



「レスキューシミュレーションリーグ」

第5回

■ 鳥海不二夫 (名古屋大学)

ロボカップレスキューシミュレーションの概要

ロボカップレスキュー⁶⁾の歴史は2000年にメルボルンで開催されたRoboCup2000に始まる。このときは、デモンストレーション展示であったが、その翌年の2001年からロボカップの正式な競技・課題として取り上げられるようになった。

ロボカップサッカーはチェスに代わる人工知能分野のランドマークの1つとして紹介され、ロボット工学分野、人工知能分野の競争的研究開発が行われる国際的なプロジェクトであるが、ロボカップレスキューはこの成果を災害救助分野に応用することを目指すプロジェクトである。

ロボカップレスキューには大きく分けて、実機リーグとシミュレーションリーグが存在する。実機リーグは、倒壊した建物などにいる被災者の発見救助を目的としたロボットの性能を競っているのに対し、シミュレーションリーグでは被災した都市全体を対象として、効率的な救助戦略を実現することを目的としている。すなわち、実機リーグはマイクロな救助現場を扱い、シミュレーションリーグはマクロな救助現場を扱っていると言える。ロボカップサッカーとは異なり、実機とシミュレーションで扱うフィールドが異なっているが、最終的には、シミュレーションリーグで発見された救助戦略に従って、実機リーグで開発されたロボットが救助戦略を行うことを目指している。

ここでは、シミュレーションリーグについて解説を行う。特に、ロボカップレスキューシミュレーションに興味を持ち、新しく参加しようと考えている研究者に向けた情報を中心に解説する。

ロボカップレスキューシミュレーションリーグのルール

シミュレーションの目的

ロボカップレスキューシミュレーションリーグの目的は、効率的な災害救助戦略の実現にあるが、

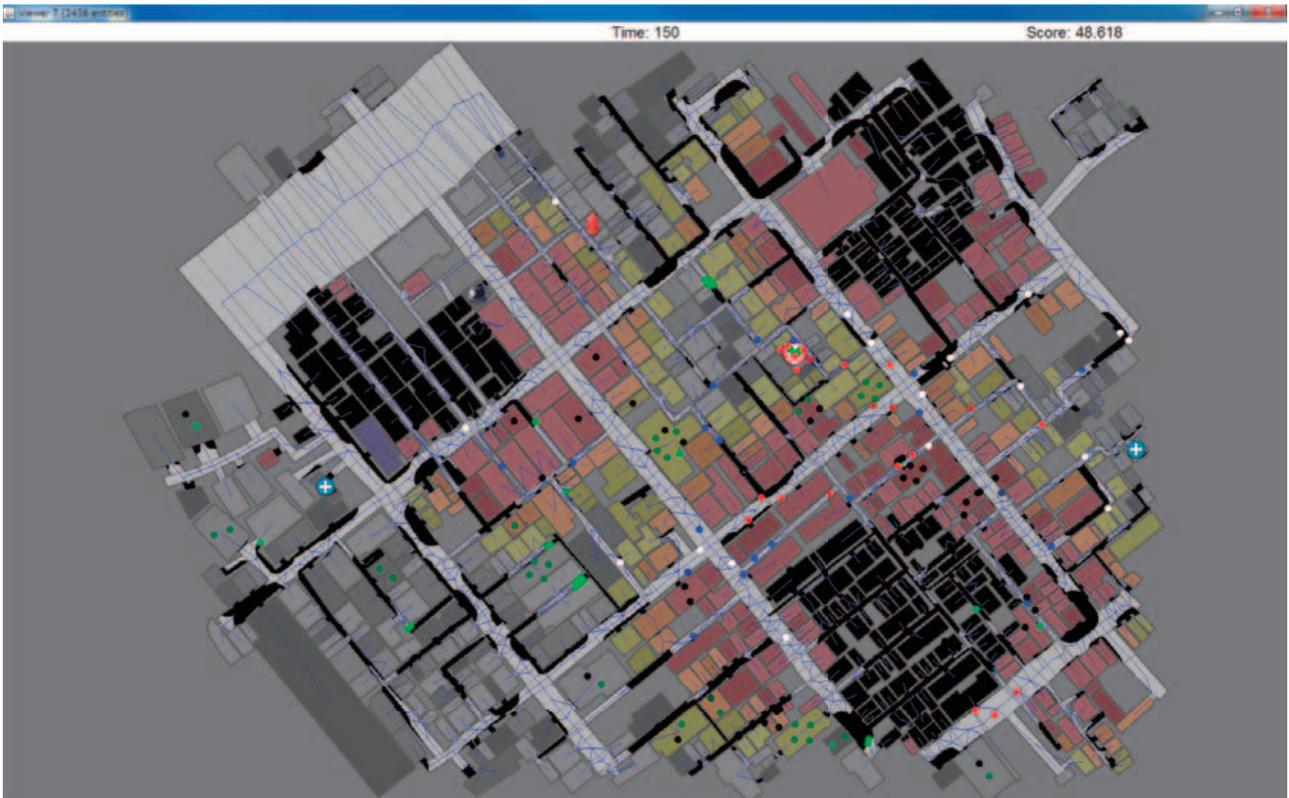


図-1 ロボカップレスキューシミュレータ(RCRS)の実行画面

個々のシミュレーションの目的は、災害現場における救助隊の行動シミュレーションおよび、それによる被害の軽減である。

ロボカップレスキューシミュレーションリーグでは、ロボカップレスキューシミュレータ (RoboCup Rescue Simulator: 以下 RCRS) によって仮想的な災害空間 (図-1) が構築される。災害空間には、建物や道路、市民などの要素からなる都市が形成されており、その都市に大地震が発生したことを想定し、火災シミュレータ¹⁾、建物倒壊シミュレータ、交通シミュレータ³⁾などを組み合わせ現実を模した災害現場を再現している。

