

対話的演出機能をもつマルチメディア提示システムの実現

的野晃整† 藤野猛士† 平野尚孝† 板谷昌洋† 横田一正†

† 岡山県立大学情報系工学研究科

{matono, fujino, hirano, itadani, yokota}@c.oka-pu.ac.jp

〒719-1197 総社市窪木 111

ネットワーク上に分散した多種多様な情報を統合するための知識表現言語 QUIK をマルチメディア情報を扱うように拡張した QUIK-II システムを研究開発している。そのシステムの実証システムとして、対話的演出機能をもつ戯曲提示システムを実現した。本稿では QUIK II システムおよび戯曲提示システムについて報告する。

QUIK II システムの特徴は、1) オブジェクト指向風のメディア表現、2) メディアの同期制御のルール表現、3) 対話的な演出機能、4) 入れ子トランザクションによる演出の制御機能、などである。戯曲提示システムとしては、1) XML と QUIK による戯曲表現、2) 戯曲の進行に対応した多彩な演出機能、などがある。

キーワード: QUIK II, マルチメディア, 同期制御, 演出, 戯曲

Implementation of Multimedia Presentation System with Interactively Directed Function

Akiyoshi Matono† Takeshi Fujino† Naotaka Hirano†

Masahiro Itadani† Kazumasa Yokota†

† Okayama Prefectural University, Graduate School of Systems Engineering

{matono, fujino, hirano, itadani, yokota}@c.oka-pu.ac.jp

Soja, Okayama 719-1197

We have developed a knowledge representation language, QUIK-II, which is an extension of QUIK, for integrating various information including multimedia information distributed over networks. Further we have developed an interactive multimedia presentation system of dramas as an application of QUIK-II. In this paper, we describe their features of the language and systems.

The characteristics of QUIK-II are 1) media representation as in object-orientation, 2) rule representation of synchronous control of media, 3) interactive director functions, and 4) nested transaction-based control of directions. The characteristics of our drama presentation system are 1) drama representation in XML and QUIK-II, and 2) rich director functions according to a drama scenario.

Key words: QUIK II, Maltimedia, Synchronus Control, Director, Drama

1 はじめに

インターネットの普及に伴いネットワーク上に散在する情報は急速な勢いで増加しつつある。それに含まれる情報は電子化された文書の他に静止画、動画および音声など様々なメディアや種々の言語によって作成されたアプリケーションなど多種多様化されてきている。このような多様な情報源を統合し、有効に利用するためのシステムが重要となってきた。

そこで我々は、異種分散情報源の統合を行うシステムとして、演繹オブジェクト指向パラダイムに基づいた知識表現言語 QUIK の研究開発を行っている [2][3]。

これまでの QUIK では、情報源として英数字などの記号で記述したオブジェクトを対象としていた。しかし、多様な情報を考えると静止画や音楽と言ったマルチメディア情報を扱うことは自然なことである。そこで昨年、QUIK でマルチメディア情報を扱うための拡張機能を設計し実装した [4][5][6]。本稿では昨年の問題点をふまえ、より対話的かつ視覚的な演出を行えるよう拡張し実装したシステム (QUIK II) について報告する。

本稿の構成は、2 節で QUIK でマルチメディア情報を扱うために追加した機能や概念などについて述べ、3 節では、マルチメディア情報提示の応用として戯曲上演・演出システムの構築・設計について述べる。

2 マルチメディア情報の導入

まず、本稿で扱う戯曲の場面などの用語を定義する。戯曲の物語全体をストーリー、台詞やト書きなどの最小単位のことをイベントと呼び、イベントの集合やその集合の集合などをシナリオと呼ぶ。また、シーンとはストーリー、シナリオ、イベントのすべてで、汎用的な意味を持つ。つまり、一般的な木構造のルートにストーリー、子を持つノードをシナリオ、リーフをイベントと呼ぶ。詳しくは 2.3 節で述べる。

