GDC における海外のゲーム関連技術についての調査

三宅　陽一郎
株式会社フロム・ソフトウェア　技術部

概論

GDC2010 は、昨年よりやや規模の縮小した形で開催された。2010 年という年は、次世代機(ここでは 2005 年以降の新プラットフォーム、PlayStation3, Xbox360, Wii 世代機を指す)に入って、２つ目の次世代機タイトルのリリースが始まる時期であり、１つ目のタイトルで構築した開発基盤の上に、２つ目のタイトル開発の技術を着実に積み上げているという印象を強く受けた。ここでは、特にゲーム AI 分野の講演に特化して、２つのタイトル「Killzone 2」(Guerrilla Games, 2009) [1]と「The Sims 3」(EA, Maxis,2009) [2]（邦題：シムピープル）のゲーム AI 技術の詳細を解説し、対照的な AI の実装を要求する両タイトルにおけるゲーム AI 技術の比較を通して、ゲーム AI 分野全体の広がりを模索して行く[3]。最後に、この先鋭的な仕事を通して、これからの１０年のゲーム AI 技術の展望を解説する。ここでは、AI と言った場合、キャラクター、及び、キャラクターの持つ知能そのものを指すものとする。本報告のベースとなる講演は以下の二講演である。

- Alex J. Champandard, "On the AI Strategy for KILLZONE 2's Multiplayer Bots" (AI Summit: Case Studies: AI in Recent Games)　[1] [4]

- Richard Evans. "Modeling Individual Personalities in The Sims 3" [2]

Killzone 2 の AI 技術

Killzone 2 概観

Killzone 2 は FPS（一人称視点シューティングゲーム）であり、オフラインモードとオンラインモードが存在する。オフラインモードについては、GDC2009 の講演に基づき昨年の報告書で解説した[5]。Killzone 2 のパス検索システム、射線判定システムについては、そちらをご覧頂きたい。また、Killzone のパス検索システム、射線判定システムも、2008 年度の報告書で解説している[6]。Killzone 2 のオンラインモードは、２チーム１６人に分かれた３２人の対戦であり、AI がチームを組んで行動することが出来る。今回の、GDC2010 の講演は、このオンラインモードにおける AI 技術の講演であり、以下、この講演の内容と

その背景を解説する。

人工知能には、記号操作を主として知能を構築しようとする流れと、ニューラルネットのように数値的なシステムで知能を構築しようとする流れがある[7]。前者は明示的に知識を表現する方法であり、後者は知識は非明示的に数値の配列の中に含まれている。

Killzone 2 は前者のようにシンボリックな思考システム（HTN プラナー）を持つと同時に、後者のように戦局全体を数値的ダイナミクスによって認識する知能を持った AI である。Killzone 2 のマップの特徴は、前作同様に、は障害物が多く段差が激しく、立体的構造物に満ちている。Killzone 2 の AI 技術が目指していたものは、こういった複雑なマップでも、長期間（数分）に渡って AI が正しい位置取りをしながら、戦術的に行動し、かつ AI チームとして大局的に戦略的に行動できる知能の構築である。

エージェントとチームの関係

各エージェント（AI）が賢明である、という意味と、チームとして賢明であることは意味が異なる[8]。各エージェントは知覚できる周囲の局所的な情報に基づき行動するのに対して、チームは、大局的な戦局の趨勢に基づいて行動しなければならない。また、チームとそこに属するエージェントは無関係ではない。チームには組織としての構造と、情報の流通が整備されなければならない。Killzone 2 の AI では、最も上位に属する「Strategy AI」という司令官 AI の下に「Squad AI」（部隊 AI、チーム AI）が複数配置され、「Squad AI」には複数の「Individual AI」が属する。命令は上から下へ伝達されるが、各 AI が判断した局所的戦局が、上位へと伝達される。また、「Individual AI」（個体 AI、エージェント）は、命令がない、或いは無効化された場合には、上位へ向けて命令の再発行を求める。

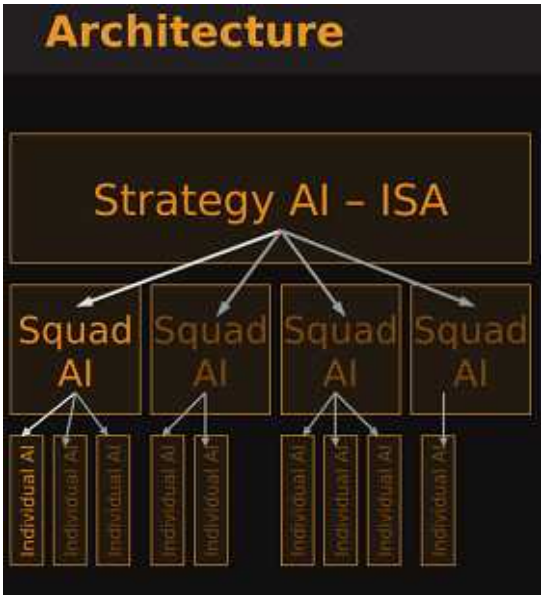


図 Killzone 2 における AI 組織図[1]　階層構造になっている。

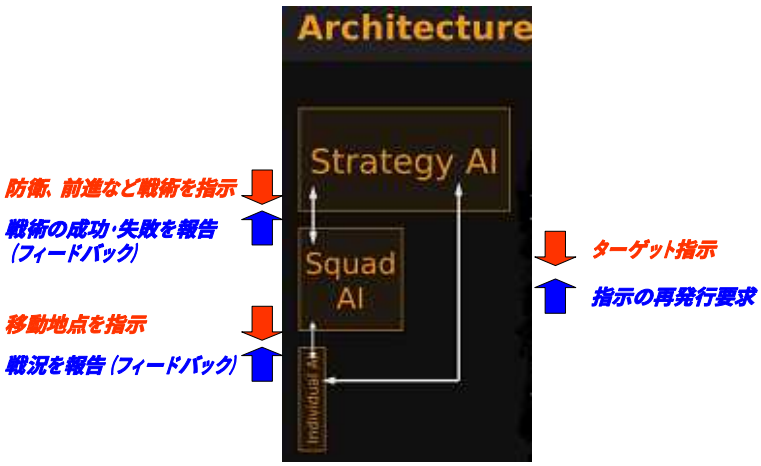


図　Killzone 2 における AI 組織の階層構造間の情報伝達フロー[1]　各階層の間には相互の情報の流れがある。上位から下位には「命令」が、下位から上位には「報告」「要求」が伝達される。

各 AI のアーキテクチャー

各エージェントのアーキテクチャーは、HTN プラナー(Hierarchical Task Network Planner)をコアとするエージェント・アーキテクチャを取っている。プラナー[7]とは行動プランを生成するシステムのことで、ここでは一連のタスクを階層的に生成するアルゴリズム(HTN Planner)である。周囲の脅威(threat)の情報、上からの命令(order)、AI 同士のメッセージ(message)などから、現在の周囲の環境世界（World State）を再構成し、そこからタスクプランを生成し、順番にそれを実行して行く(task execution)。

各AIのアーキテクチャ

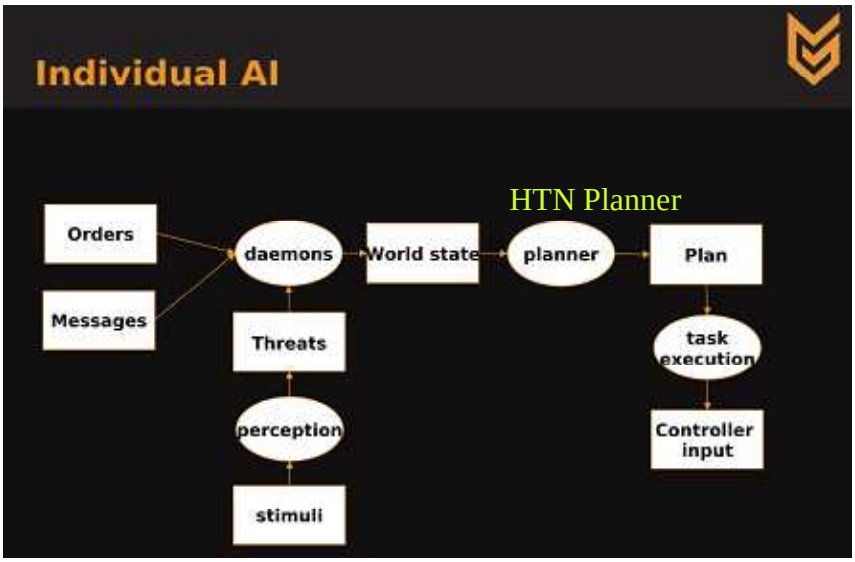


図　Killzone 2 **の各エージェントの思考構造**[1]

階層型プランニングとは、階層的な分解によって最終的なタスクリストを導くアルゴリズムである[9] [10] [7]**。階層型タスクネットワーク**(HTN)**は、タスクに分解するところまでは同じであるが、分解によって（論理的に）導出したタスクを、実行可能な順序に従って向き付けられたネットワーク状に再構成する過程を含むものである。また、この過程で、情報が補完されることでタスクがカスタマイズされる過程を含む。最終アウトプットは、タスクを要素とする向き付けられたタスクネットワークである。**

「Killzone 2**」の場合、タスクネットワーク形成の部分は単に分解したタスクを単にスタック状に並べて実行するので、それほど凝った** HTN **にはなっていない（むしろ、単に階層型プランニングと言った方がよい。おそらく前提条件が設定されているのと、実行順序が規定されるので** HTN **と言っているのであろう）。公開された擬似コードの部分には、「前提条件」**(precondition)**とタスクがセットになったコードが階層的に記述されており、ゲーム状況によって満たされた「前提条件」に対応するタスクリストが実行タスクとしてスタックされて行くことになる。**

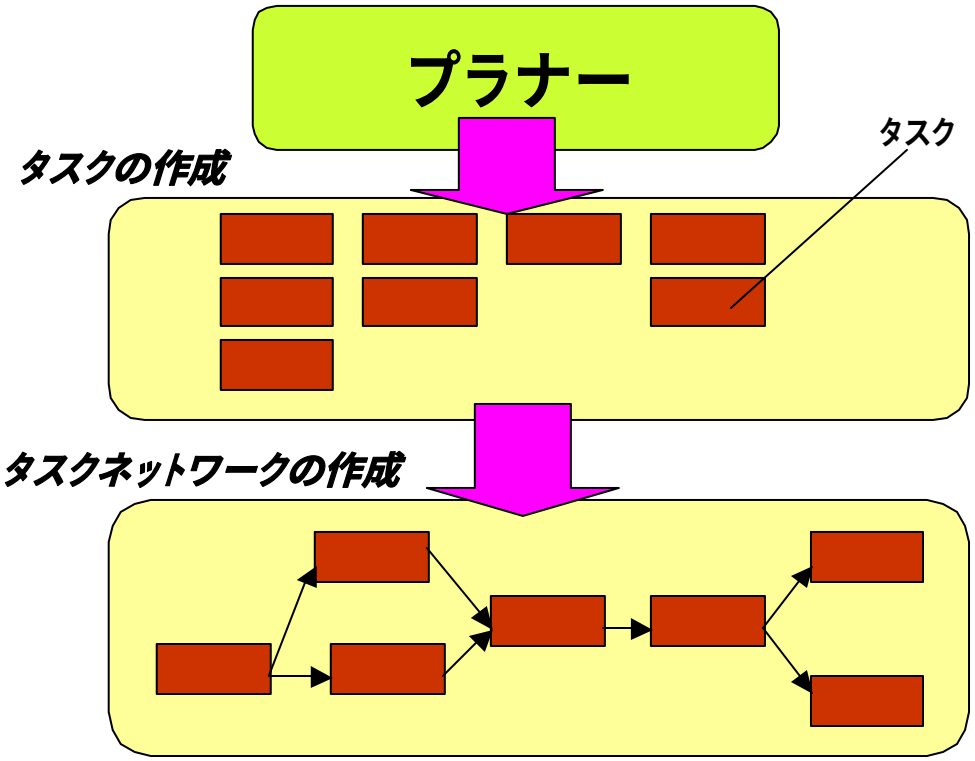


図 HTN プランニングのイメージ　導出したタスクの集合を規則に従ってタスクネットワークとして再順列する

HTN (Hierarchical Task Network)

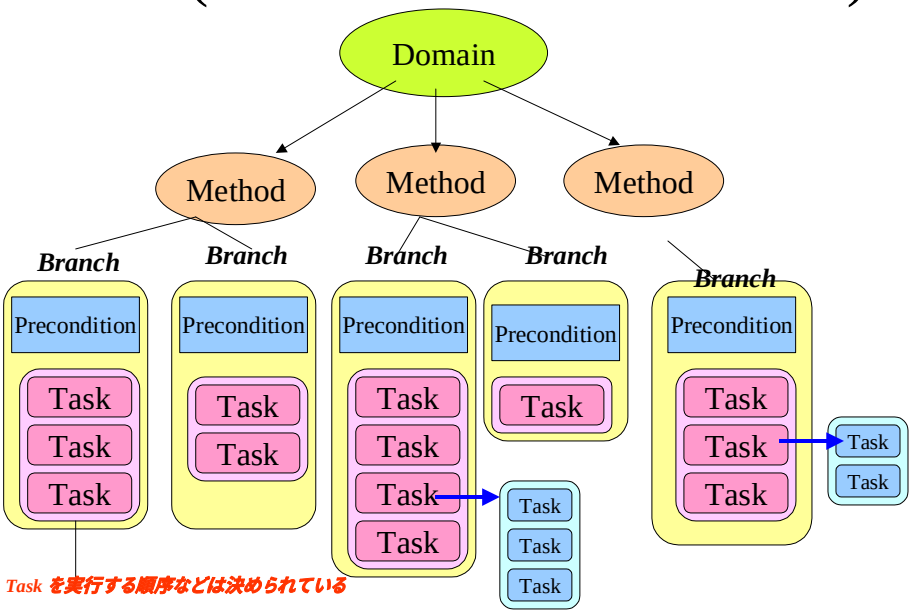


図　Killzone 2 における階層的タスク分解のイメージ　前提条件(precondition)をクリアしたブランチでタスクが階層的に分解される。同時に、各タスクは実行順序も決定される。

HTN (Hierarchical Task Network)

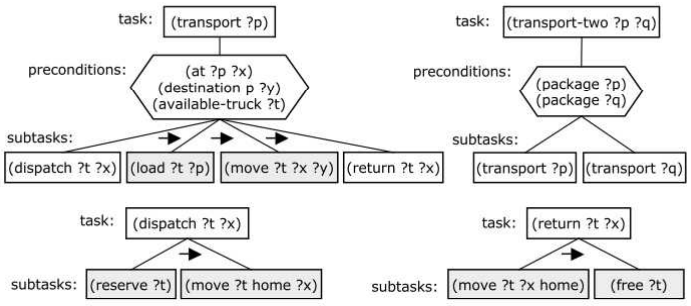


Figure 1: Methods for transporting a package ?p, transporting two packages ?p and ?q, dispatching a truck ?t, and returning the truck. Arrows are ordering constraints. The shaded subtasks are *primitive* tasks that are accomplished by the following planning operators: (load ?t ?p) loads ?p onto ?t; (move ?t ?x ?y) moves ?t from ?x to ?y; (reserve ?t) deletes (available-truck ?t) to signal that the truck is in use; (free ?t) adds (available-truck ?t) to signal that the truck is no longer in use.

