

マルチエージェントシステムのためのシナリオリハーサル

Scenario Rehearsal for Multiagent Systems

山根 昇平*¹ 石田 亨*¹
Shohei Yamane Toru Ishida

*¹京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Kyoto University

We proposed rehearsal oriented testing by which a human interrupts a multiagent simulation and modifies an agent's behavior through dialogue with it. In this research, to realize rehearsal oriented testing, we first defined functions for modifying scenarios described on the basis of finite state machine model. In addition, we designed interaction protocol between the human and the agent to get information required to execute the modification functions. Under this protocol, the agent tells the human what kind of directions the human can give in order for the human to know modification procedure. We designed this protocol so that dialogues becomes more natural and simple using context in the simulation.

1. はじめに

人間をエージェントとしてモデル化して行うマルチエージェントシミュレーションや、複数のサービスエージェントが互いに協調して人間をサポートするシステムなど、マルチエージェントシステムの応用は多岐にわたる。このようなマルチエージェントシステムの開発にあたっては、利用者の意見を取り入れることが重要となる。サービスを行うエージェントを開発する場合には、実際の利用者の意見を取り入れることで、サービスの質を向上させることができる。また、システムの評価をマルチエージェントシミュレーションを用いて行う場合には、システムの利用者がエージェントとしてモデル化されるため、より利用者を忠実に再現するモデルが求められる。

マルチエージェントシステムの開発において、利用者を参加させる参加型デザインを行う手法として、参加型シミュレーションが挙げられる [Guyot 2005]。参加型シミュレーションとは、マルチエージェントシミュレーションの中に、被験者が計算機を通して操作するエージェントを参加させるものである。参加型シミュレーションによって得られた被験者の操作ログや被験者へのインタビューから、被験者を模倣するためのエージェントのモデルを得る手法について研究がなされている [Murakami 2005]。

エージェントの修正や拡張についても、利用者の意見を取り入れることが好ましい。しかし、修正や拡張が必要になるときにエージェントの作成プロセスを繰り返すのは効率的ではない。そこで、エージェントの修正や拡張を、利用者自身が行うことができれば、利用者の意見を効果的に取り入れたエージェントの修正や拡張が可能となる。また、サービスの利用者自身がサービスを修正したり拡張したりすることによって、利用者に適したサービスが、システム開発者を介することなく実現できる。

我々の目的は、プログラミング言語やエージェント技術などエージェント開発に必要な知識を持たない一般の利用者に、エージェントの修正や拡張を行わせることのできる技術を開発することである。この目的のため、本稿では、リハーサル指向テストを提案する。リハーサル指向テストとは、エージェントと人間とがエージェントの動作に関する対話を行うことによって、舞台のリハーサルで監督が役者に指示を出すように、エ

ージェントを修正/拡張するものである。

本稿では、演劇におけるリハーサルをもとに、リハーサル指向テストのプロセスを述べるとともに、エージェントの動作を修正するための、人間とエージェントとの間の対話プロトコルを設計する。

2. リハーサル指向テスト

本研究では、エージェントの外的な振舞い（他のエージェントや環境とのインタラクション）をシナリオとして記述し、エージェントの内部モデルと分離させる手法 [Ishida 2002] をとる。エージェントの外的な振舞いであるシナリオと、エージェントの内部モデルを分離させることにより、リハーサル指向テストで修正の対象となる部分を明確化することができる。すなわち、リハーサル指向テストではシナリオを修正の対象とすればよい。また、シナリオを記述するモデルとして、状態遷移モデルを用いる。状態遷移モデルは記述が簡易かつ、図で表現することで視覚的に理解が可能であるので、計算機の非専門家であっても比較的容易にその動作を理解できる。

リハーサル指向テストの実現のため、演劇のリハーサルにおける、次のふたつの特徴に注目した。ひとつは、演劇の実行中における修正である。リハーサルでは、本番と同様の形式で演劇を行い、それを監督が観察する。そして役者の演技に問題点を発見した場合、その場で修正し、確認を行う。もうひとつは、コンテキストの共有である。監督と役者は、修正の指示を行っている時点において、舞台の状況、他の役者の状況、台本の内容と進行状況などの情報を共有している。

以上のような特徴をもとに、マルチエージェントシミュレーションにおけるリハーサル指向テストのプロセスについて述べる。リハーサル指向テストは、大きく分けてふたつのプロセスから成る。すなわち、ひとつはマルチエージェントシミュレーションを実行して問題点を発見するプロセスで、もうひとつは発見した問題点を修正するプロセスである。

マルチエージェントシミュレーションの実行と問題点の発見。マルチエージェントシミュレーションを実行する。それぞれのエージェントは、それぞれに与えられたシナリオに従って動作する。利用者はこのシミュレーションを監視し、問題のある振舞いをするエージェントがないかどうか確認する。もし問題点を発見した場合、シミュレーションを停止して問題点の修正へと移行する。

