

キーイベントに基づく対話場面の生成 Conversation Scene Generation Using Key Event Control

生野 剛* 森 博志† 星野 准一‡
Tsuyoshi Ikuno* Hiroshi Mori† Junichi Hoshino‡

筑波大学 工学システム学類* 筑波大学大学院 システム情報工学研究科† 筑波大学機能工学系‡
College of Engineering Systems, University of Tsukuba*
Systems and Information Engineering, University of Tsukuba†
University of Tsukuba‡

Abstract: In this paper, we propose a new technique for controlling the flow of conversation in interactive storytelling applications. We use key event technique to specify the utterance that the character should talk at proper timing. First, conversation network is constructed from a set of conversation modules and its relative connections. An optimal path of utterances is generated by searching the conversation network. By using the key event technique, locally interactive conversations are generated with global flow constraints. We show the example of "Sherlock Holmes" animation sequences using global flow control.

1. はじめに

映画やビデオゲーム、教育などのマルチメディアコンテンツを扱う分野、特にエンタテインメントの分野において、ストーリー性のある仮想空間を体験できる対話型映画システム [1] [2] [3] の実現が望まれている。

従来、対話型映画システムでは仮想俳優と利用者の対話の生成を、利用者の発話とストーリーボードの条件部に記述されたキーワードをマッチングすることで適切な返答を選択する手法が用いられている [7]。ところが、この方法では、ある特定の方向へ会話を誘導していく等の大局的な会話の制御を行うことが難しい。また、仮想俳優がある話題から連想して関連することを話すことも難しい。例えば、ある登場人物と出会って一緒に出かけるような場面において、一緒に出かけるという結論に達する会話の流れを作り出したいというような制御の需要に対応するという事例が考えられる。

また、対話エージェントの分野では様々な研究が行われているが [4] [5] [6]、ストーリーと連動して大局的な会話を制御することについては重点が置かれて

いない。

そこで本稿では、大局的な会話の制御や話題の連想を実現する手法として、キーとなる台詞の指定 (キーイベント [8]) の利用を提案する。本手法ではキーイベントを設定すると、そのキーイベントによって指定された台詞を必ず発話するような一連の会話の流れ (発話列) を生成する。

キーイベントは仮想俳優、台詞モジュール、発話タイミングを指定するものである。このキーイベントを記述することで大局的な会話の流れを構築する。このようにして記述されたキーイベントは一系列に接続され、キーイベント系列を構成する。

本手法では、事前に台詞モジュールを有向アークで接続して、発話される台詞の前後関係や意味的な関連を表した台詞ネットワークを構築する。ネットワー

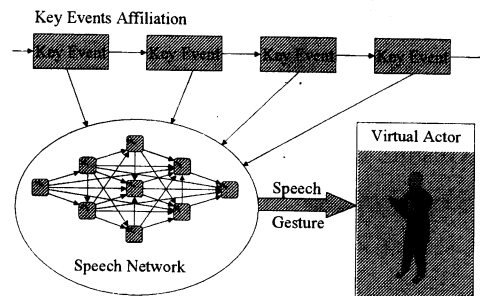


図 1 手法の概要

生野 剛 筑波大学工学システム学類
〒305-8577 茨城県つくば市天王台 1-1-1
Tel: 029-853-6555
E-mail: ikuno@graphic.esys.tsukuba.ac.jp

クの遷移コストは台詞モジュール間の距離を表し、距離が短いほど台詞同士の連続性が強いことを意味する。構築したネットワークを基に、場面内で必ず話して欲しい内容をキーイベントによって指定する。指定された台詞モジュール間のパスを探索し、算出することで仮想俳優の台詞の流れ（発話列）を生成する（図1）。

例えば、ある登場人物と出会うと一緒に出かけるような場面では、一緒に出かけるという結論になって欲しいため、ネットワークを探索してこの結論に帰結するように発話列のパスを求めらる。

2. 台詞ネットワークの構成

本章では、仮想俳優の発話の基盤となる台詞ネットワークの構築方法について述べる。台詞ネットワークは仮想俳優の発話を生成する台詞モジュールを有向グラフによって連結することで構築される。