このようにして作られた災害空間内に、参加チームがそれぞれに開発した救助エージェントを導入することで、災害空間内の被害を最小限に抑えることを目指す。災害空間内の被害は大きく、

- 建物への被害
- 市民への被害

に分けることができる。したがって、シミュレーションの目的は、火災や倒壊による建物への被害の軽

減と、被災者の救出となる。

参加チームは、複数個所から発生する火災のうちどれを優先的に消火するべきかなど、効率的な救助を実現するための戦略を設計する必要がある。被災者が取り残されている建物の近くで発生した火災を優先的に消火するか、大規模な火災に発展しそうな火災を優先するかなど、その戦略は無数に考えられる。それらの戦略を取捨選択しながら、建物への被害を減らし、多くの市民を避難所へ誘導することがシミュレーションの目的である。

シミュレータのシステム構成

RCRS のシステム構成は図-2 のようになっている。RCRS のシステムは主に、

1. RCRS Server
2. RCRS Client
3. Viewer

からなる。

まず、RCRS Server は Kernel と呼ばれるサーバ本体のほかに、Traffic Simulator, Fire Simulator

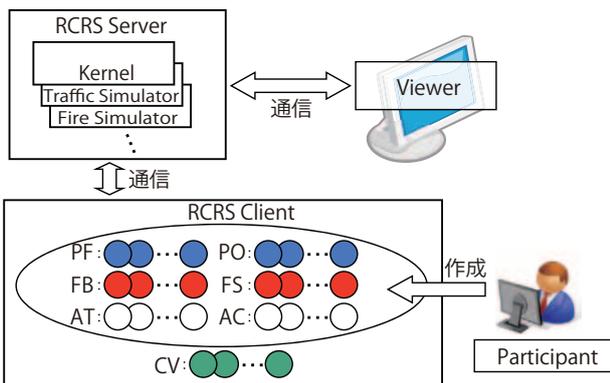


図-2 RCRS のシステム構成

などをモジュールとして搭載している。ここで、Kernel とシミュレータを分離することによって、より精度の良い各種シミュレータが開発された際には、それぞれのシミュレータを入れ替えることが可能となっている。

次に、参加者が作成するエージェントを管理する RCRS Client がある。クライアントでは、救助エージェントのほかに市民エージェントの制御も行う。なお、クライアントとサーバ間は TCP/IP 通信によって結ばれている。そのため、大規模なシミュレーションを行う場合は異なるマシンで、クライアントとサーバを動かすことが望ましい。

また、救助活動の結果は Viewer によって出力される。Viewer も TCP/IP 通信によって出力結果を得ているため、独立したマシン上で利用することが可能である。たとえば、災害現場を立体的に確認することができる 3D ビューワのような動作が重い Viewer は、異なるマシン上で動かすことが望ましい。

ルール概要

参加チームは、火災の消火を行う消防エージェント (Fire Brigade Agent : FB)、倒壊した建物から市民を救出する救急エージェント (Ambulance Team Agent : AT)、道路に散乱するガレキを撤去する啓開エージェント (Police Force Agent : PF) からなる 3 種類のエージェントと、それぞれのエージェントを管理する 3 種類の救助センター、すなわち消防セン

ター (Fire Station : FS)、救急センター (Ambulance Center : AC)、啓開センター (Police Office : PO) の行動を設計し、災害被害を食い止めることを目指す。なお、エージェントの詳細については後述する。

各エージェントはサーバから送られてくる情報のみから行うべき行動を判断し決定しなければならない。サーバから送られてくる情報には以下のようなものがある。

- 視覚情報
- 通信情報
- エージェント自身の情報

視覚情報は、エージェントが見ることのできる範囲での災害情報である。たとえば、周囲に火災が発生していれば視覚情報から火災を感知することが可能である。その際には、火災の規模なども判断できるため、災害空間の被害状況を知るためには視覚情報は重要である。一方、ある程度遠くで発生している火災については感知することはできないため、遠くにある建物の情報を知りたい場合は、そこに近づかなければならない。また、倒壊した建物に埋もれている市民などは道路から発見することができないこともある。そのような場合は、市民を発見するために実際に倒壊した建物に入り探索を行う必要がある。

通信情報は、エージェント間の情報共有に利用されるもので、大きく分けて音声通信と無線通信がある。音声通信は、エージェントが発している「声」による通信である。たとえば、倒壊した建物の中にエージェントがいれば、その近くの道路上にいるエージェントには被災者の救助を求める声が聞こえる可能性がある。ただし、具体的な声の出所などの情報は得られないため、視覚情報に比べると制限された情報といえる。また、救助エージェント同士が近くにいる場合は、音声通信による情報交換を行うことも可能である。

一方、無線通信は救助エージェントと救助センター、および救助エージェント間で行われる。これによって、各救助エージェントが得た情報をすべてのエージェントで共有することが可能になる。ただし、通信には制限がいくつか加えられており、災害空間

のすべての情報を完全に共有することは難しい。そのため、どのような情報を共有するかを検討することもシミュレーション上では大きな課題となる。なお、通信の詳細については後述する。

最後に、エージェント自身の情報は、エージェントの存在位置や体力などである。また、シミュレーション開始当初に倒壊した建物などに埋もれてしまっているエージェントは、どの程度深い場所に埋もれているかの情報なども取得することができる。さらに、火災の近くにいる場合などに当該ステップに受けるダメージ量などの情報も取得可能である。ただし、エージェントの体力やダメージの情報は必ずしも正確ではないため、ある程度の誤差を考慮に入れた上で行動を判断する必要がある。

エージェントの行動は1ステップ数秒～数十秒に対応するステップ単位で行われ、各ステップごとにエージェントは1つの行動を行うことができる。たとえば、火災現場への移動であったり、消火活動、被災者の救出活動、あるいは避難所での休憩などである。シミュレーションは100～1000ステップの間行われ、最終的に被害を受けた建物、および死亡した市民の数などから計算されるスコアによって救助活動が評価される。