次に、QUIK II でマルチメディア情報を扱う際に何をメディアとして定義するかを議論する。例えば、テキストは発話すると音声になり、描画すると画像になり得る。つまり、1 種類のメディアに対して複数の提示方法があることがわかる。我々は、メディアと提示方式を組合わせそれらの提示を同期制御することで、戯曲を上演できると考えた。本システムでは、1 つのイベントには、1 つのメディア自体の

情報とそのメディアの提示方式の情報 (アクションと呼ぶ) を 1 つの組として持たせている。

このように、オブジェクト (メディア) とメッセージ (アクション) を別々に定義し、オブジェクト思考的に、それぞれを 1 つの組として持つことでマルチメディアの制御を容易できると考えた。

2.1 対象メディアの定義

実際に QUIK II で対象とするメディアは、以下の 6 種類を扱うことにした。

- 1) 静止画 : gif, jpg, bmp などの画像ファイル。
- 2) 動画 : mpg, mov, avi などの動画ファイル。
- 3) 3D オブジェクト : VRML や Java3D などの 3D オブジェクト。
- 4) テキスト : 発話やテロップなどの文字列。
- 5) サウンド : wav, mid などの音声ファイル。
- 6) 外部メディア : SMIL や TVML などの他のメディア。

TVML や SMIL などの他のメディアを外部メディアとして取り込むことでそれらと同様の機能を QUIK 自体に実装するより、実装が容易になり、外部メディアの長所をそのまま活かせるので、演出自体の表現力を豊かにすることができる。

2.2 アクションの定義

アクションとは前述したメディアの提示方式のことで、以下の 5 種類を扱うことにした。

- 1) 出力 : メディアを表示させる。
- 2) 移動 : メディアを移動させる。
- 3) 発話 : テキストメディアを発話させる。
- 4) 照明 : メディアに照明効果をおこなう。
- 5) 消去 : 表示中のメディアを消去する。

本システムは TVML のように個別のメディアの細かい動作の指定 (アクション) は行っていない。つまり、本システムの第一目的が、後述する論理プログラミング言語で複数のメディアを同期制御し、対話的編集・演出を可能にすることである。細かい動作を必要とする応用に対しては、さらにアクションを付加したり、メディアの制御機能を追加することで可能であると考えている。

2.3 同期モデル

ここで、マルチメディア情報を戯曲として上演するために必要なモデルについて述べる。

QUIK ではオブジェクト間の関係はルールの集合で記述していた。マルチメディア情報によって戯曲を上演するためには、複数のメディアが同期をもって提示される必要がある。

そこで戯曲をマルチメディア情報で表現するためにグラフモデルを考える。前述の説明では戯曲の流れだけを表現するために木構造と述べたが正確には有向非巡回グラフになる。グラフモデルはノードにシーンを対応させたもので、ルートノードはストーリー、子を持つノードはシナリオ、リーフはイベントである。イベントはあるメディア自体の情報とそのメディアの起こすアクションの情報を持っている。つまり、複数のイベントが1つのメディアやアクションを共有することがある。

また、シーンの実行順序は逐次実行と非順序実行があり、同階層の隣り合うシーンをつなぐエッジに実行順序が対応している。実行順序については2.5節で説明する。このグラフモデルのことをシーングラフと呼ぶ。

シーングラフを QUIK のルールによって記述する場合、ヘッドをシナリオ、ボディの各要素にイベントまたはシナリオを対応づけることでそれを表現できると考えた。

ここで、例を挙げる。

例 1: 人物 A と人物 B の登場

1. 戯曲

人物 A: (登場)
人物 A: (移動)
人物 B: (登場)
人物 A: こんにちは
人物 A: (消える)

2. 簡易的 QUIK プログラム

```
scene1 <= scene2; A(speak), A(del);;  
scene2 <= A(out), A(move); B(out);;
```

3. シーングラフ

ここでの人物 A はすべて共通のメディアで、人物の登場は人物 A、B ともに同じアクションである。また、A(out) と A(move)、A(speak) と A(del) の関係は非順序実行である。この例は分りやすくするた