この台詞ネットワークの構築はキーイベントによる対話の制御の事前処理として行われる。

2.1. 台詞モジュール

台詞ネットワークにおけるノードに相当し、実行対象が遷移してくると以下の三つの機能が実行される（図2）。

1. 仮想俳優への台詞データ出力機能
2. 仮想俳優へのジェスチャデータ出力機能
3. 他の仮想俳優のキーイベント系列へ新たなキーイベントを挿入する機能

1と2は仮想俳優に対する出力機能で、対象となる仮想俳優以外に影響を与えないのに対し、3は唯一他の仮想俳優に影響を与えるという特徴がある。本稿ではユーザと仮想俳優の間での対話を想定しているが、3の機能を実装することにより他の仮想俳優の発話列に影響を与えることが可能になる。このため、これは本稿の手法を複数人仮想俳優の相互対話へ拡張対応する際に必要となる機能である。

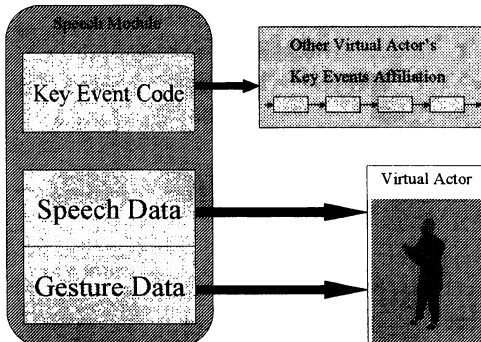


図2 台詞モジュールと仮想俳優

2.2. 台詞ネットワーク

台詞ネットワークは仮想俳優ごとに用意された固有の台詞モジュール群をノードとし、それらを有向アークで接続したものである（図3）。

以下の処理を交互に繰り返すことで一連の台詞の流れ（発話列）が出力される。

1. 台詞ネットワークを実行対象が遷移
2. 対象となった台詞モジュールが実行される

既存の台詞ネットワークへの台詞モジュールの新規追加は、台詞モジュール同士を有向アークで連結することにより可能となる。そのため、過去の蓄積データの再利用性があり、また拡張性が高いという利点がある。

2.3. 台詞ネットワークの遷移コスト

仮想俳優 α の台詞ネットワーク N_α とし、これを構

成する台詞モジュール集合を M_α とする。台詞モジュール集合を節点集合とし、節点間の遷移を表現する節点対の有限集合を E とすると、構築される有向グラフは $G=(M, E)$ として表現される。また、 G において隣

接する台詞モジュール u_M と v_M を結ぶアーク $a=(u_M, v_M)$ の長さを $length_M(a)$ と表現する。このとき、

$length_M(a)$ は台詞間の距離を表し、距離が短いほど台詞同士の意味的な連続性が強く、自然な遷移を可能にすることを意味する。

本研究においては手動で主観による判断に基づいて設定した。

3. キーイベントによる対話制御

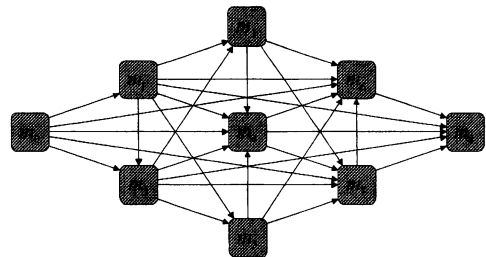


図3 台詞ネットワーク

3.1. キーイベントと台詞ネットワーク

本手法では事前処理として台詞ネットワークを構築し、これを基にキーイベントの設定を変更することで、ストーリーの拘束条件を変更した発話列を生成することができる。以下にキーイベント制御による発話列の生成過程を示す。

1. ストーリーの拘束条件として、一列にキーイベントを配置する。
2. 配置されたキーイベントを基に、台詞ネットワークの探索により、仮想俳優の発話列を生成する。

3.2. キーイベントの設定

キーイベントはストーリー中で確実に話して欲しい台詞を指定するもので、対象となる仮想俳優の台詞モジュールを指定する(図4)。以下にスクリプト言語による記述形式例を示す。

例1 直前のキーイベントを基準に指定

Actor[Virtual Actor]
Talk[Speech Module]
after[Previous Key Event]

例2 直後のキーイベントを基準に指定

Actor[Virtual Actor]
Talk[Speech Module]
before[Next Key Event]

例3 絶対時間を基準に指定

Actor[Virtual Actor]
Talk[Speech Module]
Time[Time]