災害救助の評価

指定期間のシミュレーションが終了した後、災害救助の評価が行われる。評価に利用される指標を**表-1**に示す。評価はこれらの指標に重みを付けたスコアによって決定される。

なお、余談ではあるが2008年に日本で開かれた春期競技会では、移動距離への重みが高すぎたため、サンプルとして作られた一切行動を行わない「チームエコ」が優勝に匹敵するスコアを出すという状況を生み出した^{☆1}。より現実に即したスコアリングの確立は、今後のロボカップレスキューにおける大きな課題の1つである。

評価対象	評価内容
救出市民数	救助に成功した市民の数
建物倒壊度	建物へのダメージ
消火速度	消火を行う速さ。消火を開始してから消火されるまでの時間
消火効率	消火の効率性。建物が燃えだしてから消火されるまでの時間
移動距離	救助エージェントの移動距離。距離が短いほど効率的な救助と判断される
負傷率	市民エージェントの負傷者の数、および負傷の度合い
被災者発見速度	被災者を発見するまでの時間
啓開効率	道路上の障害物の撤去時間

表-1 評価対象

災害空間の概要

ロボカップレスキューシミュレーションにおける災害空間 (Map) は、阪神大震災級の災害が発生した際の一都市を表現している。都市は、

1. 建物
2. 道路
3. 特別な建物

の3種類のオブジェクトからなる。

建物

建物などのMap内のオブジェクトは、すべてIDによって管理され基本データとして地図上の輪郭を表す座標を持つ。建物には材質も指定されており、木、コンクリート、鉄などが存在する。ただし、RCRS Ver1.0では材質の違いによる影響は考慮されず、材質は同一として扱われる^{☆2}。また、建物ごとに階数も存在する。ただし、現在階数は特に利用はされていない。したがって、現在のところ建物を定義するパラメータは主にその輪郭となる。なお、補助的なパラメータとして床の面積なども情報として与えられる。

一部の建物は地震の直後（シミュレーション開始時点）で火災が発生していることがある。また、火災が発生し炎上している建物に隣接する建物は、隣接度合いに応じて延焼が発生する。

☆1 現在のルールでは修正されている。

☆2 建物の材質や階数などのデータは、個人情報保護などの観点から扱いが難しいため、シミュレーションへの反映の優先順位は低い。



これらの火災の規模や被害の度合いも建物のパラメータとして表される。火災に関連するパラメータには以下のようなものがある。

- 温度
- 燃焼度
- 倒壊度

まず、どの程度建物が燃えているかは温度パラメータによって表される。すなわち、温度が高いほど強く燃えていることを示す。高い温度で燃えている建物は、より強いダメージを受ける可能性があるとともに、近隣の建物に飛び火する可能性も高くなる。

また、建物がどの程度燃えてしまったかは、燃焼度によって表現される。燃焼度は温度が高ければ時間とともに急速に増加し、最終的には燃え尽きるまで進行する。

倒壊度は、シミュレーション開始時に設定される。倒壊した建物の内部にいるエージェントはガレキに埋もれて身動きがとれなくなってしまう。そのため、倒壊した建物から市民エージェントを救い出すことも、救助エージェントにとっては重要なミッションとなる。

道路

道路は、建物と同様に ID および輪郭によって定義されている。また、どの道路とどの道路がつながっているかも定義されており、つながっていない道路間を移動することはできない。また、輪郭によって道の広さが定義され、細い道にたくさんのエージェントが存在する場合、渋滞を起こす可能性がある。したがって、エージェントを設計する場合は渋滞が発生しないような行動を心がける必要がある。

また、道路には建物などの倒壊によりガレキが散乱している可能性がある。そのようなガレキは閉塞と呼ばれ、閉塞が存在する道路は通行することができない。

閉塞は、建物などと同様に輪郭を持ち通行を阻害する。閉塞は啓開エージェントによって撤去することが可能であるが、撤去するためには大きさ（閉塞の面積）に応じた時間が必要となる。そのため、ど

の閉塞を優先的に撤去するかが迅速な災害救助を行うために重要な戦略となる。

特別な建物

災害空間内にある建物の中には、以下に示す特別な機能を持ったものが存在する。

1. 避難所
2. 救急隊センター
3. 消防隊センター
4. 啓開隊センター

避難所は、市民が避難する目的地である。自力で移動可能な市民エージェントは避難所を目指して移動を行い、自力で移動できない市民エージェントは救急エージェントによって避難所に搬送される。避難所にいる市民エージェントはそれ以上ダメージを受けず、安全が保証される。また、消防エージェントは避難所で給水することができる。消火活動には水が必要であるため、消防隊が連続して行える消火活動には制限がある。水がなくなった時点で避難所に移動して給水する必要がある。なお、現実での消火活動では消火栓や貯水水槽を利用するが、現実の都市におけるこれらの位置情報などを得ることが困難であり、災害空間上で現実に即したデータを作成できないため実装されていない。そのため、現在のルールでは水量には限界があるものとして、現在のモデルを利用している。このような地図作成時における不足データの補間も今後の課題である。

救急隊センター、消防隊センター、啓開隊センターは前述した3つの救助センターの存在位置を示している。災害空間上では、そこにあるということを示す以外の意味は特にない。

これらの4種類の特別な建物は、2010年現在のルールでは延焼することがない。そのため、避難所へ一度避難した市民はシミュレーション終了まで確実に生存することが保証される。ただし、たとえ都市が大炎上して周りが焼け野原となっても特殊な建物は燃えないのは不自然であり、今後ルールが変更される可能性は高い。その場合、避難所やセンターを優先的に守るといった戦略をとる必要性が生じる

ため、各チームはより深い戦略を作成する必要があると出てくるであろう。

エージェントと通信

エージェント概要

ロボカップレスキューシミュレーションで扱われるエージェントは4種類存在する。これらのエージェントは大きく2つに分けることができる。1つは市民エージェント、もう1つは救助エージェントである。