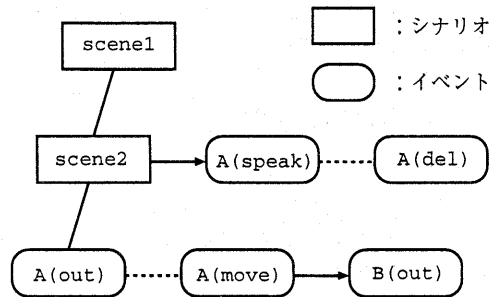


図 1: シーングラフのモデル

め、少し作為的にシナリオを分けたり実行順序を決めている。

□

2.4 時間の概念

ここでは、時間の概念について考える。本来、現実の世界での時間の概念は、まず絶対的な時間が存在し、その経過に沿って様々なことが起る。しかし、戯曲をバーチャルに再現しようとするとしても、処理や描画に要する時間を無視することはできない。つまり、絶対的な時間に沿って上演すると、マシンの状態やスペックによって動作が異なってしまう。そこで、我々は相対的な時間の概念を導入することにした。

QUIK II で取り扱うメディアに対して、メディア時間、再生時間、同期時間の以下の3種類の時間概念を考えている。

メディア時間 (media.time)

メディアそのものが持っている時間。例えば動画などの場合、始まって終るまでの時間を意味している。静止画や3Dオブジェクト、テキストの場合は、メディア時間ゼロまたは無限大と考える。

再生時間 (play.time)

実際にメディアを表示する時間。この時間によって、静止画等を表示させる時間を設定できる。

同期時間 (synchro.time)

処理するメディアが複数ある場合に、メディア間の同期をとるための相対的な時間である。

上記のような相対的な時間には、描画や通信の時間は含まれていない。絶対的な時間ではそれらの時

間も含まれてしまう危険性がある。例を挙げて具体的に時間について考える。

例 2: A, B, C が持つ時間

1. 意味

- A : 無限大時間表示しつづけて,
- B : k 時間後に m 時間表示し,
- C : l 時間後に n 時間表示する。

2. 簡易的 QUIK プログラム

```
scene <= A(out1); B(out2); C(out3);;
A(out1)/[media_time = "無限大",
  play_time = "無限大",
  synchro_time = k];;
B(out2)/[media_time = "無限大",
  play_time = m,
  synchro_time = l];;
C(out3)/[media_time = n,
  play_time = n,
  synchro_time = 0];;
```

この例を時間軸に沿って図で表すと図 2 のようになる。また、メディアの描画時間や通信時間は t_n とする。

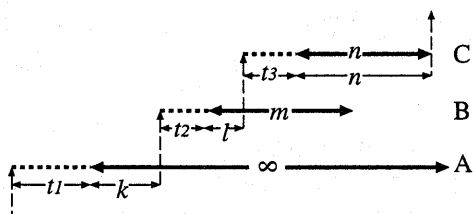


図 2: 時間の概念

さらにメディア間の依存関係に基づいて再生・同期時間が動的に変化することも考えられるが、ここでは導入していない。また、アクションの時間概念についても考慮できていない。しかし、移動における時間はプロット数を増減することで多少操作できる。

2.5 実行順序

メディアを制御する順序はすべて静的ではなく、一部動的に与えることができるようにした。この機

能によって、利用者が一部順序を曖昧に記述することができ、負担を軽減できる。

前述したようにシーングラフの同階層のエッジには実行順序が対応しており、隣り合うシーン間の関係は逐次実行か非順序実行のいずれかになっている。

逐次実行の場合は記述した順に沿って静的な順序で実行し、非順序実行の場合は記述した順に関係無く動的に順序を決定する。つまり、逐次実行のように実行する順序が明確に決定されていないため、順序を決定するためのアルゴリズムが必要になる。非順序実行の実行順序を決定するために重要なことの1つとしては、メディアとアクションの種類を認識して、その種類によって出力の順序を変更することが上げられる。