Virtual Actor

対象となる仮想俳優 ID

Speech Module

実行する台詞モジュール ID

Previous Key Event

挿入されるキーイベントの直前にあたる対象仮想俳優のキーイベント ID

Next Key Event

挿入されるキーイベントの直後にあたる対象仮想俳優のキーイベント ID

Time

キーイベントで指定する台詞モジュールを実行する仮想空間内での絶対時刻

例1は「Aの後にBを話す」と指定する場合の記述法で、例2は逆に「Cの前にBを話す」と指定をする

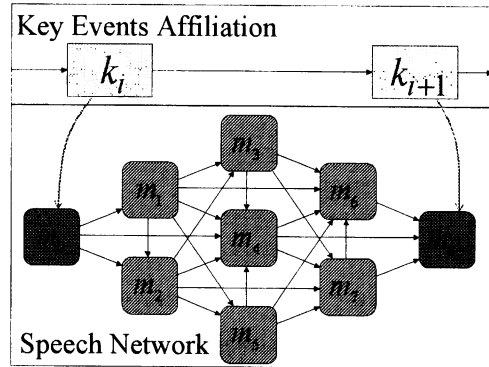


図4 キーイベントによる台詞モジュール指定

場合の記述法である。例3は他の二つとは別に、発話のタイミングを時刻で指定をしたいときに使用する記述法で、例えば「5時になったから帰りましょう」「おやつの時間ですよ」など、仮想空間内の特定の時刻に依存する台詞を指定する時に使用する。

ドラマや演劇等で使用される台本において、全俳優の台詞は時間経過に沿って配置されているが、本手法においては仮想俳優を個別に制御するため、対象となる仮想俳優ごとに台詞ネットワークを構築する。

4. ネットワーク探索による発話生成

事前処理によって構築された台詞ネットワークを基に、3章で述べたキーイベントの設定を変更することで、ストーリーの拘束条件を変更した発話の流れの生成が可能となる。以下にキーイベント制御による発話列の生成過程を示す。

1. ストーリーの拘束条件として、キーイベントを配置によって大局的な会話の流れを記述する。
2. 配置されたキーイベントを基に、台詞ネットワークの探索により、仮想俳優の発話列を生成する。

本章では、キーイベント間の発話列の生成方法について述べる。

4.1. 探索アルゴリズム

仮想俳優 α のキーイベントリストうち隣接するキーイベント k_i と k_{i+1} 間の発話列を生成する場合を考える(図4)。

キーイベント k_i と k_{i+1} によって指定された台詞モジュールをそれぞれ m_i, m_{i+1} とする。台詞ネットワーク中において m_i を始点、 m_{i+1} を終点とする経路を $P_{i,i+1}$

とする。経路 $P_{i,i+1}$ は、生成される発話列を自然な繋がりを持ったものとするために、 $P_{i,i+1}$ がとり得る経路のうち、コストの合計が最も低いものを選択する。

この最短経路探索問題に対し、本稿ではダイクストラ法を用いて最短経路を求める。こうして得られた経路に従って、実行対象の遷移・台詞モジュールの実行を繰り返すことで、一連の発話列が生成される。

この処理を k_i と k_{i+1} の間、 k_{i+1} と k_{i+2} の間……と繰り返すことで、キーイベント配置によって構成した大局的な会話の流れに沿った発話列が出力される。

4.2. キーイベント挿入による動的な発話生成

仮想俳優が話している途中にユーザ入力があった場合、仮想俳優はそのインタラクションに対し適切な反応を返す必要がある。これを実現するため、台詞モジュールの実行中に新たなキーイベントを挿入することを考える。

k_{insert} をキーイベント系列へ挿入した場合、 k_{insert} に対応した経路を再計算する必要がある。ここでキーイベント系列中の2つのキーイベント k_i と k_{i+1} によって指定された台詞モジュール m_j, m_{j+1} 間を遷移・実行中に、中断処理によって2つのキーイベント間に新たなキーイベント k_{insert} が指定された場合を考える。 k_{insert} が指定された時点で実行している台詞モジュールを m_j 、 k_{insert} によって指定された台詞モジュールを m_k とし、ここで始点を m_j 、終点を m_k として4.1節の経路探索手法によって新たな経路 $P_{j,k}$ を得る。同様に、始

点を m_k 、終点を m_{i+1} として経路を探索、発話経路 $P_{k,i+1}$ を獲得する。

以上の処理により、台詞モジュールの実行中に新たなキーイベントを挿入された場合においても、経路の再計算を行う。これによって動的に発話列を生成することが可能となる。

4.3. ユーザ入力の方法

4.3節で、ユーザ入力によるキーイベント挿入について述べた。しかし、対話型映画においてユーザが仮想俳優と対話を行う際に、いちいち3.2節で示したよ

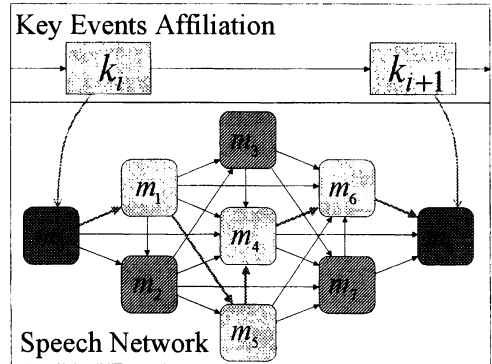


図5 ネットワーク探索による経路算出

うなスクリプト言語を打ち込むのは煩雑である。

この問題を解決するためには二つの入力方法が考えられる。一つ目はユーザの発話を選択的にし、選択した発話に関連付けられたスクリプトコマンドを発行するという手法が考えられる。