市民エージェントは災害にあった被災者であり、救助の対象となるエージェントである。この市民エージェントをできるだけ多く助けることが本シミュレーションの目的の1つである。

一方、救助エージェントは災害救助を行うエージェントであり、前述したとおり、

- 消防エージェント (Fire Brigade Agent : FB)
- 救急エージェント (Ambulance Team Agent : AT)
- 啓開エージェント (Police Force Agent : PF)

の3種類のエージェントからなる。

市民エージェント

市民エージェントは競技運営側の用意した意思決定方法で行動し、競技者は制御できない。基本的には、市民エージェントは避難所を目指して移動を行う。しかしながら、道路の閉塞が存在し避難所まで移動できない場合などもあるため、その場合は啓開エージェントによる速やかなガレキの除去が必要となる。

また、倒壊した建物やガレキの中に市民が埋もれる可能性もある。その場合は、救急エージェントによる救出が必要である。

また、負傷している市民エージェントは自分で移動ができないため、救急エージェントによる搬送が必要となる。

消防エージェント

消防エージェントは、その名の通り火災が発生し

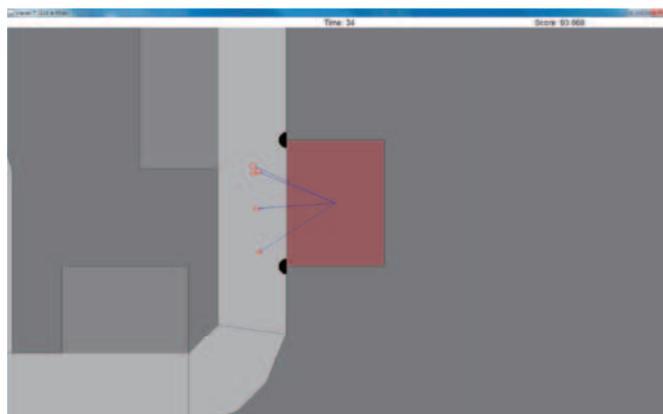


図-3 消防エージェントによる消火活動

ている建物の消火を行うエージェントである。消防エージェントは、火災に向けて放水することができ、一定以上の放水を行えば火災の温度を下げ消火することができる(図-3)。

ただし、放水には水が必要で、各消防エージェントは一定量の水しか保持することができない。したがって、一定量放水した後、避難所へ移動して給水する必要がある。

通常災害空間では複数個所で同時に火災が発生するため、どの火災現場にどの程度の消防エージェントを派遣するかが、消火活動の効果に大きく影響する。消防エージェントが足りなければ火災を止めることはできないし、1カ所に集中しすぎれば他の火災が大きくなる危険性がある。したがって、どの火災にどの程度の力を注いで消火を行うかを決定することが、戦略を構築する上で重要なポイントとなる。

救急エージェント

救急エージェントの役割は、倒壊した建物の中に取り残された市民エージェントを救出し、避難所まで搬送することである。

しかしながら、倒壊した建物にいる市民エージェントに関する情報は、シミュレーション開始時には存在しない。そのため、まずはさまざまな建物を訪れて、被災者がいないかどうかを確認する必要がある。被災者の発見自体は必ずしも救急エージェントが行う必要はないが、消防エージェントや啓開エージェントが市民エージェントを発見した場合は、セ



ンターを通して救急エージェントに連絡し、救助を行う必要がある。

倒壊した建物にいる市民エージェントは生き埋めにされているため、助け出すためには複数ステップかかる。このとき、大勢の救急エージェントが協力し合うことで、より早く市民エージェントを救出することも可能である。

生き埋めになっている市民エージェントは徐々に体力を奪われていき、最終的には死に至るため、負傷している市民エージェントは優先的に救出する必要がある。一方、生き埋めになっている建物が燃え始めた場合、市民エージェントは通常の数倍の速度で体力を奪われてしまうため、さらに迅速な救出活動が必要である。

よって、どのような順番で被災者を救助するかが救急エージェント作成のポイントとなる。体力が十分ある市民エージェントの救出は後回しにすることや、たとえ火災に巻き込まれている建物であっても、被災者を助けるために突入する自己犠牲的な行動など、さまざまな戦略が実装されている。

啓開エージェント

啓開エージェントの役目は、道路などを塞ぐ閉塞を撤去することである。

大規模な地震が発生した際には、建物の倒壊などにより道路がガレキに塞がれることがよくある。それによって生じた道路の閉塞により、火災現場や被災現場への到着が遅れ、被害が増大する可能性は高い。このため、迅速な閉塞の撤去は災害現場の救助活動の中でも重要なミッションの1つである。

啓開エージェントは、道路上の閉塞を撤去することができる。この際、閉塞の大きさによって撤去に必要な時間は異なるため、重要な道路に閉塞が発生している場合は複数の啓開エージェントによる迅速な撤去が必要な場合も多い。

啓開エージェントが優先的に考慮すべきものとして、

- 火災現場までの経路の確保
- 避難所までの経路の確保

• 閉塞に阻まれて身動きできない救助エージェントの開放
などがある。

ただし、一般に啓開行動は救助活動より前に行われる必要があるため、シミュレーションの後半は啓開エージェントは特にすべきことがなくなることが多い。そのため、シミュレーション後半には啓開エージェントによる被災者の探索を行わせていることが多い。したがって、効率的な被災者発見アルゴリズムの実装も啓開エージェント作成には重要な課題となる。

救助センター

救助センターは、エージェント間の通信を仲介する役割を担うエージェントである。1つの建物として表現されているため、移動などを行うことはできないが、各ステップごとに担当するエージェントから情報を収集し、何らかの処理を加えた上で情報を配信することができる。