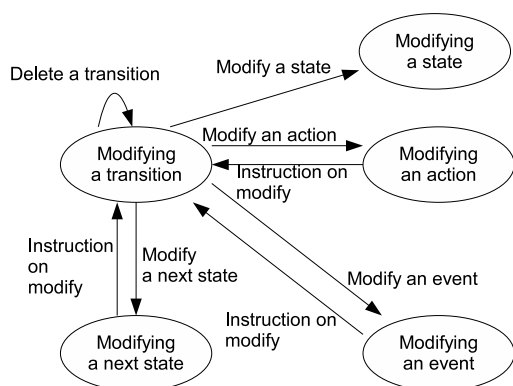


図 1: 状態遷移の修正と削除を行うための対話プロトコル

発見した問題点の修正．エージェントの振舞いに問題点を発見した場合、シミュレーションを停止して修正を行う．利用者は、エージェントに対してどのように振る舞えばよいかを指示する．エージェントは、利用者からの指示とコンテキストから、指示を満足するようシナリオの状態遷移機械に修正を加える．修正が完了すると、マルチエージェントシミュレーションを再開する．このとき、修正を指示されたエージェントは、修正後のシナリオに従って動作する．

状態遷移で記述されたシナリオの修正にあたり、考えられる修正機能は、状態の追加、状態の削除、状態遷移の追加、状態遷移の削除、状態遷移の変更の5つである．それぞれの修正を行う場合に、例えば状態の削除であれば削除する状態、状態遷移の追加であれば状態遷移の各要素といったように、人間が修正に必要な情報を与える必要がある．リハーサル指向テストでは、これらの修正機能に必要な情報を人間とエージェントの対話により収集する．

3. 対話プロトコル

本章では、シナリオの修正を行うための、利用者とエージェントとの間の対話プロトコルについて説明する．

修正を開始した時点におけるエージェントのシナリオの実行状態は、状態遷移中のアクションを実行しているか、あるいは、ある状態でイベントを待機している状態である．修正開始時の対話プロトコルは、これらふたつのシナリオ実行状態のうちどちらの状態にあるかによって、ふたつに分岐する．このときコンテキストとして、実行中の状態遷移 δ_c 、あるいは、待機中の状態 q_c が用いられる．

状態遷移実行中における対話プロトコル

図 1 に、状態遷移実行中における対話プロトコルを示す．ここでは、状態遷移の修正と削除が行われる．状態遷移関数の修正については、まず、観測するイベント、実行するアクション、遷移先の状態のどれを修正するかを選択する．そして、それぞれについてどのように修正するかを指示する．さらに修正する必要がある場合には、同様の対話を繰り返す．

状態遷移関数の削除については、削除の指示を出すことで、修正の対象となっている状態遷移関数 δ_c の削除が行われる．また、状態遷移の追加などの状態の修正を行う場合は、状態の修正を行うことを伝える．修正の対象となる状態 q_c には、状態遷移関数 δ_c の遷移前の状態が用いられる．

イベント待機中における対話プロトコル

図 2 に、イベント待機中における対話プロトコルを示す．こ

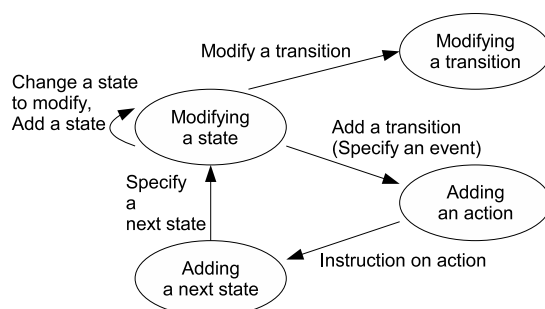


図 2: 状態の追加と削除、状態遷移の追加を行うための対話プロトコル

こでは、状態の追加と削除、状態遷移の追加が行われる．状態の追加については、状態の追加を指示することで、新しい状態を作成する．

状態遷移の追加を指示すると、まず、追加される状態遷移の遷移前の状態としては、 q_c が選択される．その後、状態遷移の追加に必要な情報をひとつずつたずねる．図 2 の対話プロトコルでは、状態遷移の追加を指示するときに、その状態遷移の発火条件となるイベントを最初に指示する．続いて、その状態遷移に伴って実行されるアクションを指示する．最後に、遷移後の状態を指定する．

また、 q_c を起点とする状態遷移の修正や削除を行う場合には、修正や削除の対象となる状態遷移を指定して、状態遷移の修正に関する対話に移行する．

4. おわりに

本稿では、シナリオの拡張や修正を行う手法として、リハーサル指向テストを提案した．本研究では特に、エージェントの動作を人間との対話によって修正するというリハーサル指向テストのステップについて注目し、詳細な機能定義と対話プロトコルの設計を行った．この対話プロトコルの設計にあたっては、修正開始時のコンテキストを用いて、自然で簡潔な対話を目指した．

謝辞

本研究は、日本学術振興会科学研究費基盤研究 (A)(18200009)、戦略的情報通信研究開発推進制度地域情報通信技術振興型研究開発の助成を受けて行われた．

参考文献

- [Guyot 2005] Guyot, P., Drogoul, A. and Lemaitre, C.: Using emergence in participatory simulations to design multi-agent systems, *Proceedings of The Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)* (2005).
- [Ishida 2002] Ishida, T.: Q: A Scenario Description Language for Interactive Agents, *IEEE Computer*, Vol. 35, No. 11, pp. 42–47 (2002).
- [Murakami 2005] Murakami, Y., Sugimoto, Y. and Ishida, T.: Modeling Human Behavior for Virtual Training Systems, *The Twentieth National Conference on Artificial Intelligence (AAAI-05)* (2005).