二つ目は従来技術であるキーワードマッチングによって、ユーザの自然言語入力からキーワードマッチングを行い、その結果としてキーイベントを設定するスクリプトコマンドを出力するという手法が考えられる。

前者はユーザ入力の自由度が制限されてしまうという欠点があり、後者はその点で入力の自由度が保証されている。

しかし、本研究の目的であるキーイベントによる大局的な発話列制御を実証するにあたって、後者の手法ではキーワードマッチングデータベースの構築を行うのが煩雑であるのに対し、前者は有限種のユーザの発話とスクリプトコマンドが直接対応しており、本稿における実験には適している。

そこで、本研究においてユーザ入力は有限種の発話パターンの中から選択するものとする。

5. 実験

本手法の有効性を示すために、キーイベントの設定から発話列の生成を行った。

本実験ではシャーロックホームズシリーズの短編「The Priory School」の一場面より台詞を抽出、追加・修正を行い、19の台詞モジュールを作成し、台詞ネットワークを構築した(図7)。

一つ目の実験として4.1節で示した、キーイベントを予め2つ設定する場合を想定した実験を行った。3.2節で示したスクリプト言語によるキーイベント記述は以下ようになる。

A: Actor[Holmes] Talk[ID:9] before[ID:B]

B: Actor[Holmes] Talk[ID:18] after[ID:A]

生成結果を表1に示す。2つのキーイベント設定による発話列の生成が確認できる。

二つ目の実験として、一つ目の実験と同様に予め2つのキーイベントを入力として与え、台詞モジュールID[12]を実行中に新たなキーイベント

C: Actor[Holmes] Talk[ID:12] after[ID:A]

D: Actor[Holmes] Talk[ID:5] before[ID:D]

を挿入した。生成結果を表2に示す。仮想俳優が話している途中にユーザ入力があった場合、挿入した時点から先のパスが再計算され、ユーザ入力に対応した発話列が生成されていることが確認できる。

6. おわりに

本稿では、キーイベントによって大局的な会話の制御・話題の連想を実現する手法を提案した。

2つのキーイベントによって大局的な会話の流れを設定、仮想俳優に必ず話して欲しい台詞を指定した。その結果、それらの台詞をそれぞれ始点・終点とする台詞ネットワーク内のパスを算出し、適切な発話列を出力できることを確認した。また、台詞モジュール実行中に新たなキーイベントが挿入された際にも、パス

を再計算することで、動的な発話列の制御が可能であることも確認した。

今後の課題としては、大規模な台詞ネットワークを作成するために必要な台詞間の距離を自動的に設定する方法の考案、4.3節で示したキーワードマッチングによるキーイベント生成の実装手法の検討、2.1節で示した他の仮想俳優のキーイベント系列へ新たなキーイベントを挿入する機能を利用した、複数人対話実現の方法の検討が挙げられる。