救助センターの主な役割としては、

- エージェントから送られてくる災害現場情報の収集
- 災害現場情報の統合および情報処理
- 災害現場情報の配信
- 各エージェントへの指令

などがある。基本的にロボカップレスキューシミュレーションのエージェントは自律分散システムとして動くものであるが、センターを介すことで中央集権的に行動することが可能となる。

ただし、救助センターはすべての災害空間に必ず存在するという保証があるわけではないため、中央集権的なエージェント群であっても、センターレスの戦略を独自に持っている必要がある。

なお、前述したとおり救助センターには対応するエージェントごとに3種類存在する。

通信

通信は、エージェントおよびセンター間での情報のやりとりを行うために導入されている。通信には

「音声通信」と「無線通信」がある。

音声通信 音声通信は、人が発生する音声を表現しており、一定距離以内にいるエージェントに情報を伝える通信手段である。シミュレーションの条件によっては、無線通信は遮断される可能性があるが、音声通信はどのような場合でも実行可能である。

無線通信 無線通信は音声通信と異なり、距離に依存せずにエージェント間の通信を実現可能である。ただし、通信にはチャンネルが存在し、同じチャンネルを使っているエージェントと通信が可能である。

通信チャンネル数およびエージェントが受け取ることができるチャンネル数はシミュレーションの設定で決定される。通常は、チャンネル数が0～3、エージェントが受け取ることができるチャンネル数が0～1である。そのため、すべての通信をくまなく受け取ることにはできないことが多い。したがって、必要に応じてどのチャンネルをどのエージェントが使うのかななどを規定しておく必要がある。

また、無線通信では1ステップに1つのメッセージしか送ることができない。

音声通信、無線通信ともに送ることができる量は制限されている。これは、1ステップに対応する時間内に必ずしもすべての情報をお互いにやりとりすることはできないであろうという仮定のもと付加されている制限である。通信は通常自然言語ではなく、チームごとに任意にコード化されたバイナリ列によって行われる。その容量が、最大4MBに制限されている。そのため、少ない容量でいかに情報をコード化、圧縮し送るかが課題となる^{☆3}。

ロボカップレスキューシミュレータの導入

プロジェクト作成準備

本章では、RCRSの導入について、サーバのダ

^{☆3} これは実際の救助活動の場面に活かされる課題ではないため、ロボカップレスキューシミュレーションの本質とは外れているとの批判もある。

ウンロードから実行までの手順を説明する。ただし、JavaやEclipseの利用についてはある程度の知識があることを前提とする。また、本稿の説明ではWindowsを前提としているため、LinuxやMacintoshを利用している読者は、適宜読み替えていただきたい。

RCRSサーバは以下のURLからダウンロードすることが可能である。

<http://roborescue.sourceforge.net/>

RCRSサーバは日々更新されているため、最新版が必ずしもうまく稼働するとは限らない。そこで、ここでは2011年1月現在稼働する最新のサーバを例として取り上げる。当該ページのSourceforgeとあるリンクから、ダウンロードページに行くことができる。ここで、ダウンロードではなく「ViewAll」から2010を選択すると、

`simulator-package.tgz`

があるので、これをダウンロードする^{☆4}。

ダウンロードしたファイルは任意のフォルダで展開する。解凍後はrescue-competition2010というフォルダ以下にサーバが展開される。

また、RCRSを実行するためには、JDKとAntが必要である。JDK、Antともに最新版をインストールことを推奨する^{☆5}。

プロジェクトの作成

RCRSはVer1.0からJavaで作られているため、開発にはEclipseを使うことが望ましい。

まずは、Eclipse上で適当なプロジェクトを作成する。ここでは、RobocupRescueという名前のプロジェクトを作成する。

そこに、

`rescue-competition2010/jars`

`rescue-competition2010/lib`

をコピーする。ちなみに中身は、

- jars 以下：RCRSのjars

^{☆4} 今後ダウンロードすべきファイルは変更される可能性が高い。後述するRoboCup SimJPで確認するのが確実である。

^{☆5} 2011年1月現在JDK1.6、Ant1.8で動作することが確認されている。

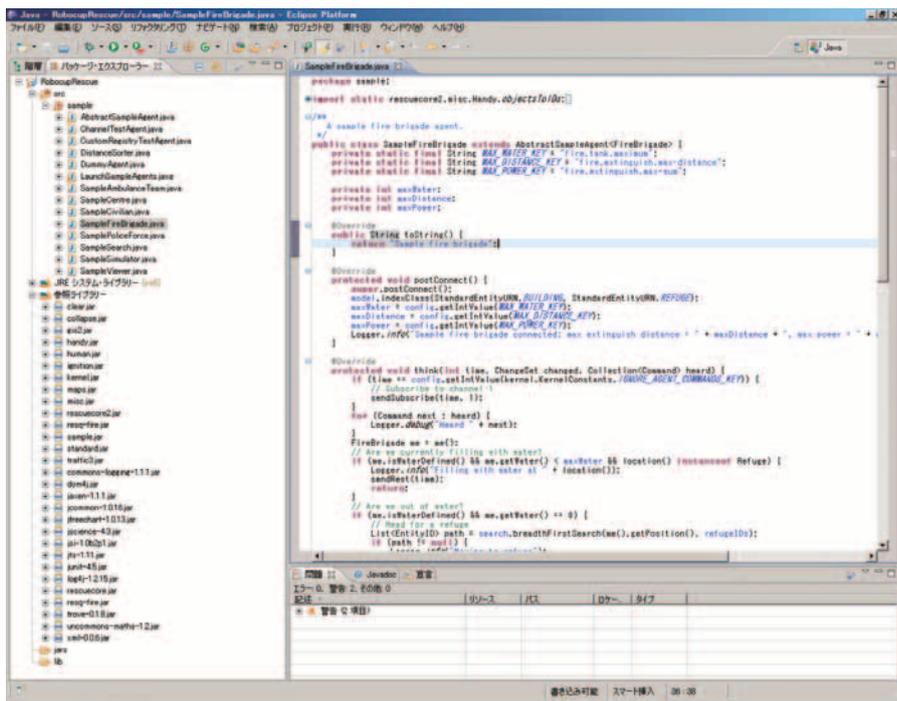


図-4 Eclipse 上でのプロジェクトの作成

• lib 以下：各種ライブラリ
 である。それぞれのフォルダに入っているすべての jar ファイルをビルドパスに追加する。以上で、RCRS サーバの導入が完了である。