具体的には、各メディア・アクションが元々保持している自メディア・アクションの種類を判別するための type の情報を利用し、それをそのまま重みとして再利用した。シーンの最小単位であるイベントの場合はこの重みをそのまま利用し、シナリオの場合はそのシナリオが持つ子の平均を重みとした。

これにより、登場人物の画像が出力されていないにも関わらず、その人物が会話することに相当するテキストを発話するなどのような簡単な矛盾を解消することができる。

ここで例 1 を少し変更した場合の実行順序について説明する。

例 3: 人物 A 登場、発話の実行順序

1. 簡易的 QUIK プログラム

```
scene1 <= A(del), A(speak), scene2;;
scene2 <= A(move), A(out); B(out);;
```

QUIK プログラムではセミコロンは逐次実行、カンマは非順序実行を意味する。実行順序の処理をする前のモデルを図 3、実際に出力される順序を図 4 に示す。

例 1 と記述が異なっても同様の結果になることがわかる。また、A の登場、移動、消去の順は人間にとって容易に決定できる。つまり、明確なアルゴリズムがあり、それに従って順序の決定を行っているため、そういった部分は順序決定の処理にまかせることができるが、A と B のどちらが先に登場するかなどは、人間にとっても曖昧なため明記しなければ意図した結果と異なることがある。

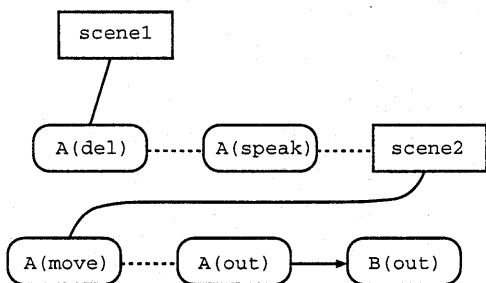


図 3: 実行順序処理前のモデル

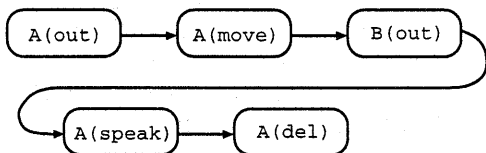


図 4: 実際の実行順序

2.6 関連研究との比較

QUIK をマルチメディア情報提示制御言語として拡張する際、関連したものとして SMIL[7], TVML[10], MHEG, HyTime[8] などがある。本システムは、前述した数種のメディアとアクションでは本質的に QUIK II の表現力が限られてしまい、以下にあげるような特徴的な機能すべてを QUIK II で実現するのは困難であるので、TVML や SMIL といった全く別の言語を外部メディアとして取り込むように設計してある。

SMIL は XML で記述されている言語で、3D オブジェクトは表示することはできない。また、編集はソースを直接変更する必要があるため本システムのような視覚的な編集はできない。TVML は既定の CG キャラクタが動き、カメラワークを指定できる。情報が膨大なため編集は難しい。MHEG はユーザー入力に対する応答動作を規定することができる。情報が符号化されているため直接編集はできない。HyTime は SGML にハイパーリンク機能を拡張した言語で、可読性はあるが要素名が多く記述されているため、一見しただけではストーリーが見えない。

QUIK II は視覚的かつ対話的な仮説演出を特徴としており、上記関連システムにはこの機能は無い。また、本システムの言語は論理プログラミングなの

で、戯曲の流れを容易に汲み取れる。上記関連システムを外部メディアとすることにより、それぞれの長所を包括的するようなシステムを目指している。

3 戯曲上演演出システムの構築

マルチメディア情報を同期的に制御できるよう拡張した QUIK II システムを応用し、戯曲上演・演出システムについて説明する。戯曲上演・演出システムの全体像を図 5 で示す。

