, MIT Press (1990).
- [Maes 91] Maes, P.: Learning Behavior Networks from Experience, *1st European Conf. on Artificial Life* (1991).
- [McDermott 78] McDermott, D.: Planning and Action, *Cognitive Sci.*, Vol. 2, pp. 71-110 (1978).
- [Mitchell 86] Mitchell, T. M., Keller, R. K. and Kedar-Cabelli, S.: Explanation-Based Generalization: A Unifying View, *Machine Learning*, Vol. 1, No. 1, pp. 47-80 (1986).
- [Mitchell 90] Mitchell, T. M.: Becoming Increasingly Reactive, *AAAI-90*, pp. 1051-1058 (1990).
- [大沢 93] 大沢，沼岡，石田：分散人工知能における標準的小問題，コンピュータソフトウェア，Vol.10, No.3, pp.3-19(1993).
- [Pollack 90] Pollack, M. E. and Ringuette, M.: Introducing the Tileworld: Experimentally Evaluating Agent Architectures, *AAAI-90*, pp. 183-189 (1990).
- [Rosenschein 85] Rosenschein, S.: Formal theories of knowledge in AI and Robotics, *New Generation Computing*, Vol. 3, No. 4, pp. 345-357 (1985).
- [Rosenschein 86] Rosenschein, S. and Kaelbling, L.: The Synthesis of Digital Machines with Provable Epistemic Properties, *Workshop on Theoretical Aspects of Knowledge*, pp. 83-98 (1986).
- [Schoppers 87] Schoppers, M. J.: Universal Plans for Reactive Robots in Unpredictable Environments, *IJCAI-87*, pp. 1039-1046 (1987).
- [Shapiro 92] Shapiro, S. C. (ed.): *Encyclopedia of Artificial Intelligence (2nd ed.)*, pp. 1171-1181, Wiley-Interscience Pub. (1992).
- [Simon 69] Simon, H. A.: *Sciences of the Artificial*, MIT Press (1969).
- [山田 91] 山田誠二：インタリーブによるリアクティブ・プランニング，情処学会人工知能研，91-AI-79-8 (1991).

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。同年，基礎工学部制御工学科助手。1991年より大阪大学産業科学研究所講師，現在に至る。工学博士。人工知能，特に，説明に基づく学習，リアクティブプランニング，定性的ロボットナビゲーションなどの研究に従事。1990年人工知能学会全国大会優秀論文賞。1992年日本認知科学会奨励賞。1993年情報処理学会研究賞，情報処理学会，日本認知科学会，日本ソフトウェア科学会，AAAI，IEEE各会員。