次節で、サンプルによる実行を行うため、サンプルエージェントをプロジェクトにコピーする。サンプルエージェントは、レスキューサーバ内の

```
rescue-competition2010/modules/  
sample/src/
```

にあるため src 以下のすべてのファイルを、プロジェクトの src にコピーする。サンプルの中には、最低限のシミュレーションを実行するために必要なエージェントなどが含まれている。

プロジェクトを作成した状態の Eclipse のウィンドウを図-4に示す。Eclipse のバージョンなどによって多少の違いはあるが、おおよそそのような構成ができてはいるはずである。

サンプルプロジェクトの実行

RCRS は、サーバとクライアントからなっている。サーバは、災害空間を管理し、クライアントは作成したエージェントを管理する。それぞれは独立に動

くため、プロジェクトを起動する場合はサーバとクライアントをそれぞれ動かさなければならない。

まず、サーバの起動を行う。コマンドプロンプトを立ち上げ先ほど RCRS サーバを起動したフォルダに移動し、サーバを Ant を利用して起動する。

```
> ant start-kernel
```

なお、これで動かない場合は

```
rescue-competition2010/modules/kernel/  
build.xml
```

内にある、maxmemory の値を 1024m などに変更すると起動する可能性が高い。

実行後に出てくるウィンドウ (図-5) から Maximum にチェックがついているものを Civilian 以外はすべて外し、OK ボタンをクリックすると、KernelGUI と Viewer の 2 つの Window が立ち上がる。これで、サーバの起動は完了である (図-6 参照)。

次に、クライアントを実行する。Eclipse で先ほど作成したプロジェクト内の

```
sample/LaunghSampleAgent
```

を実行する。これで、クライアントが実行され、サーバと接続される。KernelGUI 内に表示される Component manager の表示がすべて消えれば、エー

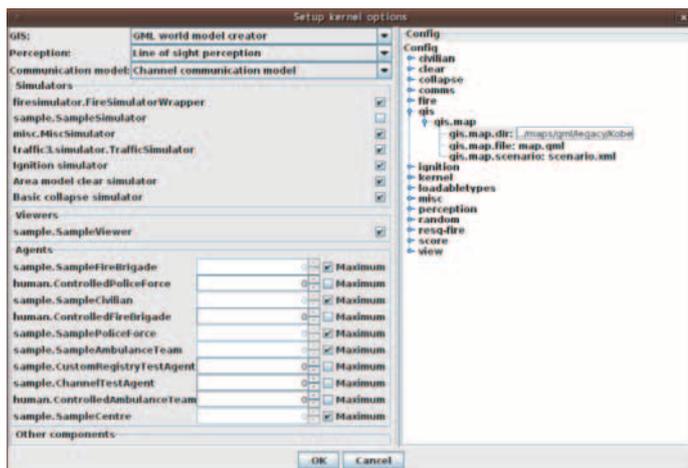


図-5 Setup Kernel Option ウィンドウ

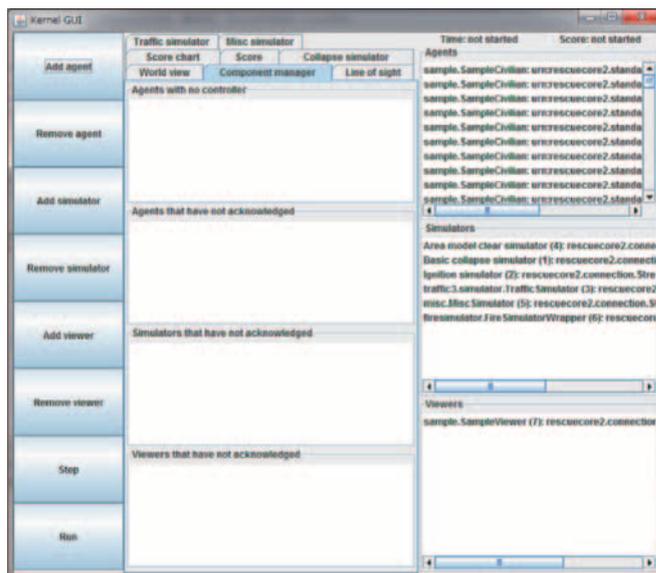


図-6 KernelGUI ウィンドウ

ジェントの接続が完了である。KernelGUIの左下にあるRUNボタンを押せばシミュレーションが開始される。なお、サンプルは非常に単純な動きしかないため、シミュレーション終了時には街が燃え尽きることになる(図-7)。このような状態にならないためにエージェントを開発するのが、ロボカップレスキューシミュレーション参加者の責務である。

マップの変更

前節でサンプルエージェントを動かしたMapは非常に大きいため、実行には時間がかかる。そのため、エージェントを開発してその動きを見るには適さない。そこで、実行するMapを簡単なものに変更する。

利用するMap情報は、

```
rescue-competition2010/boot/config/
gis.cfg
```

で設定する。このファイルを以下のように変更すると、Mapをlegacy/Kobeからtestに変更できる。

```
# The map directory to read
# gis.map.dir : ../maps/gml/legacy/Kobe
gis.map.dir : ../maps/gml/test
```

このMapは各種エージェントが1種類ずつしかない小さいMapである。そのため、実行に必要な

時間も少なく、またエージェントも少ないため動きを確認するのに適している。

エージェントの作成

各エージェントの基本となるクラスは、`rescuecore2.standard.components.StandardAgent`

である。このクラスを継承することによって、各種エージェントを作成する。ただし、エージェントの種類は型総称性(Generics)を利用して決定する。そのため、たとえば啓開エージェントを作成する場合は、

```
StandardAgent<PoliceForce>
```