図6 画面出力例

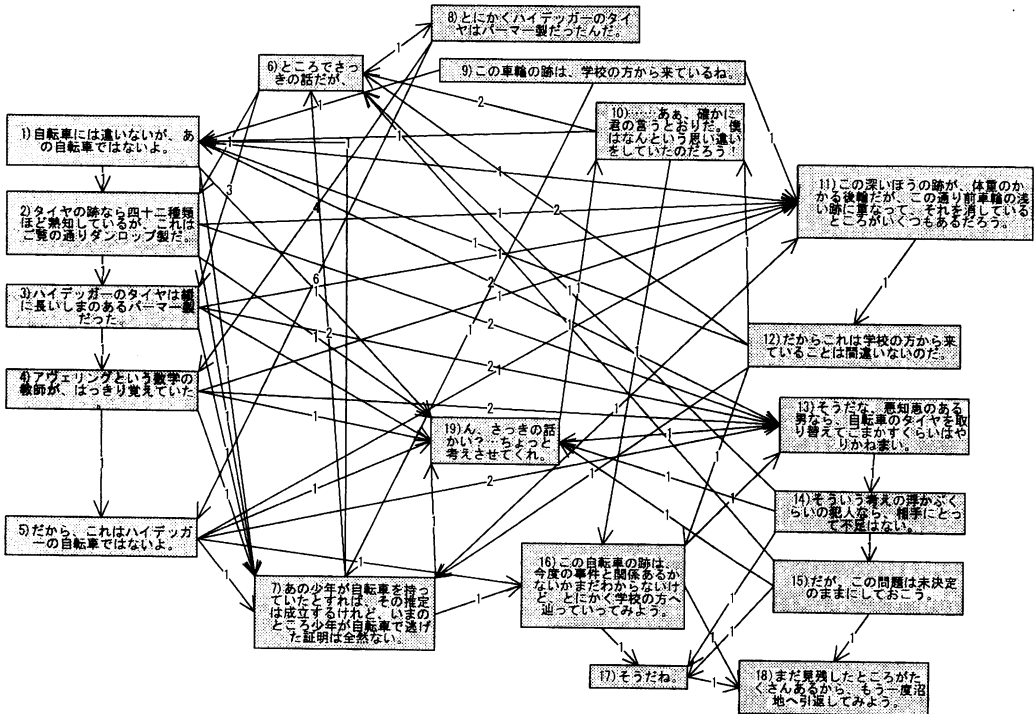


図7 実験に使用した台詞ネットワーク

表 1 キーイベントを二箇所設定した場合の実験結果

Actor	Speech Module ID	The contents of the speech
Holmes	9	この車輪の跡は、学校の方から来ているね。
Holmes	11	この深いまの跡が、体重のかかる後輪だが、この通り前車輪の浅い跡に重なって、それを消しているところがいくつもあるだろう。
Holmes	12	だからこれは学校の方から来ていることは間違いないのだ。
Holmes	16	この自転車の跡は、今度の事件と関係あるかないかまだわからないけど、とにかく学校の方へ辿っていってみよう。
Holmes	18	まだ見残したところがたくさんあるから、もう一度沼地へ引返してみよう。

表 2 キーイベント挿入による実験結果

Actor	Speech Module ID	The contents of the speech	Speech module ID specified by new key event
Holmes	9	この車輪の跡は、学校の方から来ているね。	
Holmes	11	この深いまの跡が、体重のかかる後輪だが、この通り前車輪の浅い跡に重なって、それを消しているところがいくつもあるだろう。	
Holmes	12	だからこれは学校の方から来ていることは間違いないのだ。	
User's Speech		ハイデッカーの自転車だよ？	5
Holmes	1	自転車には違いないが、あの自転車ではないよ。	
Holmes	2	タイヤの跡なら四十二種類ほど熟知しているが、これはご覧の通りダンロップ製だ。	
Holmes	3	ハイデッカーのタイヤは密に長いし、あるパーマー製だった。	
Holmes	4	アヴェリングという数学の教師が、はっきり覚えていた。	
Holmes	5	だから、これはハイデッカーの自転車ではないよ。	
Holmes	16	この自転車の跡は、今度の事件と関係あるかないかまだわからないけど、とにかく学校の方へ辿っていってみよう。	
Holmes	18	まだ見残したところがたくさんあるから、もう一度沼地へ引返してみよう。	

参考文献

- [1] 高沢潤, 後藤大輔, 星野准一: "対話型映画のための会話生成法", 電子情報通信学会技術報告, Vol.103, No.166, pp.29-34, 2003
- [2] 高沢潤, 星野准一: "対話的仮想シアターのためのオーサリング環境", 日本バーチャルリアリティ学会第7回大会論文集, pp.187-188, 2002
- [3] 中野敦, 益子宗, 星野准一: "対話型映画のための実写に基づく仮想俳優の生成", 電子情報通信学会技術報告, Vol.103, No.166, pp.17-22, 2003
- [4] 土肥浩, 石塚満: "WWW と連携する擬人化エージェントとの HAI", 人工知能学会誌, Vol.17, No.6, pp.693-700, 2002
- [5] 大澤一郎: "基づく柔らかなAIシステム", 日本ソフトウェア科学会第16回大会論文集, 1999
- [6] David Traum and Jeff Rickel: "Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds", AAMAS 2002, pp. 766-773, July 2002.
- [7] 星野准一: "没入型会話空間における知識共有", 社会技術研究論文集, Vol.1, pp.116-122, 2003
- [8] 森博志, 星野准一: "キーイベント制御による仮想俳優の動作生成", 電子情報通信学会技術研究報告, MVE2003-57~64, 信学技報, Vol.103 No.351, pp.37-42