Dana Nau et al., “SHOP2: An HTN Planning System”,
Journal of Artificial Intelligence Research 20 (2003) 379-404

図　一般的な HTN の解説(1) [11]　ここでは２つのタスクが前提条件を満たした場合の分解が解説されている。

HTN (Hierarchical Task Network)

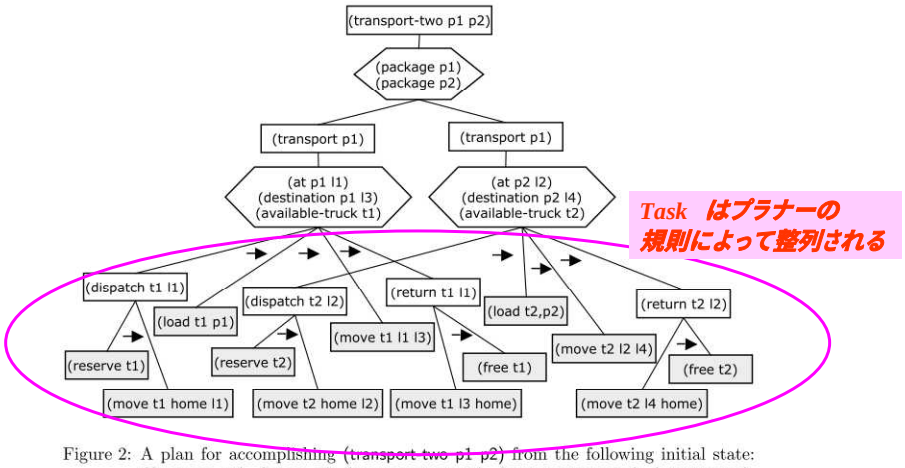
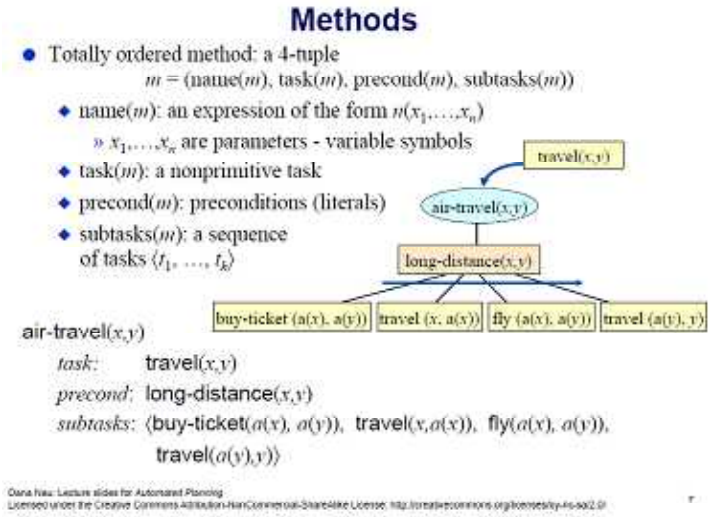


Figure 2: A plan for accomplishing (transport-two p1 p2) from the following initial state: {(package p1), (at p1 l1), (destination p1 l3), (available-truck t1), (at t1 home), (package p2), (at p2 l2), (destination p2 l4), (available-truck t2), (at t2 home)}.

Dana Nau et al., “SHOP2: An HTN Planning System”,
Journal of Artificial Intelligence Research 20 (2003) 379-404

図　一般的な HTN **の解説(2) [11]　一つ前の図の二つのタスク生成が同時に起こった場合、生成されたタスクの実行順序が調整され、再順列されていることがわかる。**

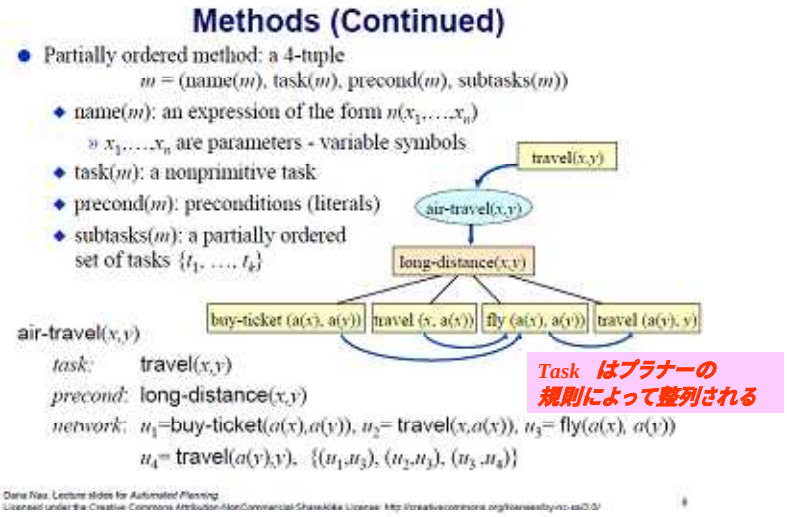
HTN (Hierarchical Task Network)



Dana S. Nau,"Hierarchical Task Network Planning", Lecture slides for Automated Planning:
Theory and Practice http://www.cs.umd.edu/~nau/cmsc722/notes/chapter11.pdf

図　一般的な HTN **の解説(3) [12] タスク生成と全タスクの実行順序が** subtasks(m)**に指定されている。**

HTN (Hierarchical Task Network)



Dana S. Nau,"Hierarchical Task Network Planning", Lecture slides for Automated Planning: Theory and Practice http://www.cs.umd.edu/~nau/cmsc722/notes/chapter11.pdf

図　一般的な HTN の解説(4) [12]　一つ前の図で生成されたタスクがネットワーク的に再順列されている。タスクネットワークの生成である

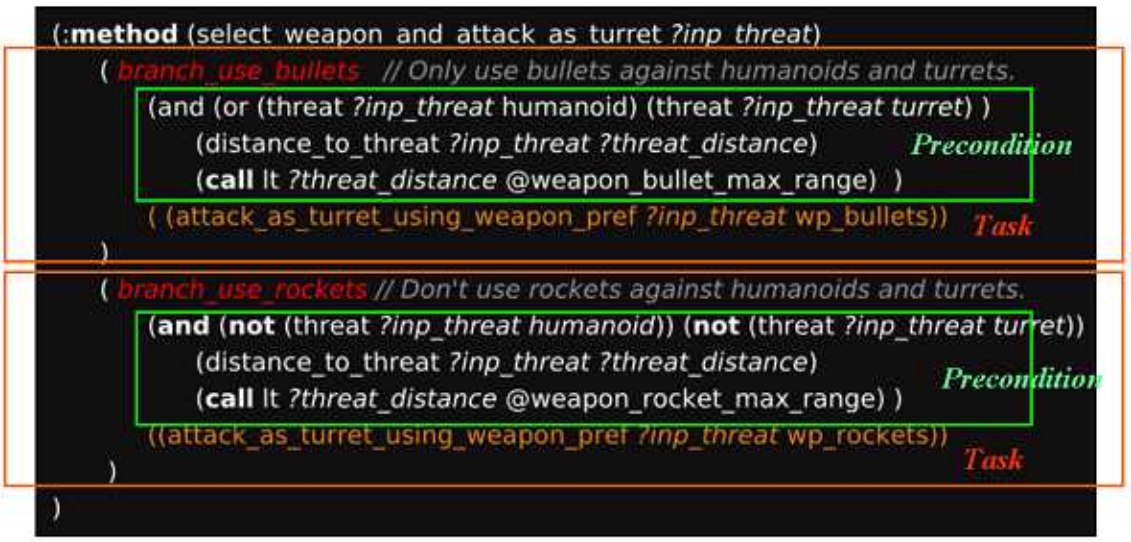


図　Killzone 2 の HTN プラナーの擬似コード[1]　前提条件とその下のタスクがセットになっていることがわかる

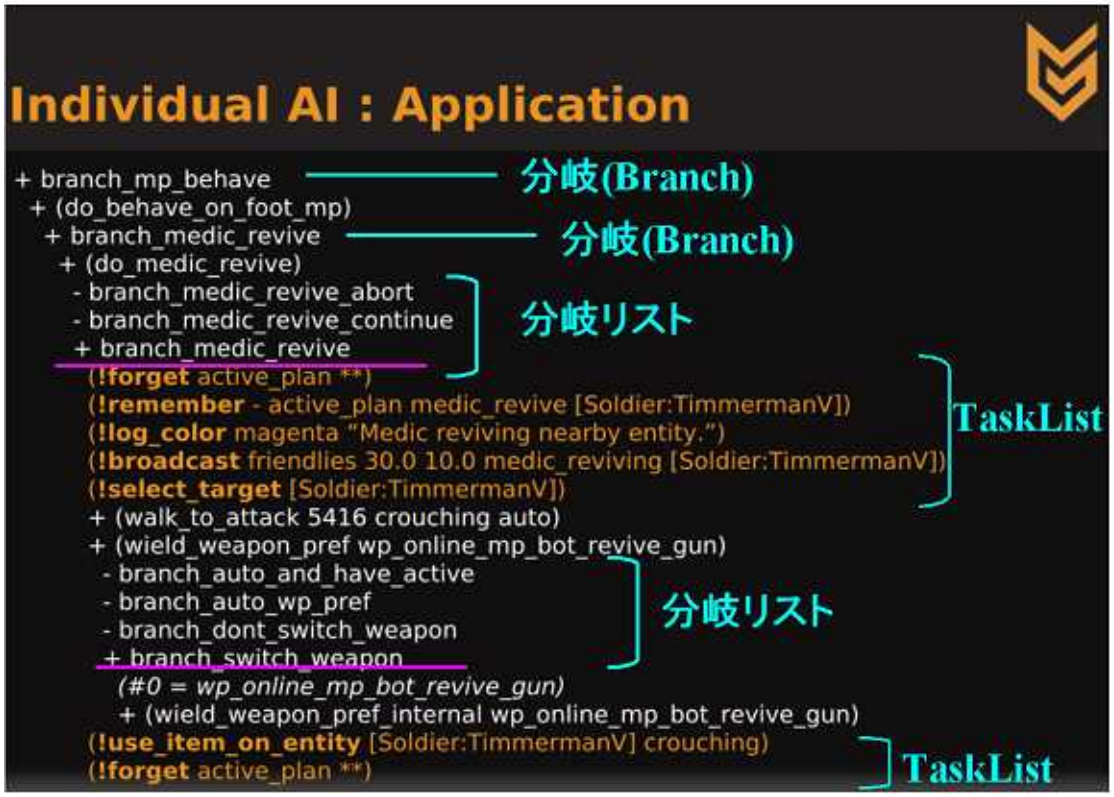


図　Killzone 2 の擬似コード[1]　タスクが実際に展開されて行く様子

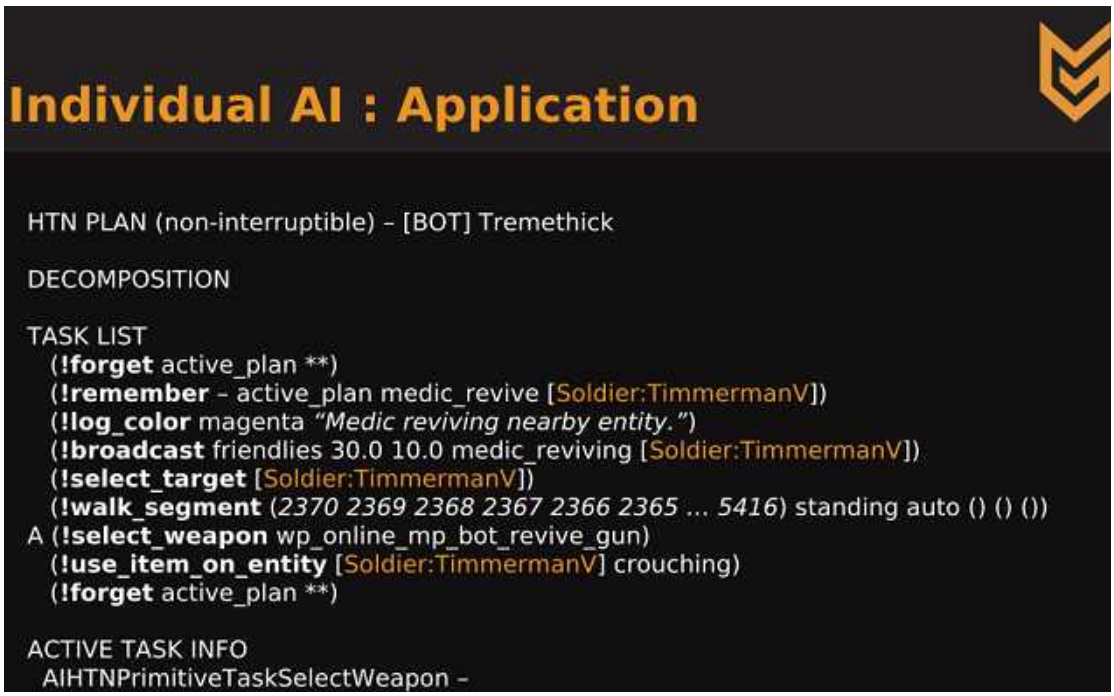


図　Killzone 2 の擬似コード[1]　生成されたタスクリスト　A は現在、Active なタスクを表現している。分解された順番に実行される。これは兵士を介抱しに行くプランである。

Squad AI

Squad AI の思考構造も、殆ど、各 AI の思考構造と同じである。ただ、命令(order)は、上位の「Strategy AI」から降りて来るものであり、また、下位の各エージェントからメッセージが届く仕組みである。それらの情報から現在の周囲の環境情報を再構成し、HTN プラナーによって、兵士への命令リスト（タスクネットワーク）をアウトプットする仕組みである。

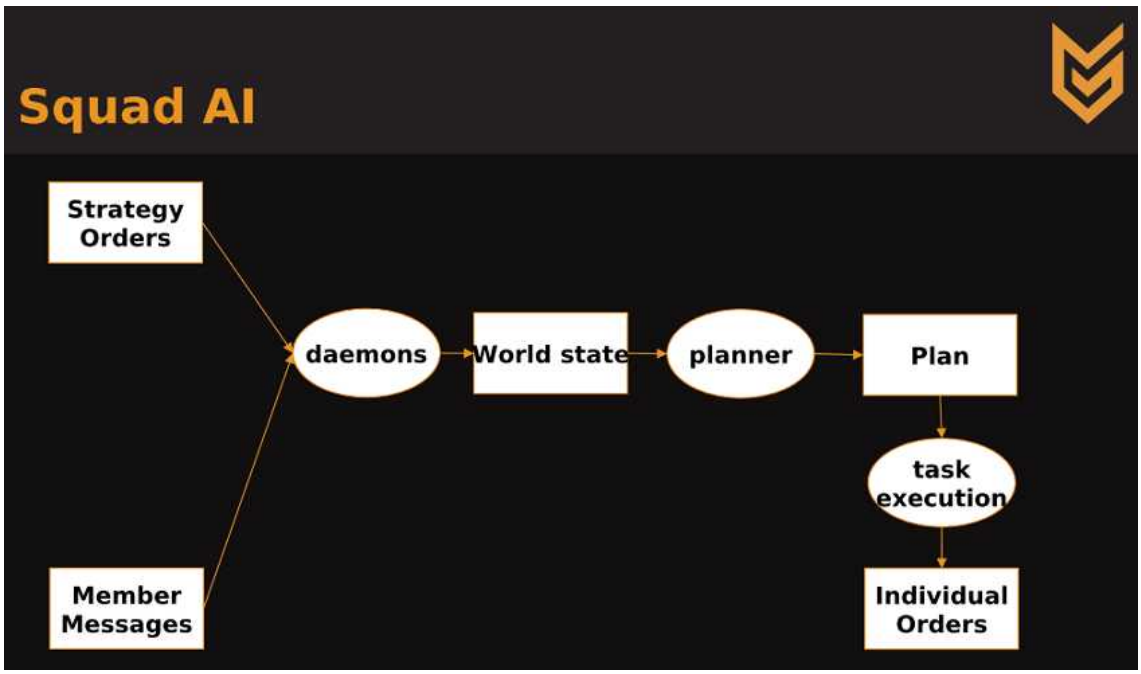


図 Killzone 2 の Squad AI の思考構成 [1]