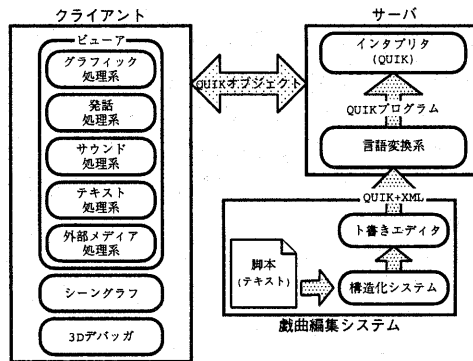


図 5: システムの全体像

本システムの特徴を以下に示す。

- 人的コストを下げるため、戯曲編集システムによってプレインテキスト→XML 文書→QUIK プログラムへの変換することができる。
- 後述するクライアントの処理系によって複数の視点からの同期的な提示ができる。ビューアで上演し、シーングラフウィンドウではモデル化したシーングラフを表示しているため物語の流れなどを視覚的に知ることができる。
- シーングラフウィンドウや 3D エディタによって複数の視点から視覚的な編集ができる。それぞれの機能の特徴はシーングラフウィンドウではシーンやメディア、アクションの交換や削除、追加など時間的な情報を視覚的に編集できる。3D エディタでは各イベントのメディアやアクションの詳細な情報を視覚的に編集できる。
- 入れ子トランザクションによって上記の演出を保存することなく行うことができる。この複数の視点から試行錯誤的に視覚的演出を行うことを仮説といい、本システムの最大の特徴である。

3.1 戯曲編集システム

戯曲編集システムは構造化システムとト書きエディタの2つの処理系から構成される。構造化システムはテキストで書かれた脚本から定められた文法にしたがってXML文書への構造化を行う。ト書きエディタは構造化システムで構造化された脚本のト書き部分を具体的なメディアに対応させるなどのために必要である。戯曲編集について例1を用いて説明する。

例 4: 人物 A と人物 B の登場

例1の場合で戯曲編集システムによってXML文書が生成された例を考える。図6は構造化システムによって自動的に生成されたXML文書である。図7はト書きエディタによって編集したXML文書である。

```
<story>
:
<dialog_part>
  <number>1-3</number>
  <actor>人物 B</actor>
  <direction>登場</direction>
</dialog_part>
<dialog_part>
  <number>1-4</number>
  <actor>人物 A</actor>
  <dialog>こんにちわ</dialog>
</dialog_part>
:
</story>
```

図 6: 自動生成された XML 文書

```
<story>
  <sinario>
    :
    <event>
      <a_pointer>
        <s_number>1.3</s_number>
      </a_pointer>
      <media>
        <m_name>人物 B</m_name>
        <m_type>1</m_type>
        <url>人物 B.jpg</url>
      </media>
      <action>
        <a_name>登場</a_name>
        <a_type>1</a_type>
        <point>
          <point_x>-3.0</point_x>
          <point_y>0.0</point_y>
          <point_z>0.0</point_z>
        </point>
      </action>
    </event>
    <sequence/>
  </sinario>
</story>
```

図 7: ト書きエディタ編集後の XML 文書

3.2 サーバ

サーバは以下の2つの処理系から構成される。

- 言語変換系

XML文書をQUIKプログラムに変換する。

- インタプリタ (QUIK)

言語変換系から得られたQUIKプログラムの実行順序に従って、各メディアをクライアント側に送信するエンジンである。

このインタプリタはオブジェクト管理系と実行系の2つの処理系から成っている。前者は、QUIKプログラムを内部表現に変換し、実行に必要なオブジェクト群をトランザクション機能によって管理する。後者は、オブジェクトやルールをクライアントからの問い合わせに対応して必要に応じ取り出し、解釈、実行する。