を継承したクラスを作成することになる。

作成したクラスにあるthinkメソッドの中で各ステップごとのエージェントの行動を記述する。thinkの中身については、サンプルエージェントなどを参照にして作成して欲しい。

作成したエージェントで動かす場合は、

```
sample.LaunchSampleAgents
```

を自分の開発パッケージにコピーし、クライアントの実行時にこれを実行する。このとき、コピーしたLaunchSampleAgentsのconnectメソッドで立ち上げるエージェントを指定することができる

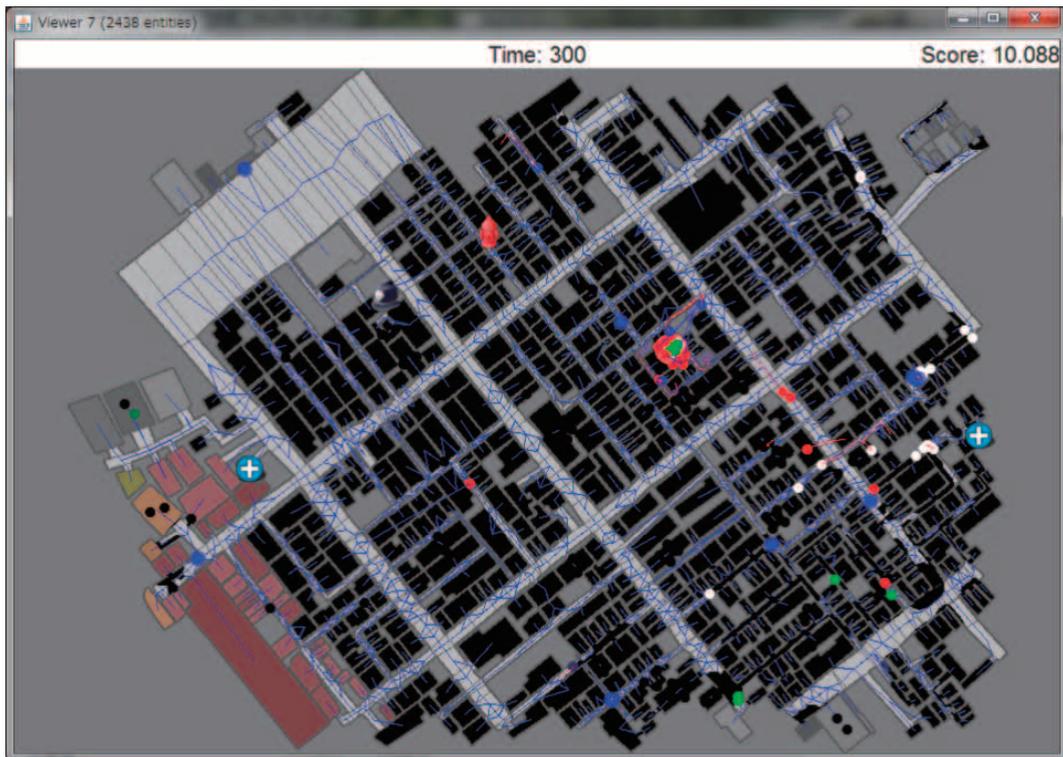


図-7 サンプルエージェントの力及ばず燃え尽きた都市

ため、作成したエージェントと SampleAgent を入れ替える。たとえば、SamplePoliceForceAgent を MyPoliceForceAgent というエージェントに変更するのであれば、

```
launcher.connect
(new SamplePoliceForceAgent());
```

を、

```
launcher.connect
(new MyPoliceForceAgent());
```

に変更すればよい。

エージェントの作成には多くの作業が存在するため、一からエージェントを作成するのは難しい。当初はサンプルエージェントを改造し、どのような動きをするのかを確認しながら少しずつエージェントの作成方法について学んでいくのが良いだろう。

研究課題

ロボカップレスキューシミュレーションの競技としての研究課題は、効率的な救助のためのエージェント間協調やプランニングであるが、それら以外に

も災害シミュレーションあるいはエージェントシミュレーション一般としての研究課題も多い。

現在のロボカップレスキューシミュレーションにおける課題を以下に述べる。

サーバシステムの開発

2010年にRCRSはVer1.0へと更新された。これによって、PureJavaによる開発が可能となり、開発環境を問わなくなり参加者への敷居は大きく下がった。

一方で、サーバシステムの開発も参加者たちの手によって行われており、まだまだ未開発の個所も多い。

サーバシステムはモジュール化して開発されており、

- Kernel (シミュレータサーバ本体)
 - 交通シミュレータ
 - 倒壊シミュレータ
 - 火災シミュレータ
 - 発火シミュレータ
- などからなる。

これらのシミュレータは独自に開発が進められており、たとえば被災者の人間関係を考慮した避難行動を交通シミュレータに導入する試み⁴⁾なども行われている。

これ以外にも、参加者が研究テーマとして扱っている災害現象があれば、それらのシミュレータを開発・改良し、システムに組み込む提案を行うことが可能である。なお、システムの改良については年に1回行われている世界大会において、Infrastructure competition で発表し承認されることでシステムに組み込むことが可能となる。過去には、洪水などの災害シミュレータ^{5), 2)}などが提案されている。

初心者用開発プラットフォームの開発

現在、ロボカップレスキューシミュレーションの研究を開始するためには、3種類のエージェントを開発しなければいけないだけでなく、通信プロトコルの設定などさまざまな課題を解決する必要がある。本来は、他のチームが開発したエージェントを再利用するなどして、切磋琢磨しながらエージェントの救助活動を高度化していくことが目論まれていたが、エージェント間の通信に互換性がないことなどから、実現が難しい。