動的なチーム生成

さて、ここで、GDC2008 で講演のあった Halo3 における Squad の生成について思い出してみよう(2008 年度の報告書で解説) [6,13]。Halo3 における Squad は、あくまでゲームデザイナーの手によってカスタマイズされたものであった。そして、そうして作った複数の Squad たちを、戦況が必要とする複数のチームタスクに割り当てたのであった。これは、比較的狭いエリアで戦場を作り出すという目標があった上で、Squad を形成する Halo3 のゲーム性の要求から来ている。一方、プレイヤー、AI が流動的にマップ全域を動き回る Killzone 2 においては、Squad は戦況に応じて動的に作られる。集められるメンバーは、目的に応じて、或いは、コアとなるメンバーからの近い距離にいる AI を Squad メンバーとして割り当てられる仕組みとなっている。

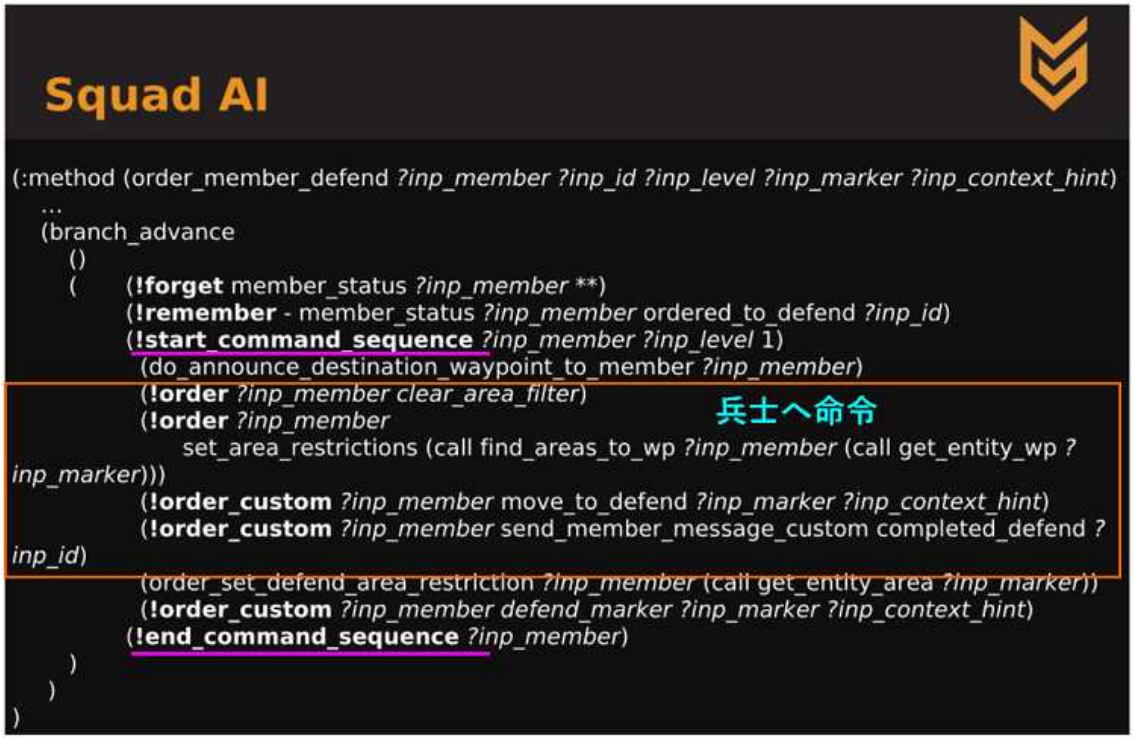


図　Killzone 2 Squad AI の HTN プラナーの擬似コード[1]　条件に合う兵士を探して命令を送っている

図　Halo3 と Killzone 2 の Squad 編成の比較[1]　Halo3 の Squad はデザイナーが事前に指定するが、Killzone 2 では動的にゲーム内で構成される

戦略思考

最上位の Strategy AI、Squad の戦略的思考について解説する。まず、戦略的思考の基礎となるのが、戦況全体の把握である。Strategy AI に各 AI から報告が上がって来る情報は、兵士がいる場所の情報の集合に過ぎない。戦況を把握するには敵兵士全体、味方兵士全体の勢力を常にモニターしておく必要がある（これはチートと呼ぶかどうかは、それほど簡単な議論ではない。一切、実際に知覚したデータ以外の情報を廃するならば、それは人間的にも正確ではない。人間は見えない情報も、とりあえずなんとか補完した上で行動している）。

まず、そういった戦局把握のために、マップ全体地形をウェイポイントで埋め尽くす（実は、こういった工程はそれほど自明ではない。ナビゲーションメッシュやウェイポイント自体を自動的に配列、かつ質が高くできればよいが、複雑な地形においては技術的な困難がある。Killzone 2　が自動的に配置しているのか、デザイナーが手動で置いているのかは不明）。勿論、このデータの第一の目的は各 AI が移動用に用いるデータである。各 AI のために、各ポイントに動的に情報を埋め込むことも可能だが（例えば、敵のいるウェイポイントに印を付ける）、Squad が用いる場所スケールとして小さ過ぎる。そこで、一定数のウェイポイントを 1 クラスター（塊）として、マップ全体をクラスター分割する。具体的には、一定数のウェイポイントがなるべく四角になるように自動処理して分割して行くのである。こうしてマップを一定の大きさに分割した後は、各クラスターの隣接情報を再構築する。隣のクラスターとのリンク情報を用いて、クラスターを要素とするネットワークグラフを構築する。こうして出来たグラフを戦略グラフ(Strategic Graph)と呼ぶ。そして、この戦略グラフ上に動的に戦局の情報を、影響マップ（Influence map）のテクニックを用いてモニターして行くことで戦局を把握するのである。

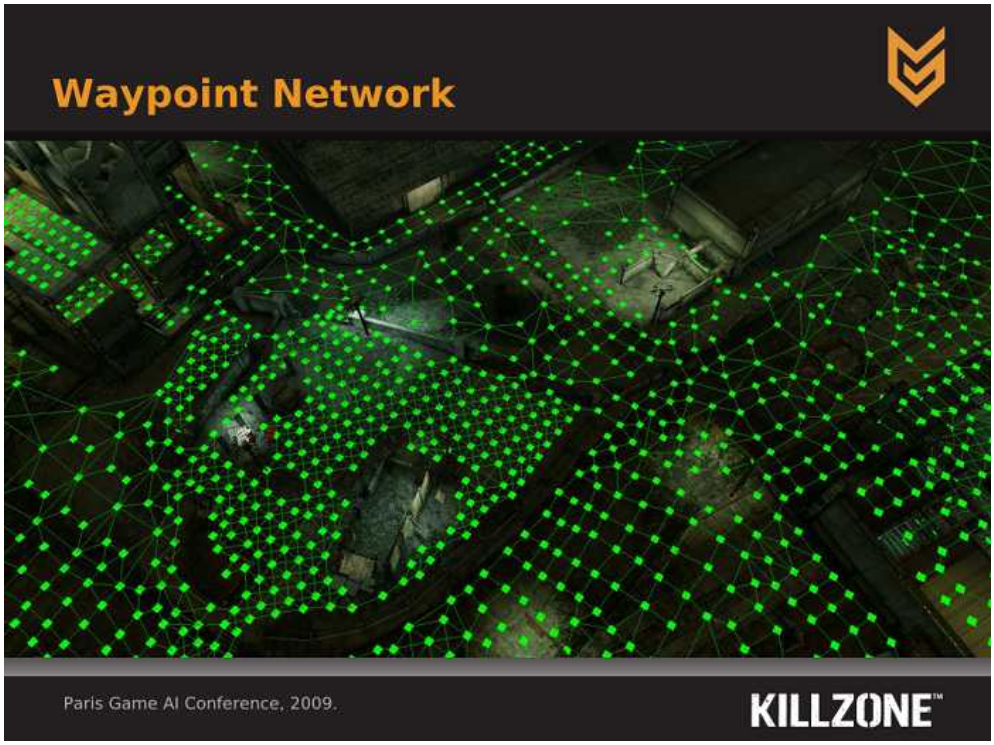


図 Killzone 2 **のウェイポイントマップ**[1]

図　ウェイポイントのクラスター化[1]　**ウェイポイントを一定数ごとに（なるべく）四角い形になるようにクラスター化（自動処理）**

図　戦略グラフの作成[1]　クラスタ同士が接するポイントで各クラスターを結ぶ。クラスターを一つの要素として作ったグラフを戦略グラフと呼ぶ

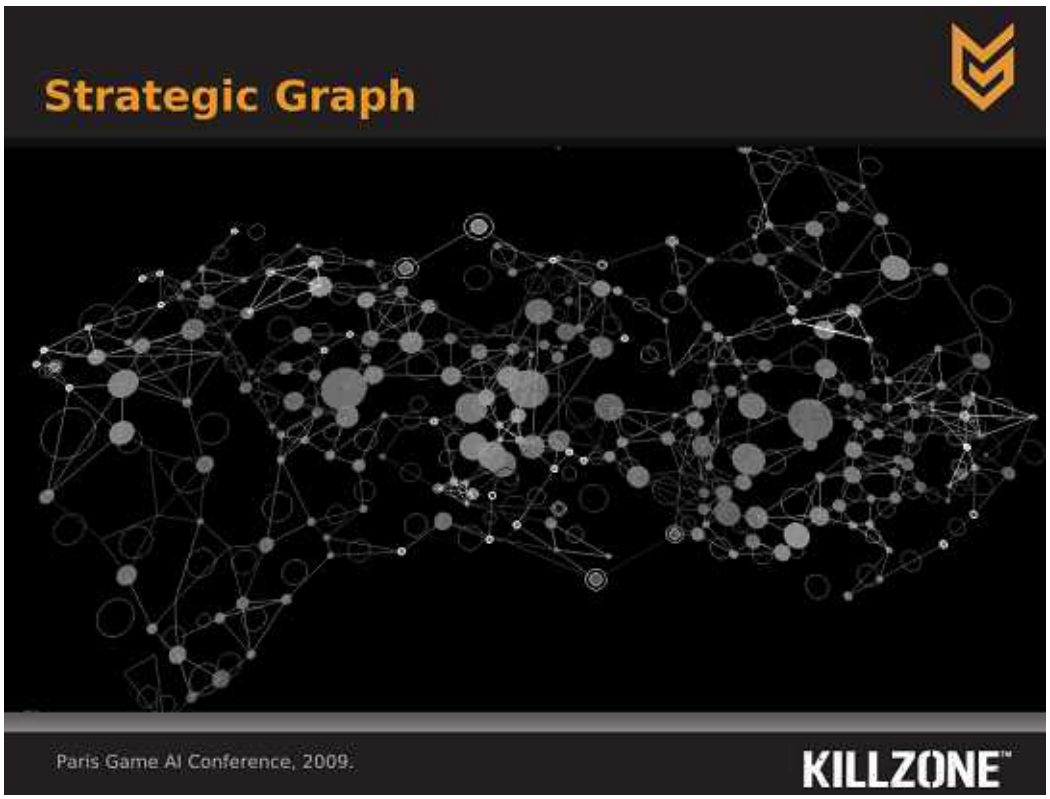


図 Killzone 2 **戦略グラフ[1]　戦略思考の基礎となる世界表現である。**

影響マップ

影響マップ(Influence Map)の方法は、戦略ゲームでよく用いられる方法で、マップ上の勢力情報などをモニターするために用いられる。影響マップ原理は単純で、影響を与える（例えば、脅威となる影響）キャラクターがいる位置から距離などに応じて、その影響度を減少させつつマッピングして行く。逆（マイナス）の影響を持つものがいる場合は、マイナスの値をマップ上にマッピングして行く。最終的に、複数のキャラクター、或いは複数のオブジェクトからの影響度を重ね合わせる（足し合わせる）ことで、マップ全体の「影響度マップ」が出来上がる。この影響度マップを基に、例えば、敵の勢力の強い地域や弱い地域が把握できるようになるという仕組みである。「Age of Empire」(Ensemble Studios)シリーズでは、影響マップを地形解析に用いており[14]、また、SimCity シリーズ(EA、Maxis) では、動的に街の発展アルゴリズムとして用いている[15]。

Killzone 2 における影響マップは、味方の死んだポイント、砲台、キャラクターの位置の情報を基に、敵、味方の脅威度を計算し戦略マップにアップデートされる。各 Squad を目的に応じたパス検索アルゴリズムを所持しており、エリアを限定（防衛 Squad なら味方陣地周辺など）したり、影響度マップの情報を利用して Sqaud のパスを決定する。例えば、敵と遭遇をすることが目的であれば、危険度の高いエリアを通るようなパスを決定すればよく、安全な道を通りたければ、敵の脅威度の少ないパスを辿ればよい。

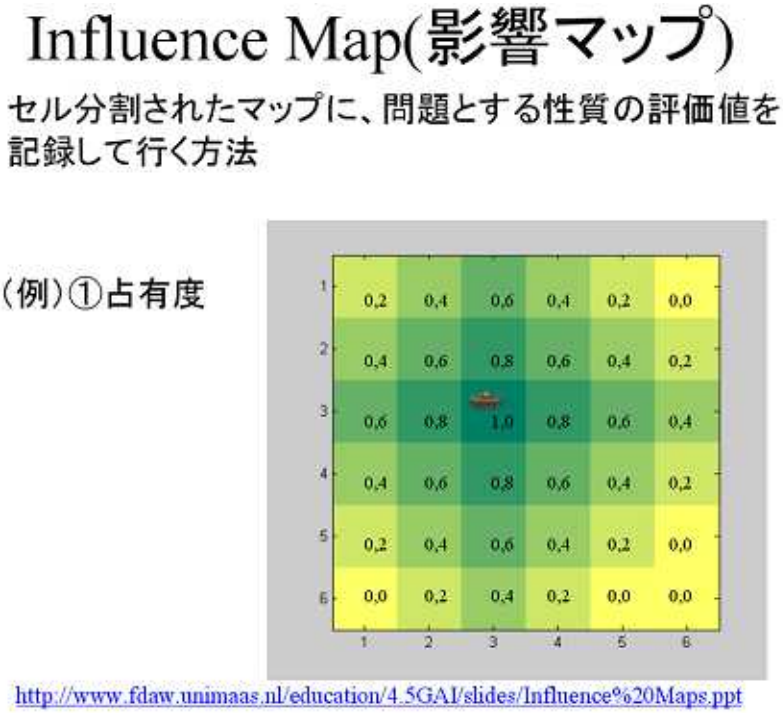


図　影響マップの一般的解説(1) [16] 戦車の占有度の影響力のマッピング　この他の敵、味方の戦車の影響度などを加味しつつ全体の影響マップが出来上がる。

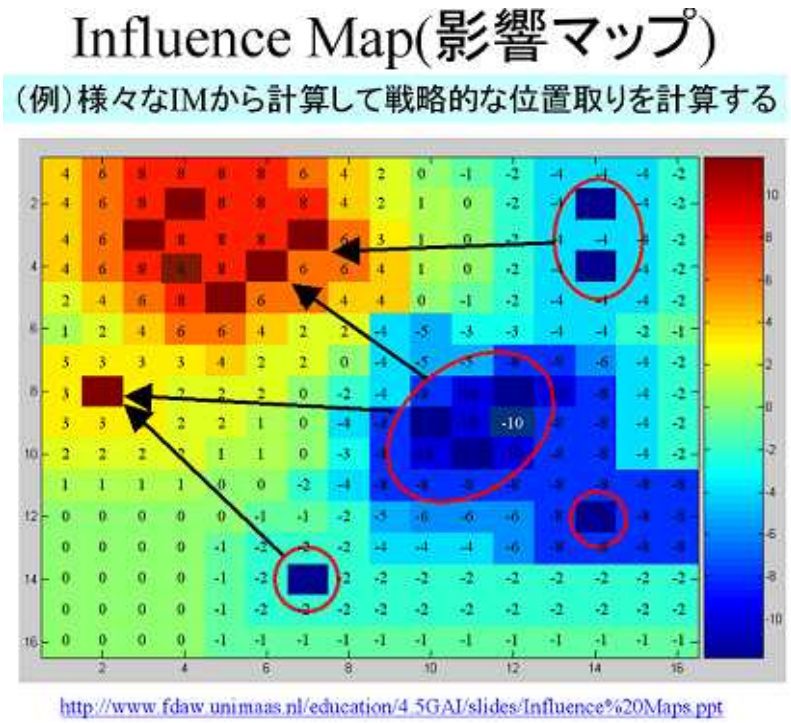


図　影響マップの一般的解説(2) [16] 敵の勢力が赤、味方の勢力が青として影響マップが形成されており、この情報を基に戦略的移動を行う

図 Killzone 2 の影響マップ[1]　先に作成した戦略用グラフ上にリアルタイムに敵、味方勢力の影響度をマッピングして随時アップデートする。

Killzone 2 における戦略的パス検索

スカッド毎にコストと影響マップを使って戦略パスを見出す

図　Killzone 2 の戦略的パス検索[1]　例えば、影響度をもとに敵と遭遇しにくい、或いは遭遇しやすいパス、或いは防衛エリア内でのパスなど、ミッションに特化したパス検索を行うことが出来る。