3.3 クライアント

クライアントはマルチメディア情報の同期的提示を大きく分けて3つの処理系で行う。

- ビューア

サーバから送られてきたメディアの表示、発話等の処理を行う。図8に示す。

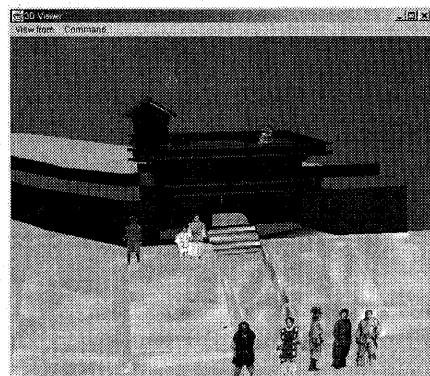


図 8: ビューア

- シーングラフウィンドウ

シーングラフは、シーンの実行順序を有向グラフでモデル化したものである。シーングラフウィンドウはそれの表示・操作ができる。それにより、利用者はビデオ編集のような感覚で対話的にマルチメディア情報の制御が行なえ戯曲を楽しむことができる。図9に示す。

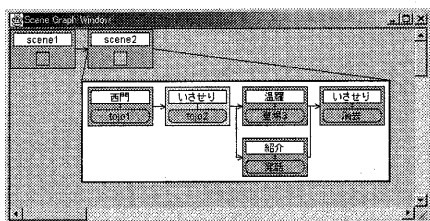


図 9: シーングラフウィンドウ

● 3D エディタ

テキストで書かれた脚本のト書きだけを元にして視覚化を行なうが、ト書きに含まれている詳細情報だけでは、どうしても取り出せない情報がある。例えば、キャラクタの立ち位置や移動情報などの情報を修正するための処理系、図 10 に示す。

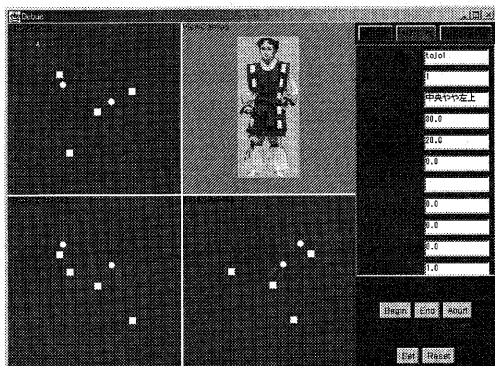


図 10: 3D エディタ

3.4 仮説機能の導入

マルチメディア情報提示システムの応用として戯曲の上演を行う際に、試行錯誤的に何度も演出をし、比較などをしたいという要望が挙げられる。そこで従来の QUIK (Quixote[1]) にある仮説の概念を導入することによって実現する。

仮説機能とは、従来の視覚的な編集を行い、その結果を保存する事なく閲覧でき、またアンドゥ機能によって以前の状態に戻したりする、試行錯誤的な演出のことである。この機能の実現のために以下に示す機能を導入した。

3.4.1 演出機能

演出は複数の視点から行える機能を実現した。また、ユーザの負担が軽くなるようにシーングラフウィンドウや 3D デバッガを用いて視覚的に与える (つまりユーザに QUIK プログラムを意識させない) 方法を考える。

その種類と与え方を以下のようにした。

● シーングラフウィンドウ

シーンやメディアの順番の入れ替えやメディア・アクションの追加・削除など主に物語の流れを変更することができる

● 3D エディタ

メディアのタイプや属性の変更又はアクション (移動情報など) の属性の変更などマルチメディアの詳細な情報の更新を行うことができる

3.4.2 データ管理

本システムは、対話的演出を行うことを特徴としているため、アンドゥ機能は必要である。そこで、クライアントによって仮説を生成した場合に生じる更新データの管理を入れ子トランザクションによって制御することを考えた。

つまり、仮説によるデータの変更が開始される (トランザクションが発生する) 度に一時的にデータを保存し、トランザクションコミット時に変更を保存するようにした。

ここで仮説を使用したときの概念図を図 11 に示す。

4 終りに

本稿では、対象とする情報源を英数字などの記号で記述されたオブジェクトだけでなく、マルチメディア情報を扱うための方法について議論してきた。また、マルチメディア情報提示の応用としてテキストで書かれた戯曲を同期的に視覚化し、それらの対話的な仮説を複数の視点から行う方法を議論し、実装した。