そのため、初心者が簡単にエージェントを開発できるようなプラットフォームを開発することも重要な課題の1つである。

そのようなプラットフォーム作成の一環として、筆者らのグループが開発している通信ライブラリプロジェクトがある。本システムは、センターとエージェントを結ぶ通信のプロトコルを規定することにより、互いを独立に開発することを目指したものである。本システムを用いて作成されたエージェントは互いに入れ替えが可能で、他のチームが作成したエージェントに指示を出すセンターの開発などを行うことができる。これによって、新しくRCRSプロジェクトに参加したチームが、複雑なエージェントの作成をすることなく、チーム全体の戦略のみを立てることによって、シミュレーションを行うことが可能となる。

なお、2011年現在日本でロボカップレスキューシミュレーションに参加しているチーム全体で、本システムの高度化を目指している。

本システムは、

<http://sourceforge.jp/projects/rcrscs/releases/>

からダウンロードすることが可能である。

競技方法の検討

ロボカップレスキューシミュレーションリーグは、都市災害の救助戦略を対象としたマクロなシミュレーションからスタートした。このため、次の2つの目標が考えられる。

1. 被災地で活躍する人間の災害救助隊に対する意思決定支援
2. 被災地で活躍する災害救助ロボットの行動決定(戦略決定)

プロジェクトの開始時には1.を想定していたが、ロボカップの性質上、2.と勘違いされることが多く、参加者から2.を目標に据えることを提案されることもある。

また、実機リーグでは建物内での災害救助を取り扱っているが、ロボットの構造上の問題を取り扱うことが多いため、その純粋な救助戦略をシミュレーションしたいという要求があり、2007年から仮想ロボットによる競技が小規模ではあるが並列に開催されている。この時間軸も違えば大きさも違う2つの競技を1つの競技にしようという提案もある。

現在は当初の1.を維持しながらプロジェクトを発展させていくことが確認されている。

研究者向けプラットフォーム提供の検討

エージェントのチーム開発は容易ではないため、特定のアルゴリズムやプロトコルを研究・開発している研究者向けにベンチマークとしてパッケージ化して提供しようという動きが国内外で始まっている。たとえば、ドイツのAlexander Kleinerらのパッケージ Rescue Multi-Agent Benchmarking^{☆6}

^{☆6} <http://kaspar.informatik.uni-freiburg.de/~rslb/benchmarks.html>



や、先に挙げた通信ライブラリプロジェクトなどがある。これらの研究については、RoboCup2011のInfrastructure Competitionで発表されることになっている。今後は、さまざまなベンチマークパッケージをまとめ、多くの研究者に利用されることを目指していく。

スコアリングの検討

スコアリングに対する考え方は2種類あり、「エージェント個々の行動を評価」するか「災害救助シミュレーションの結果を評価」するかである。現在では後者が採用されているが、前者をうまく使って後者を評価したいと考えたいとする研究者もいる。現状ではエージェントの協調行動が災害救助を達成していくことから、「災害救助シミュレーションの結果を評価」するとしているが、

- 人命救助だけを評価する
- 保険金額で評価する

などさまざまな意見があり、議論の必要がある課題である。

最新情報の入手

本稿ではロボカップレスキューシミュレーションについて、主に新規参入を目指す読者向けの解説を行った。しかしながら、RCRSサーバは日々更新されており、また、大会ごとに新たなルールが追加されている。本稿は2011年1月現在の状況を記しているが、今後ルールの変更、サーバの変更などが行われるため、情報が古くなる可能性も高い。

ロボカップレスキューシミュレーションに関する最新情報を得たい場合はWEBサイト「SimJP」、

<http://rc-oz.sourceforge.jp/>

で手に入れることが可能である。ロボカップレスキューシミュレーションを新たに始めたいと考える読者は、ぜひこちらのページを訪れて欲しい。

また、ロボカップシミュレーションリーグでは、年に1回開かれるJapanOpenおよび世界大会だけではなく、秋合宿や春期競技会などを積極的に行っている。これらの情報についてもSimJPで掲載されているため、ぜひ確認のうえ参加して欲しいと願う次第である。

参考文献

- 1) Nüssle, T. A., Kleiner, A. and Brenner, M. : Approaching urban Disaster Reality : The Resq Firesimulator, In RoboCup 2004 : Robot Soccer World Cup VIII, Springer, pp.474-482 (2005).
- 2) Takahashi, T., Koto, T., Takeuchi, I. and Noda, I. : A Simulation of Fluid Objects in Disasters - Tokai Heavy Rainfall Simulation using Idss -, In Proc. of the 3rd International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster, pp.24-29 (2006).
- 3) Yotsukura, S., Okaya, M., Sato, K. and Takahashi, T. : Agent Evacuation Simulation using a Hybrid Network & Free Space Models, In Proc of the 12 th International Conference on Principles of Practice in Multi-Agent Systems, to appear (2009).
- 4) 岡谷 賢, 高橋友一: 避難シミュレーションにおける人間関係を考慮した行動モデルの提案, JAWS2010 (2010).
- 5) 谷川昌也, 高橋友一: 分散処理環境における洪水シミュレーターの一実現方法, 第67回情報処理学会全国大会講演論文集, pp.515-516 (2005).
- 6) 田所 論, 北野宏明監修: ロボカップレスキュー: 緊急大規模災害への挑戦, 共立出版 (2000).
(平成23年1月31日受付)

■鳥海不二夫 tori@is.nagoya-u.ac.jp

2004年東京工業大学大学院理工学研究科機械制御システム工学専攻博士課程修了, 同年名古屋大学情報科学研究科助手, 2007年同助教, 現在に至る。エージェントベースシミュレーション, 社会ネットワーク分析, 人工市場などの研究に従事。

本稿作成にあたりさまざまな情報, アドバイスをご提供いただいた愛知工業大学情報科学部 伊藤暢浩准教授および, 名古屋大学大学院情報科学研究科 太田健文氏に感謝いたします。