Killzone 2 の階層構造

以上のように「HTN プランニング」「戦略マップ上の影響マップ手法」が Killzone 2 の主要な技術であるが、この節では、Killzone 2 に見られる AI システム全体の性質について考察してみよう。

Killzone 2 AI では AI システムの随所に階層構造が取り入れられている。

- 「Strategy AI - Squad AI - Individual AI」階層構造、
- 「HTN プラナー」階層構造
- 「Strategy Map - Waypoint Map」階層構造

という３つの階層構造が用いられている。

階層構造を導入することは、AI にスケーリングを与えることである。スケーリングとは、ここでは複数の尺度で物事を捉える能力のことである。高度な知性であればあるほど、時間、空間における複数のスケーリングを持っている。例えば、時間においては、瞬時、３

分、１時間、１年、一生など、知性はさまざまな時間スケールの思考を行うことが可能であり、また、空間においても、身の周り、街、国、地球、など、様々な空間スケールの思考が可能である。こういったスケーリング能力を持つおかげで、人間は、世界の様々な諸相を捉え思考することが出来るのである。

こういった高度な AI を実現するには、逆に、階層構造を人の手で AI に導入すればよい。上記の階層構造はそれぞれ、組織としての階層構造は時間と空間のスケーリングを（上位になればなるほど長期大局、下位になるほど、短時間局所）、HTN は時間的階層構造（長期プランから短期プランに分解して行く）、Strategy Graph は、Waypoint という局所的スケールから、大局的な戦略的なスケールへと AI の思考を移す空間階層化の機能を実現しているのである。

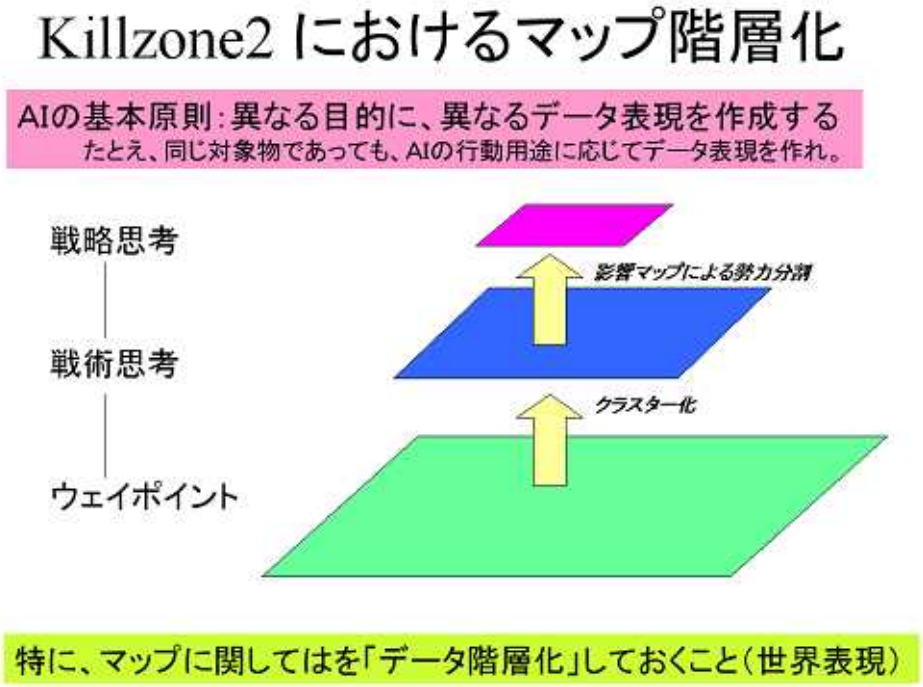


図　Killzone 2 における地形データ階層化　もとの地形からまずウェイポイントデータへ、さらにそれをクラスター化した戦略マップ、そして、その上にさらに影響マップを用いて勢力図に分割している。こうした階層化は、各階層スケールの思考を可能にする。例えば、ウェイポイントは詳細な地形に応じたパス移動を実現し、戦略マップはクラスターを粒度とする思考を可能にし、影響マップはマップ全体の勢力図を見越した戦略思考を可能にするのである。人工知能の標語に「異なる目的には、異なるデータ表現を作成せよ」という言葉がある。知能の機能は、事物の表現と思考がセットになって初めて実現する。思考だけでは知能ではない。よって多層な表現と各層毎の思考は高度に構造化された知能を可能にするのである。

Killzone 2 AI のまとめ

Killzone 2 のオンラインモードにおける AI の仕様についてまとめて来た。オンラインゲームは複数のプレイヤーが相手であるから、AI システムは、常に敵・味方問わず、プレイヤー全体の動向を把握しながら意思決定を行う必要がある。そこで、戦略マップによって戦局をモニターしながらスカッドの大局的行動を決定し、個々の AI は HTN プラナーによって局所的な戦闘の思考を精緻に作りこむという方法を取った。実際、オンラインの長期的な戦闘に耐えるためには、プラナーによる行動のスケジューリングなしに人間らしい行動を構築することは難しい。Killzone 2 は、前作の Killzone の上に、これまで伝統的に使われて来た影響マップの手法を新しい形で応用すること、各 AI に新しくプラナーを導入することで、全体として柔軟かつ堅牢な AI システムを作り上げた。

The Sims の AI の技術

The Sims というゲームは先の Killzone 2 と対照的に、平和な街の中で自律的に動作するたくさんの人（AI）がインタラクションしている様子を見ながら、時々、命令したり、オブジェクトを配置したりする、所謂、箱庭ゲームである。そこで必要とされる AI 技術も自然、FPS と違ったものとなり、また、同じ技術でも用い方が全く違って来る。そういった意味で、この両極にある２つのゲームの AI 技術は鋭い対照を成す。この対照を解説することは、ゲーム AI 技術全般に示唆を与えるものである。以下では、GDC2010 における講演[2]を踏まえて、The Sims の AI システムをシリーズを通した総合的な解説を行う。

The Sims が求める AI

街や家の中で社会的な生活を営む AI の街を再現する、そして、プレイヤーがそれを観察しながら、上から影響を及ぼす、それが The Sims というゲームである。AI は、0.1 秒やわずか 10cm で生死を分かつ FPS の戦場ではなく、ゆったりとした時間の中で、様々な人（別の AI）や物とインタラクションしながら生活を営む。「AI の人（他の AI）や物とインタラクション」を実現することが、The Sims の AI の本質と言っていいだろう。FPS の作る戦場という極限状態では、単に遮蔽物としてしか意味を持たなかった様々な構造物も、或いは敵・味方としてしか意味を持たなかった他の AI も、The Sims では、多様なインタラクション、多様な人間関係の中で、新しく多様な意味を持ち始める。そういった人や物との多様な関係性を積み重ねて行くところに、The Sims のゲームとしての奥深さがあるのである。

そしてゲームとしては、このような AI によって「街の中で自然発生的にさまざまなドラ

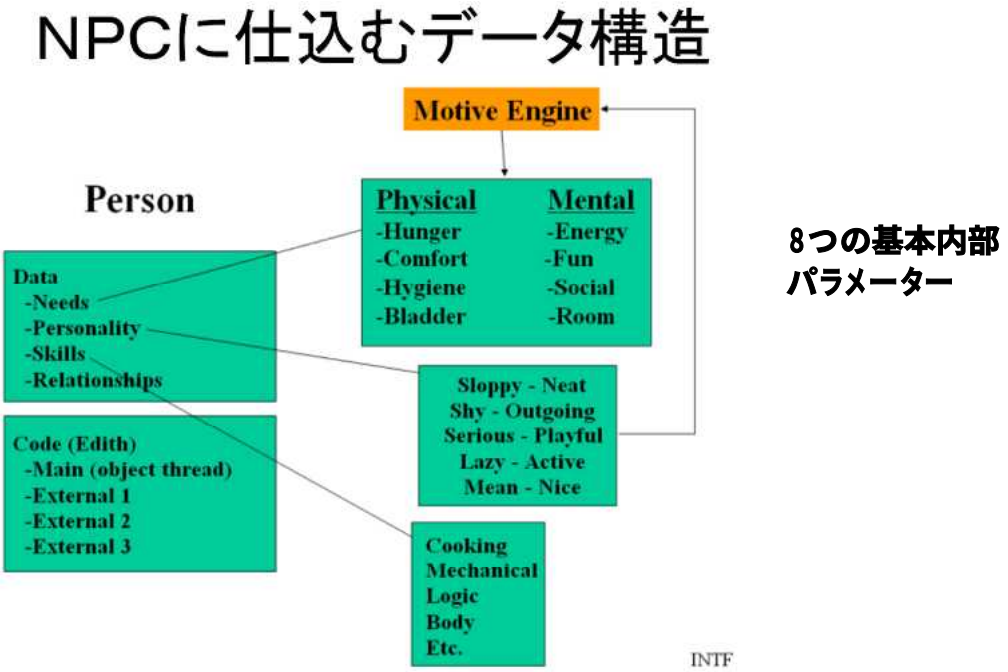
マが産まれる」ようダイナミクスを実現することを目標にしている。

The Sims AI の原理

The Sims 1,2　AI の原理は昨年の報告書でも解説しているが[17]、The Sims 3 への変化を見るために、もう一度総合的に解説しておこう。

FPS の AI は殺人衝動と保守本能という極限化された内面状況によって駆動しているが、日常系の AI は、多様な内部パラメーター(動機パラメーターとも呼ばれるが、ここでは内面パラメーターと呼ぶことにする)が動的に変化するシステムとして内面を形成している。この内面は、精神的（メンタル、mental）パラメーター（楽しみ、部屋の快適さ、社交、エネルギー)、生理的（フィジカル、Physical）なパラメーター（尿意、空腹、衛生、快適さ）の８つで構成されている。The Sims の AI は、これらが変動パラメーターとして保持しており、環境、行動、時間がこれらのパラメーターを変化させる。直接このパラメーターを変化させることは出来ないが（ちょうど人間が自分の感情を自由にコントロールできないように)、AI は行為を通じて自分の内面の状態を変化させることが出来る。例えば、風呂に入れば衛生パラメーターが上がる、などである。また、自律的に変化するパラメーター（空腹）や外部環境によって変化するパラメーター（快適さ）がある。

これらのパラメーターにウェイトをかけて足し合わせた値（Mood）を総合幸福度として、AI は基本的に、この総合幸福度を最大化する方向の行動を選択して行うのである(Motive Engine)。



Ken Forbus, “Simulation and Modeling: Under the hood of The Sims” (NorthWerstern大学、講義資料)
http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm

図 The Sims 1,2 における AI の内部の仕組み[18]　８つの内面パラメーター（生理的な欲求と精神的な欲求、各４つからなる）の変動によって駆動する（Motive Engine)。この基本構造は The Sims 3 でも同様である。

The Sims の行動原理

The Sims の AI の行動は、一つの関係で結ばれている「内部パラメーター」「ＡＩの行為」「環境のオブジェクトは」からなるシステムによって駆動される。これを解説する。

各オブジェクトには、そのオブジェクトに対して行い得る行為（アフォーダンスされた行為）の情報と、その行為による内部パラメーターの変化が定義されている。例えば、「風呂」オブジェクトには「洗う」「入る」などの行為リストが付随されており、そういった行為が上記の８つの内面パラメーターをどう変化するかが設定されている。つまり、「風呂」(Bathtub)に「入る」(Take Bath)ことで、内面パラメーターが変化し、その結果、Mood の値が変化する。

AI は常に周囲の複数のオブジェクトをリストアップし、さらに、その一つ一つのオブジェクトに関係する行動（例えば風呂ならば、洗うと入る）をリストアップし、この中から、Mood を最大化する行為を選択する。つまり、AI は、周囲のオブジェクトに対して行い得る行為とその効果を全て事前に計算した上で、Mood を最大化する行為（つまり最大のプラス変化をもたらす行為）を選択するのである。

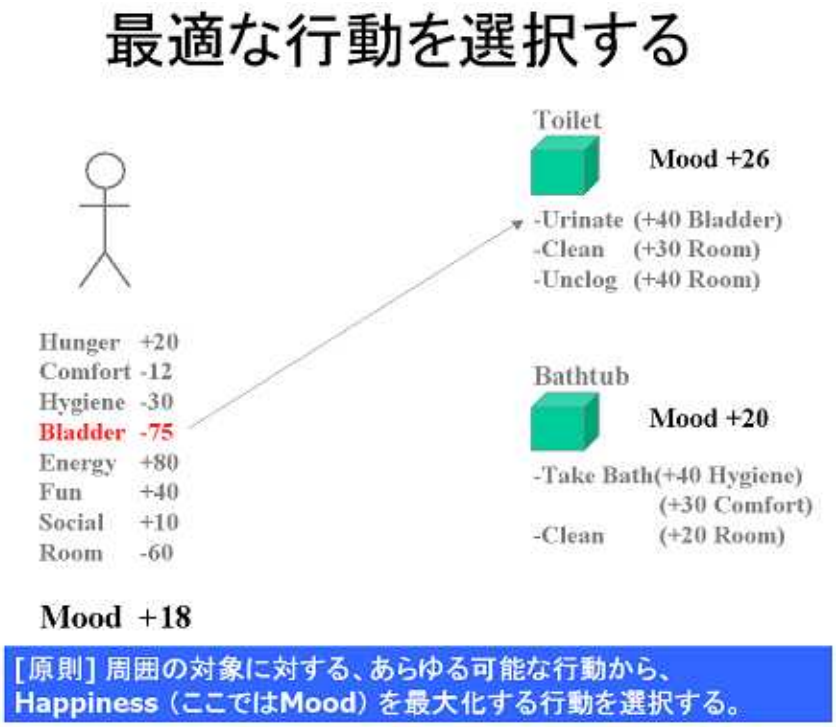


図　The Sims における最適な行動の選択原理[18]　オブジェクトには AI がそのオブジェ

クトに取りえる行動がリストされており、各行動は、AI の内面パラメーターを変動させる。その中で、Mood を一番上昇させる行為を選択する。ここでは、+40Bladder（尿意解消）の変化をもたらす Urinate(排尿する)が、総合幸福度を+26 にするため、この行動が選択される。