今後の課題としては、QUIK の特徴であるメディアエータの自動探索を用いた演出の実現、またそれとは正反対の意味で、より簡単なシステムでも戯曲の上演を行えるようにシステムのスタンドアローン化などがあげられる。また今後の QUIK II に対する要求としては以下にあげたものなどを検討している。

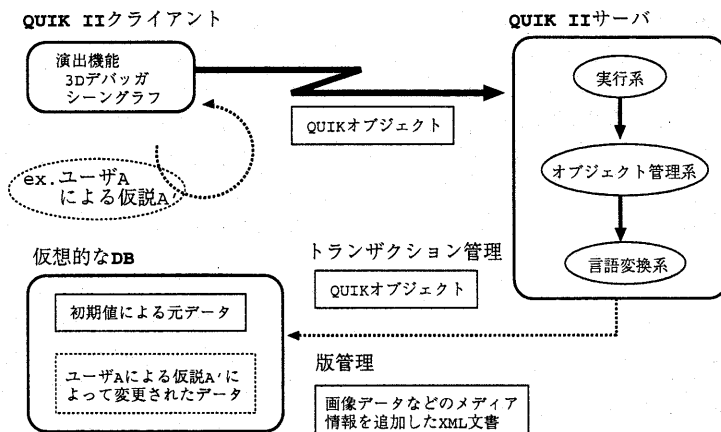


図 11: 仮説を使用したときの概念図

- 今回の脚本の提供元の総社市民劇団「温羅」などの専門家による評価
- メディア変更のためのサポートシステム
- 他の演出との比較機能
- ビューアのフリーソフト化
- Web ブラウザとの統合

これらの問題を解決していくことで、多種多様なメディアを統合し、柔軟で高度な演出、幅広い上演などより対話的にマルチメディアの提示・編集が行えるようになると思われる。

謝辞

さまざまな議論を頂く岡山県立大学横田研究室の皆様に感謝します。

参考文献

- [1] K.Yokota and H.Yasukawa, "Towards an Integrated KnowledgeBase Management System - Overview of R&D on Databases and Knowledge-Bases in the FGCS Project", FGCS Tokyo June 1-5,1992.
- [2] 杉本健二, 横川賢, 緒方啓孝, 國島文生, 横田一正, "QUIK システムにおける分散探索の制御", 電気・情報関連学会中国支部大会, 1998 年 10 月 24 日.
- [3] Bojiang Liu, Kazumasa Yokota, and Nobutaka Ogata, "Specific Features of the QUIK Mediator System", *IEICE Transactions on Information and Systems*, vol.82, no.1, pp.180-188, Jan., 1999.
- [4] 緒方啓孝, 野宮一生, 國島文生, 横田一正, "QUIK による戯曲の視覚化", 電気・情報関連学会中国支部大会, 1999 年 10 月 23 日.
- [5] 藤野猛士, 野宮一生, 横田一正, 國島文生, 三宅忠明, "構造化文書に基づいた対話的戯曲提示システムの実現", 情報処理学会データベースシステム研究会, 東京, 2000 年 5 月 25-26 日.
- [6] 杉本健二, 緒方啓孝, 的野晃整, 國島文生, 横田一正, "対話的マルチメディア情報提示システム実現のための QUIK の拡張", 情報処理学会データベースシステム研究会, 2000 年 5 月 25-26 日
- [7] Hoschka.P.(ed), "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification" (W3C Recommendation), 1998-06-15, (<http://www.w3.org/TR/1998/REC-smil-19980615>)
- [8] <http://www.hytime.org/>
- [9] <http://www.metamata.com/JavaCC/>
- [10] <http://www.strl.nhk.or.jp/TVML/> 林, 折原, 下田, 上田, 横山, 八重樫, 栗原, 安村, "テレビ番組制作言語 TVML とその言語仕様", 映像情報メディア学会冬季大会, 4-4, pp. 87, (1997)