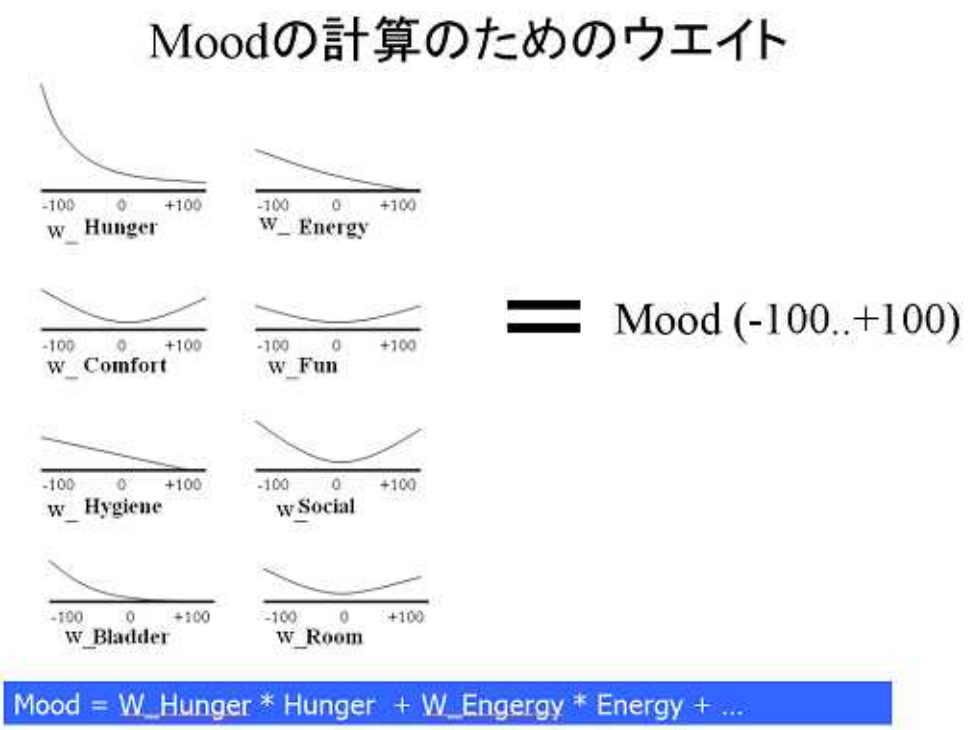


図 Mood 計算とウェイト[18]　上記の８つのグラフは、８つの内面パラメーターが各値においてどれだけ Mood に貢献するかを表す。例えば、Social パラメーターは、W_Social というグラフは Social の値が X_Social の時、W_Social(X_Social)＊X_Social だけ Mood に貢献する、 Mood は同様に計算した全てのパラメーターの足し合わせである。

さて、ここでウェイトのグラフに注目して頂きたい。このウェイトのグラフは、各内部パラメーターが、Mood に対してどれぐらいの度合いで貢献するかを表している。例えば、Hunger パラメーターの値が今 70 とすれば、70 の時のウェイトグラフの値を W_Hunger(70)として、W_Hunger(70)*70 が Mood に貢献する値である。つまり、

Mood = W_Hunger(X_Hunger)*X_Hunger + W_Energy(X_Energy)*X_Energy + ...

という値が Mood の値になるのである(X_... は各パラメーターの現在の値とする)。Bladder、Energy, Hunger などは、値が大きくなると、ウェイトの値が０に近くになっている。つまり、こういった欲求は満たされれば気にしないし（Mood に貢献しない)、不足

するとひどく幸福度を損なう（Mood にマイナスの作用をもたらす）ように、ウェイトグラフが設計されているのである。Hygiene(衛生)も同様である。一方、Fun(楽しみ)、Social(社交)といった精神的な楽しみは、満たされれば満たされるほど大きな幸福度を生み出すようにウェイトグラフが設計されているのである。こういった設計は人間の感情的性質を表現するように表現されているのである。

限界効用逓減（ていげん）の法則

また、ここからもう一つわかることは、一旦、あるパラメーターが満たされてしまうと、そのパラメーターを上昇させても、幸福度にさして変化がないということである。例えば、Social が 60 から 90 による変化は

$$\Delta = W\_Social(90)*90 - W\_Social(60)*60$$

である。一方、満たされていないパラメーターを満たす方が、幸福度は大きくアップする。例えば、Hunger が -80 から -50 になったとすれば、

$$\Delta' = W\_Hunger(-50)*(-50) - W\_Hunger(-80)*(-80)$$

の変化分が Mood をアップさせる。つまり、一旦、あるパラメーターがある程度満たされると、そのパラメーターよりは、満たされていないパラメーターを上昇させる方が、全体の幸福度（Mood）に貢献する度合いが大きいということである。つまり、満たされていないパラメーターを変化させる行為が選択される傾向が強い。これも人間の性格をよく現している。最初のビールの一杯は、常にそれ以降の二杯目、三杯目よりも一番おいしいのである。これを経済学では、限界効用逓減の法則と呼ぶ。

シミュレーションの負荷低減の方法

ここからはプログラミングのテクニカルなところであるが、この原理だと、AI の周囲にある全てのオブジェクトに対する Mood 変化率を計算しないと、行為を選択できない。実際、The Sims1,2 では、あらゆる計算を行っていた。しかし、殆どのパラメーターは普段、ある程度満たされている（らしい）。そういったパラメーターを変化させても、それほど Mood は変化しない。そこで、満たされていないパラメーターを上昇させる（Mood が大きく変化する）行為のみ選択して評価する、という仕組みを作れば、わざわざ、満たされてしまったパラメーターを変化させる行為を評価する計算の手間を省くことが出来る。

こういった計算の効率化を行うため、The Sims 3 からは、内面パラメーターと行為の対応表をあらかじめ作っておき、変化させるべきパラメーター（満たされていないパラメーター）から行為を検索できる仕組みを導入している。この検索によって、満たされていな

いパラメーターから選ばれた行為の集合を評価することで、最も幸福度をアップさせる行為を効率よく見つけ出すことが出来るのである。

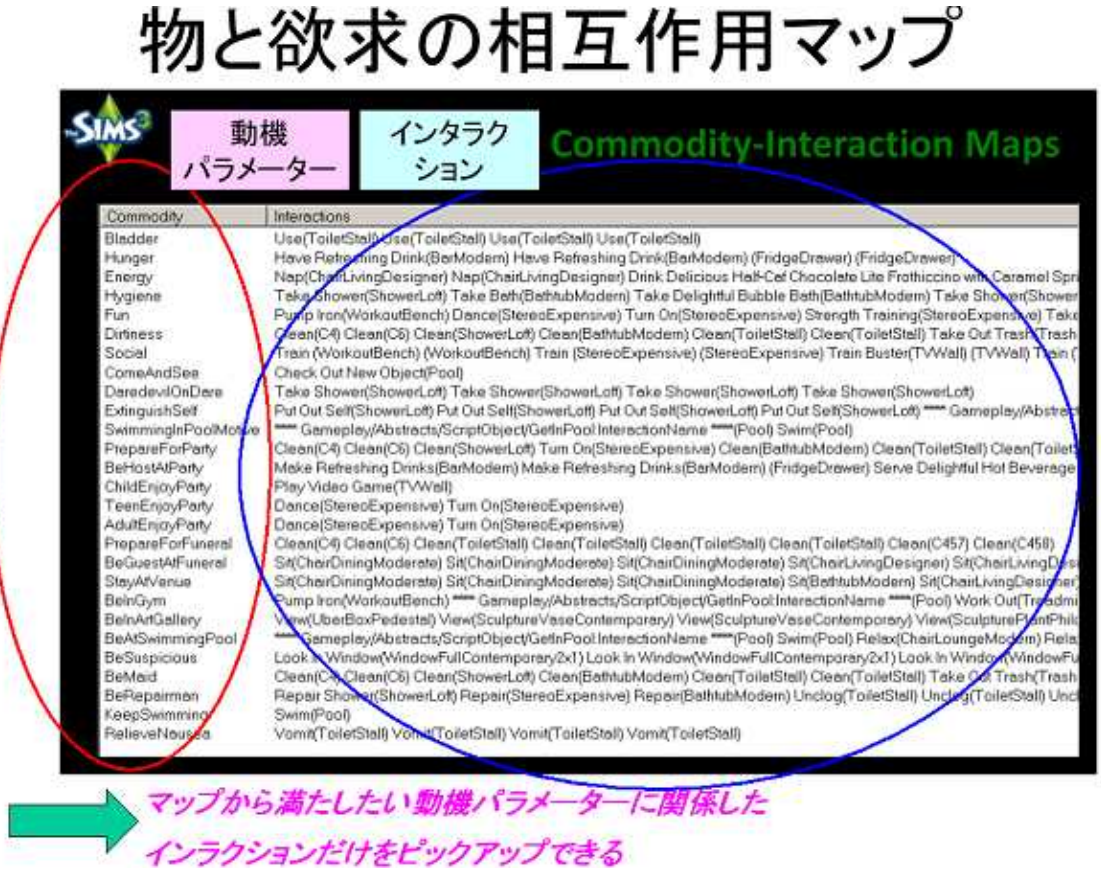


図 The Sims 3 における内面パラメーターと行為の対応表[2]　The Sims では、経験的に、ある種のパラメーターは殆どの場合、大きく満たされている。つまり、そういったパラメーターを変化させる行為を行っても、Mood を上昇させることは僅かである。そこで、満たされていないパラメーターがあれば、そのパラメーターを変化させた方が Mood は大きく変化するのであるから、そういったパラメーターのみの着目して、行為を評価すればよい。そこで、内面パラメーターと行為の対応表から、そういった行為のみをピックアップして、周囲のオブジェクトからそういった行為のみをまず評価して見るのである。

　また The Sims 3 からは、Hunger パラメーターなど、自明な生理的欲求は、ある程度欲求が強くなると、自動的に満たすような処理(Auto-satisfaction)が施されている。これによって毎回の食事のようなゲーム内で何度も繰り返さなければならない行為に煩わされないで済む。

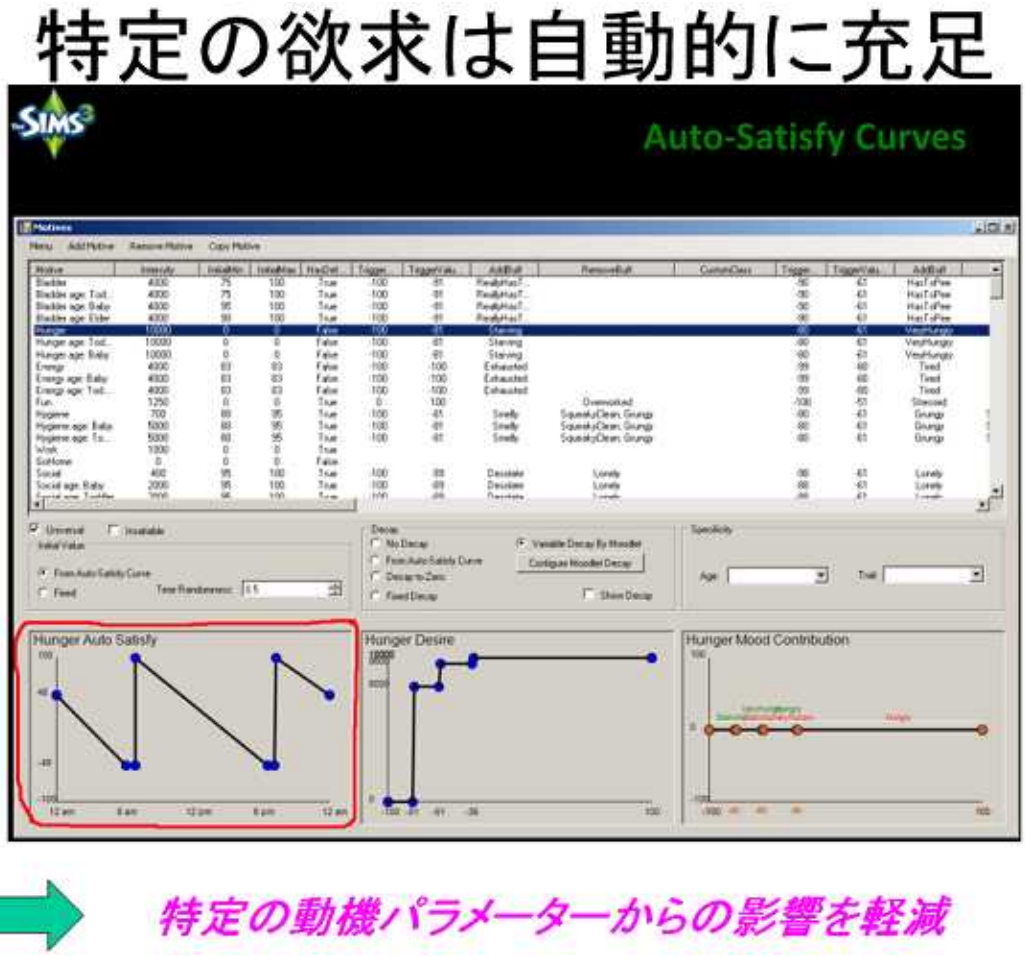


図　Hunger パラメーターの自動充足機能[2]　**Hunger パラメーターは AI に食事を取らせようとするが、ゲーム的に一日何度も食事をするのは煩わしいので、自動充足されている。**

AI 全体の制御アルゴリズム

The Sims 3 は、The Sims 1,2 より、広大な世界におけるシミュレーションを目指している。そこで、The Sims 3 では、AI シミュレーションにかかる計算負荷を軽減するため、以下のような工夫を行っている。The Sims 1,2 では、対象とするエリアで、各エージェント(AI)に対して、あらゆるインタラクションを考慮してシミュレーションしていた。しかし、The Sims では、あるエリアのエージェントに対して、そのエージェントをインタラクションの対象とするエージェントをピックアップして、そのインタラクションをシミュレーションする。このように、The Sims 3 のエージェント同士のインラクション全体を階層的に効率化している。

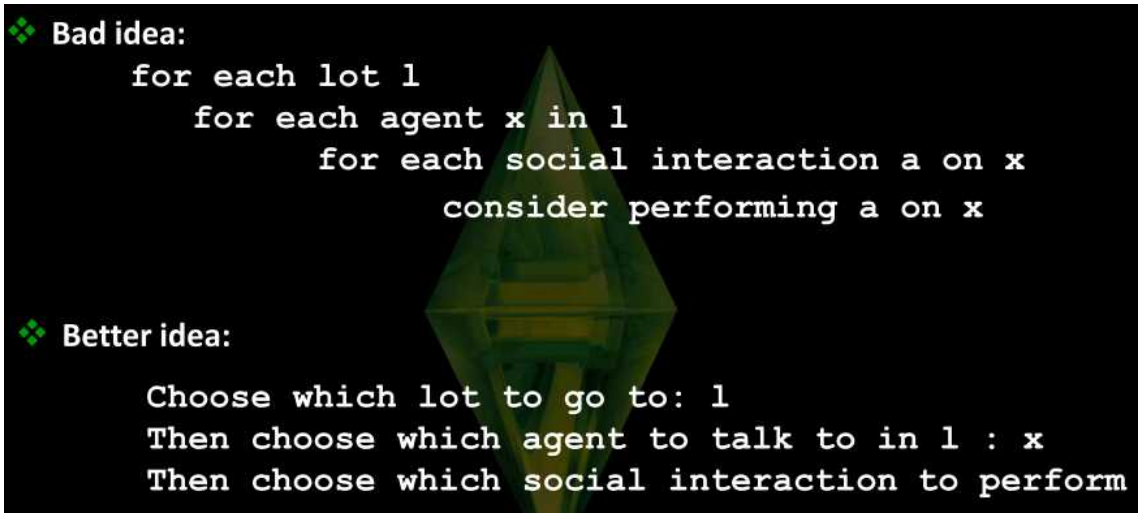


図　The Sims 1,2 から The Sims 3 へのインタラクションのシミュレーションアルゴリズムの変化[2]　階層化され効率化されている

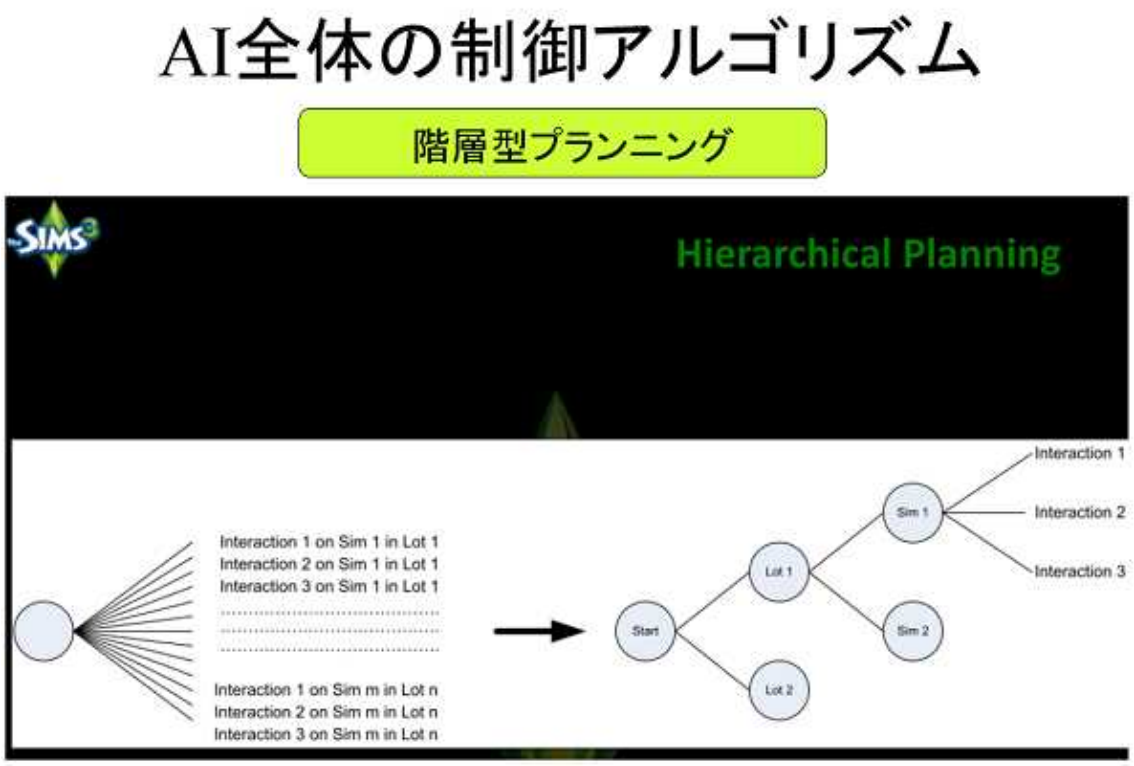


図　The Sims における階層型プランニング[2]　上記のアルゴリズムはエージェント全体のシミュレーション制御を階層型プランニングを用いて効率化していると言える。全体の状況を局所に分解してシミュレーションして行くのがコツである。

The Sims AI　の発展、個性化へ

The Sims は人間の社会的なインタラクションをエンターテインメントすることに踏み込んだタイトルである。プレイヤー（人間）が人間（AI）社会を見て楽しむのであるから、各 AI には、人間的な社会的欲求のリアリティーが求められる。The Sims 3 の講演では、こういった人間の欲求を明確に捉えるため、「Maslow の欲求階層図」が解説のために提示された。「Maslow の欲求階層図」は人間の欲求は段階的な階層があるという説に基づき、下層から生理的欲求、完全欲求、社交的欲求、個性的欲求（自己実現、尊敬を受けたい）、道徳的欲求といった階層構造を為す図である。つまり、下層に行くほど肉体的な欲求であり、上層へ行くほど社会的な欲求、或いは抽象度の高い欲求となっている。The Sims の AI も、この下層から上層へ向けて順番に実装されて来たと推測できる。そして、The Sims 3 の目標とは、「AI に特徴（Trait）を与えること」「長期的な目標を与えること」である。つまり、Maslow の図で言えば、第二段の欲求を AI に実装することとなる。

図　Maslow の欲求階層図[2]　人間の持つ欲求が階層的に表現されている。下位の欲求が満たされれば一つ上の階層の欲求を満たされたくなる。Trait Satisfaction　（個性的満足）

とは、自己の個性や特徴を社会や人に認めて貰いたい、ということである。

The Sims 3 においてそれぞれの AI に特徴を与えるのは以下のような方法である。まず、個性化とは、ゲームの場合、ユーザーがそれを見て分かるようなものでなければ意味がない。そこで、個性化を「普遍的な内面の構造」と「社交（人との係わり合い）における行動」に分けて実装することにする。

まず「普遍的な内面の構造」の特徴付けを解説する。８０個ほどの特徴(trait)を準備する。これらの特徴は例えば、「盗癖がある」「ロマンティックな TV を見る」「インターネットフォーラムでチャットする」とか、そういったユーザーから見て行為的な特徴のことである。そして、これらの特徴に対応する、動機（内面パラメーター）を設定する（これは先に解説した８つの基本的な内部パラメーターと行為の対応と同様である）。例えば、「盗む」「ロマンティックなＴＶを見る」パラメーターなどである。各 AI を特徴付けを行うときには、この８０個の中からランダムに５個の特徴を選んで、そのキャラクターの特徴とする。これまで、The Sims 1,2 では、あらゆる AI がこれまで解説して来たような８個の共通の内面パラメーターのみを持っていたところに、５つのパラメーターが増設されることになる。また、こういった追加的な特徴パラメーターは、特に発動する必要のない時間帯があるため、一日のうちで、追加したパラメーターを取り除いて動作させる時間帯を設置している。例えば、「ポテトチップスを食べながらＴＶを見る」という特徴は、休日の昼間や夜以外はふさわしい行為ではない。つまり、時間帯によっても AI の特徴が変化する仕組みになっている。

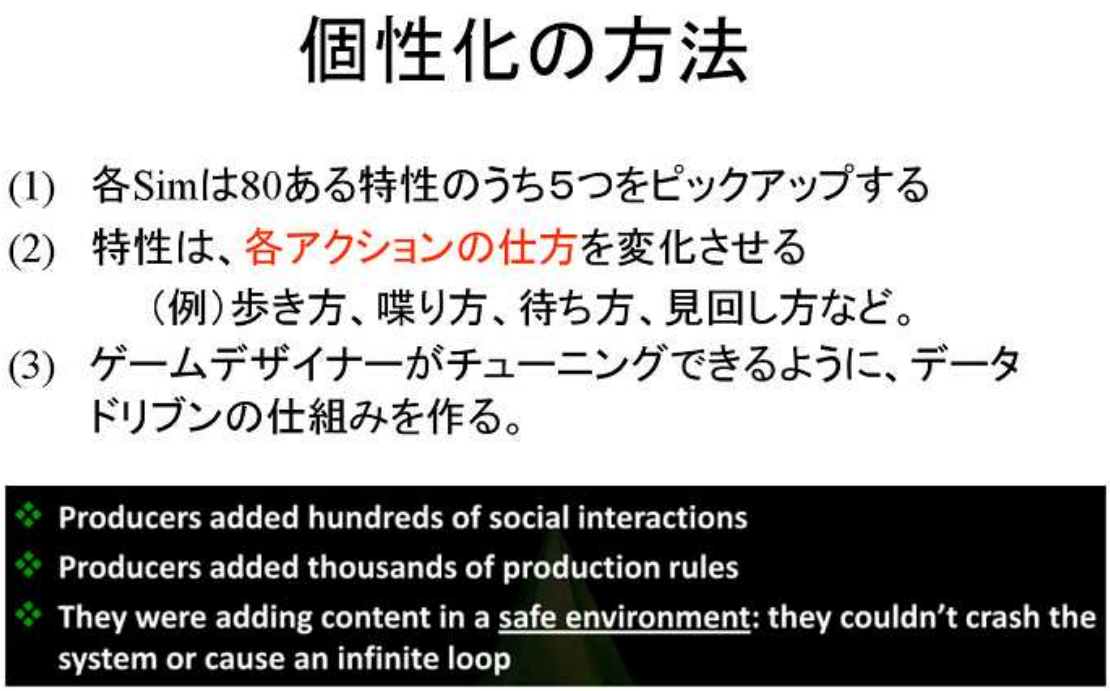


図　The Sims 3 における個性化の方法[2]

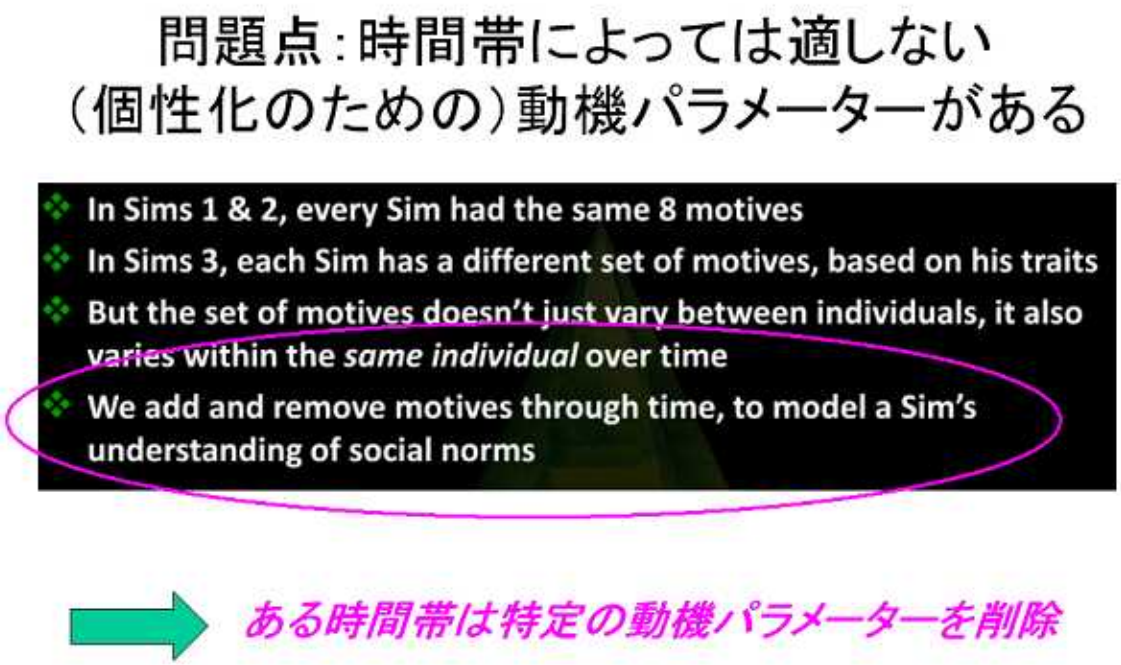


図　時間帯によっては特徴付けの内面パラメーターは除かれる[2]

さて、もう一つの「社交における行動」の各 AI におけるカスタマイズであるが、これはプロダクション・ルールを用いて行う。プロダクション・ルールとはここでは、「もし～だったら～する」といったルールの集合である。こういったプロダクション・ルールの全体は、データとしてプログラムから読み取らせるようになっている。つまり、コードを変えることなく、このルールのデータを読み取ることで駆動される仕組みになっている（データ駆動型システム）。このため、非プログラマーであるゲームデザイナーやプロデューサーがプログラマーの手を煩わせることなく、自由にルールを追加することが出来る。このような分業体制のもとで、数百個のインタラクション（行為）と数千のプロダクション・ルールのデータがプロデューサーによって作成された。

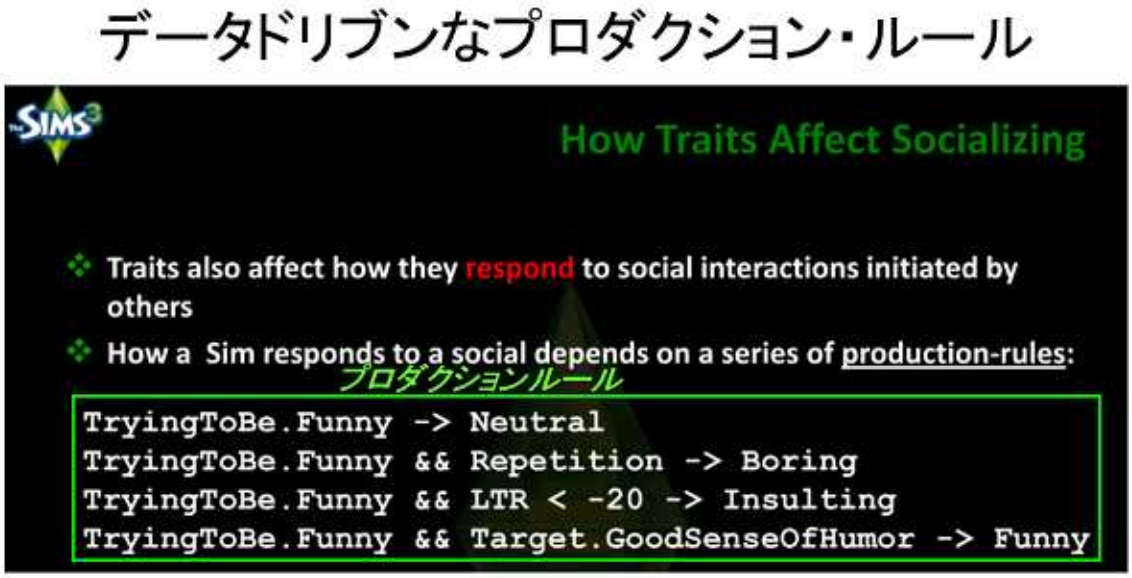


図　社交における行為をカスタマイズするプロダクション・ルール[2]

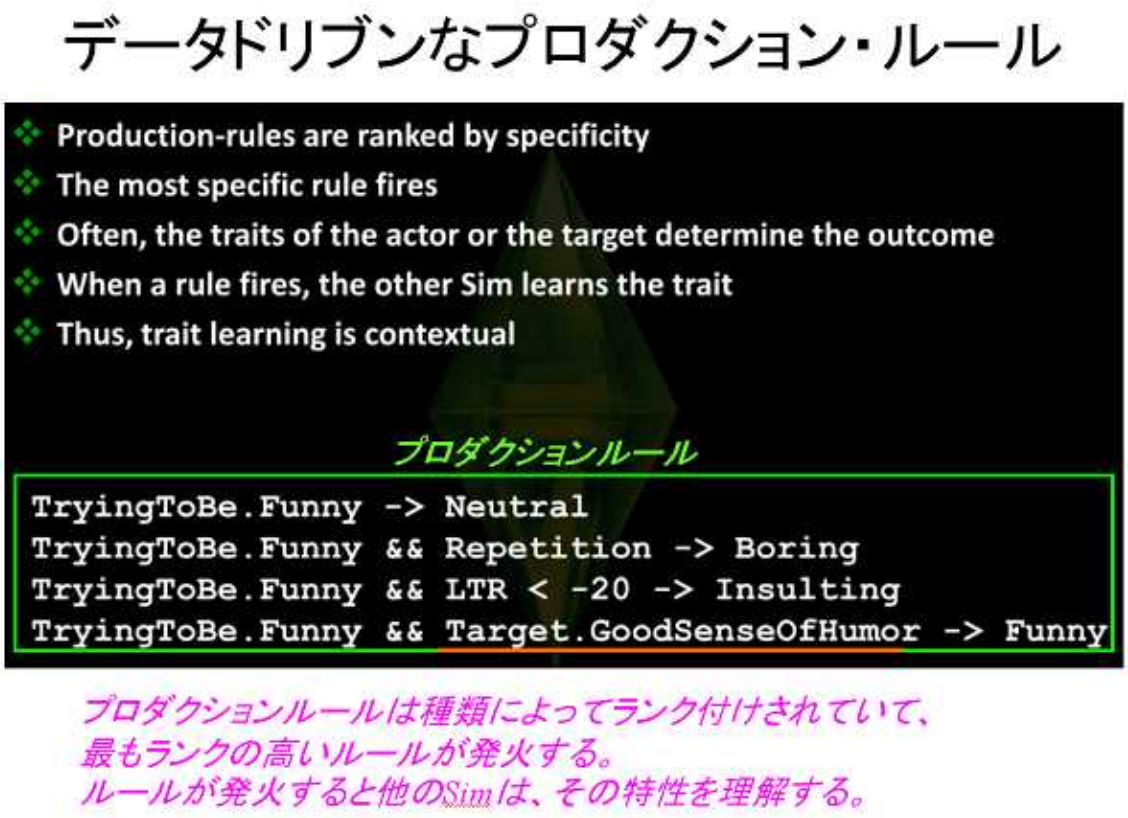


図　プロダション・ルールの全体の仕組み[2]

街全体を制御する AI

The Sims 3 では、各 AI とは独立に、街全体を統制する AI も実装されており、NPC を増やしたり消したりし、雇用をしたり職を奪ったりしながら、雇用率や男女比をコントロールしている。このコントロールは、街全体で様々な人間が点在し、様々なドラマが創発されるような状況を作り出すためである。

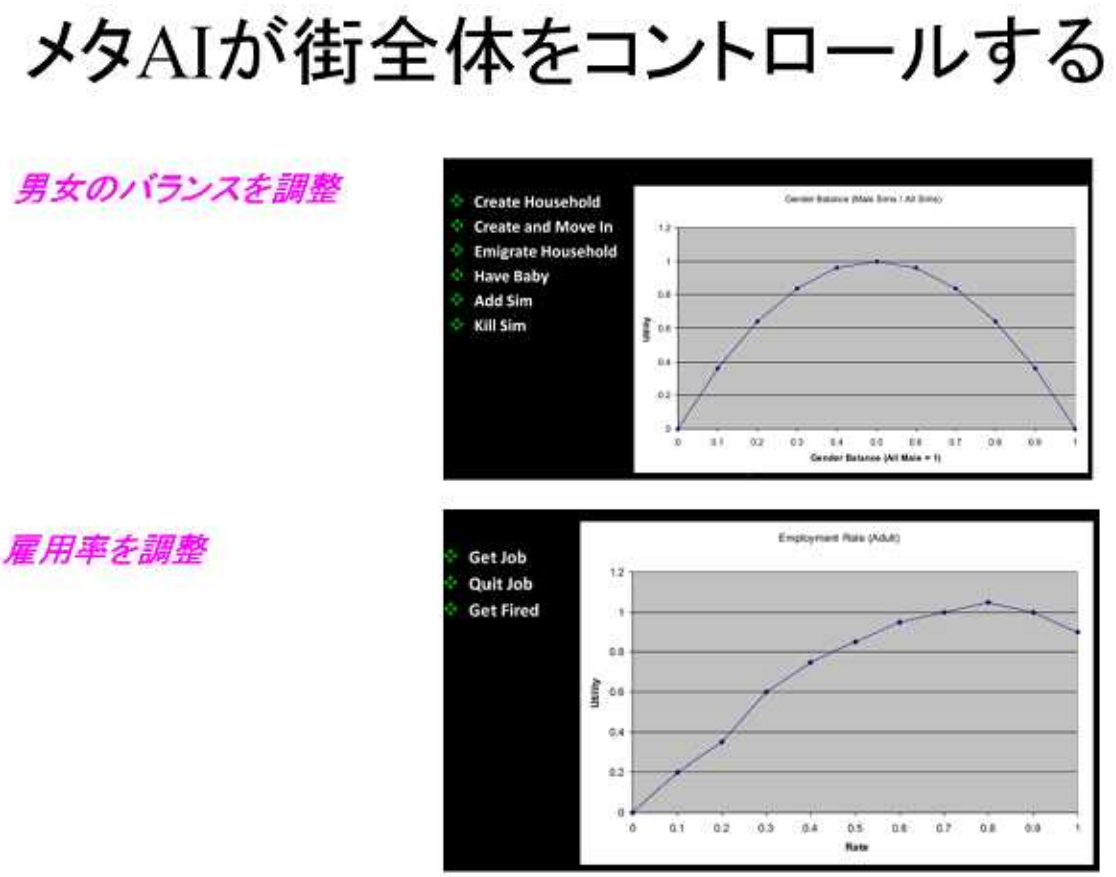


図　メタ AI が街全体の状況をコントロールする[2]

FPS(戦闘) ⇔ 日常系の AI の比較

以上が、Killzone 2 と The Sims 3 の AI の解説である。FPS と日常系 AI という対照的な AI を解説したが、ここでは幾つかのファクターに分けて２つの AI 技術の相違をより明確に対比したい。

(1) AI の内面構造

まず AI の内面構造であるが、FPS の AI は思考が複雑であっても、内面構造自体は非常にシンプルである。戦場という極限状況において、感情や周囲との関係性は極めて単純な形に還元されている。「敵を倒す」「仲間を助ける」「自己保全」という３つの動機にほぼ集約できると言ってよい。また、構造物も、敵に射線が通るか、敵の視点から隠れているか、という障害物としての機能に還元されているのである。

一方、日常系の AI の内面は、複数の内面パラメーターから構成され、その競合の中から行動が決定される。合理性よりは人間性のエミュレーションを目標としており、環境との多様な関係の中で自己の行為を決定して行く。周囲の物、一つ一つが多様な意味と可能な行為の集合を保持しており、またそれが内面パラメーターと深く結びつくことで、AI と物、また AI と AI 同士の多様なインタラクションを産み出すのである。

FPS(戦闘) ⇔ 日常系のAIの比較

	FPS	日常系
内面	シンプル	複雑
思考	複雑	シンプル
周囲の環境の見え方	視線・射線に対する障壁	多様な行為のアフォーダンス
産み出したいもの	戦場	人間ドラマ
組織・人間関係	階層的・厳密	平坦的・緩い繋がり
スコープ時間	短期～長期	短期～長期
重要なもの	敵味方の生死	自分の幸福度
インタラクション	暴力. 治癒	多様
学術との関係	アルゴリズム	哲学、認知科学、心理学など

キャラクターAIの技術はこの両極の間にある

図 FPS（戦闘）と日常系ゲームにいおいて必要とされる AI 技術の比較[3]

(2) 思考

次に思考であるが、FPS における AI の思考は極めて高度である。動機が敵の殲滅と自己の保持にあるとはいえ、その分、生死をかけた繊細さで、状況を極めて緻密に解析しつ

つ行動を瞬時に決定する必要がある。敵勢力、味方勢力、自分の状態、敵の状態、残り弾薬数、地形の状態、など、様々なファクターが、生死の上に極めてセンシティブな意味と効果を持つのが戦場である。こういった状況を全て把握、そして未知の情報は推測した上で、意思決定を行うのである。

一方、日常系の AI にとって、世界はどこまで行っても日常的な空間であり、状況というものには、それほどセンシティブではない（特に The Sims の街は、少々の無礼が許される平和的な街である）。道路や公園、周囲の人、人の家、自分の家、と大まかな状況しかない。どちらかと言えば、自分の内面的なパラメーターの力学によって行為が選択されており、意思決定の原理(Motive Engine)自体は極めて単純である。

(3) 集団としての AI

また集合としての AI を考えてみる。Killzone 2 では、戦場の組織として極めて厳密な階層的な構造が形成されていた。また、当然のことであるが、FPS では戦場の雰囲気を再現するように AI が動作させられている。

一方、The Sims では、大勢の人が緩やかで多様な関係で結ばれている。The Sims では、人間ドラマを産み出すようなシステムが根底に敷くことを目的としているのである。

(4) スコープ時間

スコープ時間とはここでは、AI が活動の単位とする時間の目安である。或いは、ユーザーから見て首尾一貫した知性的な振る舞いを維持できる時間である。

FPS は初期のオンライン専用ゲームであった頃には、部屋に固定された AI がプレイヤーがステージを通り過ぎる極めて短時間の思考が出来ればよかった。しかし、ステージが拡大し、オンラインにＡＩが参戦するような現代では、数分から十数分といった極めて継続的な時間を知的に活動をし続ける必要が発生したのである。

一方、日常系の AI も、かつての、簡単な Motive Engine によって周囲の人、オブジェクトとインタラクションを重ねていたところから、より長期的な目標や夢を持って日々を組み立てるように実装されており、長期的に見ても人間らしい活動を行える方向が目指されている。

(5) 学術分野との関係

また他の学術分野との関係を考えたとき、FPS の AI は、意思決定の問題を極めてアルゴリスティックに解決するために、人工知能、及び制御工学の伝統的な分野（プランニング、有限状態機械、プロダクション・ルールなど）との結びつきが強い。

一方、日常系の AI は、どちらかと言えば、人間の社会性・心理を穿ったシステムの構築を目指すために、心理学・認知科学的な分野との結びつきや引用が多いのが特徴である。

このように、FPS と日常系の AI は極めて対照的な特徴を持っており、重要なことは、この二極のゲーム AI の間には両極の特徴を兼ね備えた多様な AI のフィールドがあり、そこには、まだ、様々なゲーム AI の可能性、ひいてはゲームデザインの可能性が眠っているということである。こういった可能性を一つ一つ取り出し、発展させ、ゲーム AI の広大な領野を耕し、ゲーム AI のポテンシャルを上げ、知識と方法を蓄積して行くことが、今後１０年のゲーム AI の発展に必要な仕事である。

図　ゲーム AI マップ　ゲーム AI は広大で多様な分野であるが、FPS の AI と日常系 AI は、このようなフィールドの良い指標となるだろう

AI における階層構造

「Killzone 2」「The Sims」に共通する特徴はその階層型構造にあった。ここでは、なぜゲーム AI にとって階層構造が大切なのかを解説する。

人間の知性は様々な階層的な世界の捉え方を無意識に行っている。その仲で、主な二つの階層化は「空間階層化」と「時間階層化」である。他の階層化は、この二つのメタファーとして産み出されると言ってよい。

空間階層化は、空間を様々なスケールで切り取ることで、自分の周囲だけの空間、街全体の空間、地域、国、宇宙、或いは翻って蟻の巣穴のスケールなど、思考しやすい単位の空間を切り取って、それぞれの段階で思考する能力を持っている。かつ、それらをスケールに沿って階層的に捉える能力を持っている。広大な仮想世界で活動する AI にも、これらの能力は必須である。

時間階層化は、瞬時の判断から、中期、長期、一生に渡るようなプランまで、活動する

べき時間を様々なスケールで切り取って、各段階で思考している。長時間の活動（数分～十数分）が必要とされて来たゲーム AI にとってもこの能力は必要である。そして、こういった階層化構造を導入することで、AI は、一つレベルの高い AI に発展する土台を得るのである。階層型 AI が導入されたゲームの例は多く、また最近では、一つの設計方針として頭に入れておくべきものである。

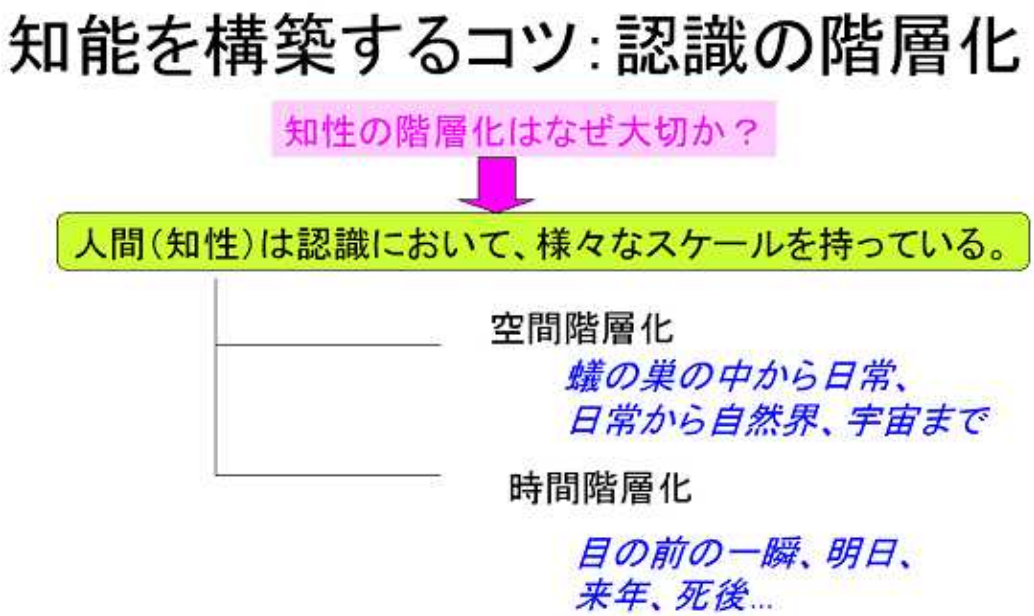


図　知性の二つの階層構造[3]　空間階層化、時間階層化は、ゲーム AI の設計の基本ポリシーと言ってよいだろう。広大なマップを自由に動き回る AI とは、長時間・長距離を支配すると同時に、局所的戦闘、瞬時の判断を行う能力を持っていなければならない。そこで、瞬時の判断から長期行動プランまでを、階層的な思考として持っておくこと、或いは空間の把握においても戦略的に地形全体を捉えるところから、数十 m 四方の局所戦闘思考まで、多様な階層レベルの思考を実現することが重要になって来るのである。

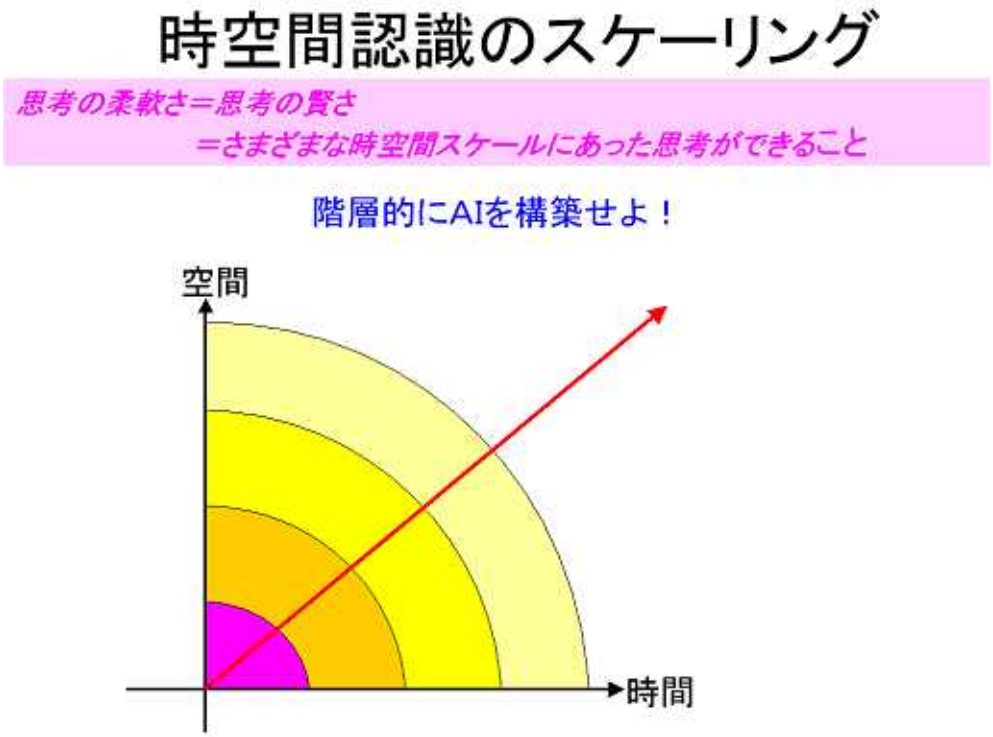


図　時空間認識のスケーリング[3]　人間は無意識のうちに世界を階層的に捉えている。人間の場合、その階層はより精緻で複雑なものであろうが、AI の思考には、時間、空間に渡る明確な階層性を導入することで、柔軟な意思決定機構を築くことが出来る。

階層型思考を持つ人間から、階層型思考を持たない AI を見た場合、それはちょうど上から下を見下ろすように、拙い存在に見える。例えば、短期的な思考しか持てない AI には、長期的戦略によって勝つことが出来るし、局所戦闘しか頭にない AI には、戦略的な罠や、退路を断って攻略すればよい。つまり、自分の作成している AI が何か足りないと思ったら、まず階層構造で足りていないかを検証するべきである（詳細は以下の文献[19]にまとめているので参照されたい）。

００年代から１０年代のゲーム AI へ

００年代のゲーム AI

００年代は、ゲーム AI がゲームエンジンから独立した一つのシステムとなろうとした時期であり、独立したシステムとして発展すると同時に、ゲームエンジン本体との関係を模索した時期でもあった。

欧米ではＦＰＳを通じて学術から AI 技術が流入し、ゲーム AI 思考を高度に発展させ、

パス検索用のデータは大規模化・階層化され、それらをまとめるフレームとしてエージェント・アーキテクチャが導入された。つまり、００年代という年代はゲーム AI がひとり立ちを始めると同時に、さまざまに必要になりそうな技術が試されながら、だんだんと荒削りながらも、ゲーム AI という分野の全体像が浮かび上がって来た時期である。

この主要な項目を挙げると、以下のようになるだろう。

- エージェント・アーキテクチャ
- パスプランニング
- パスデータ階層化
- 階層型思考
- アニメーションと意思決定
- プランニング

これらは、GDC や学会で議論され、数多くの英語の文献として実例やアイデアとして蓄積されている[20]。日本は、だいたいにおいてこの流れの外にあったと言ってよい。その理由は複数あるが、

- 第一は日本のゲーム産業が開発情報を極めてクローズドにして来たこと
- 第二はそういった技術を発展する英語圏のコミュニケーションが出来なかったこと
- 第三に日本のゲームデザインと欧米のゲームデザインに隔たりがあり、自然、求める AI の方向が違ったこと
- 第四に、方向が違ったなら自分たちで築けば良かったが、そういった技術的積み上げの基盤がなかったこと
- 第五に、ゲームデザインが必ずゲーム開発技術に優先するべきという思い込みがあったこと（現在でも）

である。日本が技術的にも、ゲームデザインとしても、市場規模の拡大としても行き詰っている間に、欧米のゲーム AI 分野は急速にその土台を築いて行った。

００年代のゲーム AI 最大の成果

さて、こういった００年代の流れの中で、欧米の各ゲーム AI 開発者、各ゲーム開発企業は、ゲーム AI の開発の力点とも言うべきものを、それぞれ見出し始めた。以前は、見当はずれな、或いは、工程ばかりが増えて行くポイントに力を入れていたことも多かった。そして、ごてごてとしたメンテナンスのしにくい AI システムが出来上がって行った。しかし、そう言った、さまざまな失敗例と、幾つかの成功例を研究し、AI の開発の力点を何処に集中するべきかを学習した。そして欧米では、そういったノウハウが、各種ゲームカンファ

レンスや学会を通して、共通に認識する技術やノウハウとして行った。さらに共通の認識とすることで、同じ言葉、同じ技術を、ゲーム産業全体で研究し応用し議論し合い発展させるという体制を作って行った。このことは、00 年代における欧米ゲーム産業の AI 分野最大の成果と言ってよい。

「パスシステムを設計し質の高いパスデータを階層化して築いておくこと」「エージェント・アーキテクチャのフレームの中で、センサー、意思決定、身体制御、意思決定部分に導入する技術を検討し」「各部分の関係性を検討すること」、こういった、ゲーム AI 製作の基本姿勢も、この１０年を通して、次第に明確になって来たことであった。

１０年代のゲーム AI へ

そして、今年の GDC2010 では、ゲーム AI のこの１０年の発展から次の１０年の発展へ向けての発展の兆しが垣間見られた。静的なパスシステム（オブジェクトがマップ上に落ちても対応できない）から、次世代のパスシステムとして動的オパスシステム（動的なオブジェクトに対応するパスシステム。例えば落下したオブジェクトに合わせてメッシュが再分割されて変化する）への移行が複数のタイトルで実現されていた(「Sprinter Cell: Conviction」(Ubisoft,2010)、各種 AI ミドルウェア)。思考部分では、プランニング技術がほぼ基礎技術として、複数のタイトルに導入されていた(「Killzone 2」「The Sims」など)。

おそらく、これからの１０年は、各種プラナー（プランニング・アルゴリズム）の開発が、AI 思考技術の中核となると予想される。Killzone 2 の AI が 1992 年の学術論文を基礎にしたように、30 年以上に及ぶ学術におけるプラナーの研究成果がゲーム AI に応用されて行くと考えられる。プラナーはちょうど CG 分野の「シェーダ」と同じように、AI 分野の共通フレームとしての役割をもって AI 技術の標準的な技術として定着して行くと予想される。

このように１０年代は手探りで探り当てたゲーム AI 技術分野の全体の領野が、一つ一つ本格的に探求され、成果を産む時代がこれからの 10 年である。その発展には学術と技術双方のバックグランドを持つ優秀な人材が必要であり、既に専門化しつつあるゲーム AI 分野は、欧米では、そういった人々を数多く引き寄せ、さらなる発展を遂げるだろう。しかし、残念ながら、まだ日本にはこの基盤が出来てない。

日本のゲーム AI の未来のために

００年代は、ゲーム AI 発展の流れの完全な外にあった日本である。しかし、これまでのゲーム AI の各分野の探求の荒削りの結果が、膨大な英語の文献や論文に記されている（そういったゲーム AI のこれまでの歴史や参考文献は、これまでの報告書に記して来た[6][17]）。そして、それらは、十分に探求されずに放置されたゲーム AI 技術の萌芽である。幾

つかの芽は、欧米の FPS やシミュレーションゲームで大きな成長を遂げた。しかし、幾つかの芽は、日本のゲームデザインの土壌でこそ大きく発展を遂げるものであろう。

　こういった文献を一つ一つ読み解き、各技術事項の持つ可能性を十分に吟味し、我々が持つ独自で多様なゲームデザインの上に新しく応用の展開し応を見出すとき、日本はこの１０年の遅れを取り戻し、さらに、そこから新しいゲーム AI の発展の形を築くことができる。そして、欧米で育まれたゲーム AI の流れと、我々が築くゲーム AI 技術の流れが合流を果たすことで、全体として新しいゲーム AI の広がりを産みだし、世界のゲーム AI の発展の流れに大きく貢献することになるだろう。

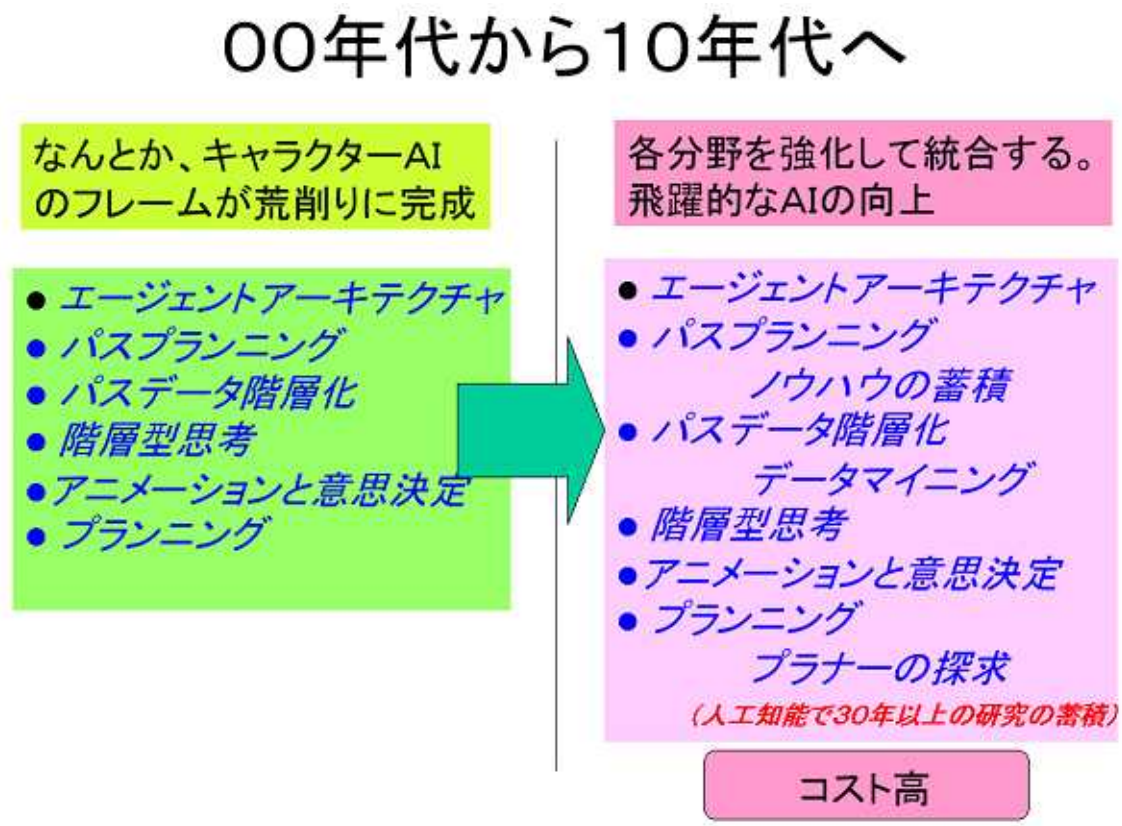


図　ゲーム AI の００年代から１０年代への発展・変化[3]

参考文献

[1] Alex Champandard, Tim Verweij, Remco Straatman, Killzone 2 Multiplayer Bots, Game AI Conference, Paris, 2009 (GDC2010でも同様の資料が用いられた)
http://files.aigamedev.com/coverage/GAIC09_Killzone2Bots_StraatmanChampandard.pdf

[2] Richard Evans, Modeling Individual Personalities in The Sims 3, GDC2010, 2010
http://cmpmedia.vo.llnwd.net/o1/vault/gdc10/slides/Evans_Richard_ModelingIndividualPersonalitiesInTheSims3.pdf

[3] 三宅陽一郎, "「00年代から10年代へ ゲームＡＩとプロシージャル手法の進化」", GDC2010報告会(IGDA日本) , 2010
http://igda.sakura.ne.jp/sblo_files/ai-igdajp/GDC2010/GDC_Report_2010_YMiyake_0403.pdf

[4] "AI Summit '10: Slides, Notes, Highlights and Photos", AIGameDev, 2010
http://aigamedev.com/open/coverage/gdc10-slides-highlights/

[5] 三宅　陽一郎, 「GDCにおける海外のゲーム関連技術についての調査」, 財団法人デジタルコンテンツ協会「デジタルコンテンツ制作の先端技術応用に関する調査研究報告書」(2008年度), pp.331-369, 2009 http://www.dcaj.org/report/2008/data/dc_08_03.pdf

[6] 三宅　陽一郎, 「ゲームAI分野」, 財団法人デジタルコンテンツ協会「デジタルコンテンツ制作の先端技術応用に関する調査研究報告書」(2007年度), pp.37-113, 2008
http://www.dcaj.org/report/2007/data/dc08_07.pdf

[7] スチュワート ラッセル、ピーター ノーヴィグ, 「エージェントアプローチ 人工知能」, 共立出版, 1997

[8] 生天目 章, 「マルチエージェントと複雑系」, 森北出版, 1998

[9] 三宅陽一郎, IGDA日本 ゲームAI連続セミナー第2回「F.E.A.R におけるゴール指向型アクションプランニング」資料, 2007 http://blogai.igda.jp/article/33936286.html

[10] 三宅陽一郎, IGDA日本 ゲームAI連続セミナー第3回「Chrome Hounds におけるチームＡＩ」資料, 2007 http://blogai.igda.jp/article/33936286.html

[11] Nau, D. et al., "SHOP2: An HTN Planning System", Journal of Artificial Intelligence Research, vol.20 pp.379-404, 2003
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.72.188&rep=rep1&type=pdf

[12] Dana S. Nau, "Hierarchical Task Network Planning, Lecture slides for Automated Planning: Theory and Practice",
http://www.cs.umd.edu/~nau/cmsc722/notes/chapter11.pdf

[13] Damian Isla, "Building a Better Battle: HALO 3 AI Objectives", GDC2008 , 2008
http://www.bungie.net/Inside/publications.aspx

[14] Dave C. Pottinger, "Terrain Analysis in Realtime Strategy Games", http://zeniroy.springnote.com/pages/481669/attachments/212469
[15] 多摩 豊, "ウィル・ライトが明かすシムシティーのすべて", 角川書店 , 1990
[16] Maurice Bergsma and Peter Meij, "Influence Maps", Student Lecture GAI , 2006 http://www.slideshare.net/mobius.cn/influence-map
[17] 三宅　陽一郎, 「プログラミングAI」, 財団法人デジタルコンテンツ協会「デジタルコンテンツ制作の先端技術応用に関する調査研究報告書」(2008年度), pp.73-137, 2009 http://www.dcaj.org/report/2008/data/dc_08_03.pdf
[18] Ken Forbus, "Simulation and Modeling: Under the hood of The Sims", http://www.cs.northwestern.edu/%7Eforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm
[19] 三宅陽一郎, "ゲームＡＩにおける４つの階層", y_miyake のゲームAI千夜一夜 , 2009 http://blogai.igda.jp/article/32493117.html
[20] 三宅陽一郎, IGDA日本 ゲームAI連続セミナー資料集, http://blogai.igda.jp/article/33936286.html