Proceedings of the ISCA
18th International Conference

COMPUTERS AND THEIR APPLICATIONS

Honolulu, Hawaii USA
March 26-28, 2003

Editor: N. Debnath

A Publication of
The International Society for
Computers and Their Applications - ISCA

ISBN: 1-880843-46-3

The Proceedings of the ISCA 18th International Conference on Computers and Their Applications, held in Honolulu, Hawaii, USA, March 26-28, 2003.

SPONSOR

The International Society for Computers and Their Applications - ISCA

INTERNATIONAL PROGRAM COMMITTEE

ISCA, 975 Walnut Street, Suite 132, Cary, NC 27511 · Ph: (919) 467-5559 · Fax: (919) 467-3430
E-mail: isca@ipass.net WWW site: http://www.isca-hq.org

Proc. of ISCA 18th International Conference on Computers
and Their Applications (CATA 2003) (Mar. 2003)

A System for Presenting Background Scenes of Karaoke Using an Active Database System

Tsutomu TERADA
Cybermedia Center
Osaka University
tsutomu@cmc.osaka-u.ac.jp

Masahiko TSUKAMOTO
Grad. School of Information
Science and Technology
Osaka University
tuka@ist.osaka-u.ac.jp

Shojiro NISHIO
Grad. School of Information
Science and Technology
Osaka University
nishio@ist.osaka-u.ac.jp

Abstract

In recent years, it has become popular to use multimedia data for building presentations. In cases of real-time contents are built, a powerful mechanism is necessary for extracting materials suited to the situation from a large amount of multimedia data. In this study, as an example of such a mechanism, we constructed the Active Karaoke system, which generates background scenes for karaoke using an active database. An active database is a database system that processes prescribed actions automatically in response to the occurrence of an event. We show that Active Karaoke can extract appropriate multimedia data from the database in response to the change of music tones and contents of lyrics, and display them as background scenes. Since an active database is used for storing and managing data, the system behavior can be easily and flexibly described using ECA rules, which are the behavior description language of an active database.

1 Introduction

In recent years, the evolution of multimedia technologies has focused a lot of attention on techniques for using multimedia contents in databases for real time presentations. A real time presentation is one that shows materials extracted in real time from a database in response to a change of situations without prescribed scenarios (Figure 1). Since situations come under the influences of a system state and external environments, a real-time presentation-generating system must extract appropriate materials from databases in response to a change of situations. In this study, we pick up karaoke as an example of real time presentation systems. Karaoke is the act of amplified singing to taped accompaniment. It is a popular form of entertainment in Japan, and also gaining popularity around the world.

Although singing songs is a prime purpose of karaoke, it is also a good way to communicate with other people. It is this reason that personal karaoke systems are not widespread, despite high popularity of karaoke rooms[4]. Consequently, karaoke systems have been developing with various technologies to make singing exciting.

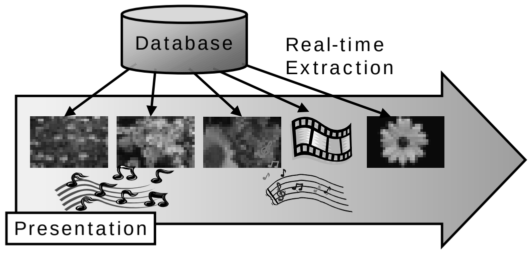


Figure 1: Real-time Presentation System

A karaoke system is composed of a singer's voice, background music, a background video with lyrics and other miscellaneous elements. Many functions to excite singers have been already realized on commercial karaoke systems. For example, with respect to voices, the *"virtual duet" function with only one singer* plus a voice effect function makes the singer excited. In regards to background music (BGM), the user can sing comfortably with the automatic transforming function to adjust the user's voice to the exact pitch. In terms of other functions, *DAM-DDR* is a function where a user can dance while singing in accordance with instructions displayed on the screen of the karaoke system. *Calorie Karaoke* is a function that displays calories consumed by singing, and so on[1].

On the other hand, most of back ground videos for karaoke play nothing but a video movie that is related to the song to some degree. Hence, the goal of our study is to develop a system for generating background scenes of karaoke as an example of real time presentation generat-

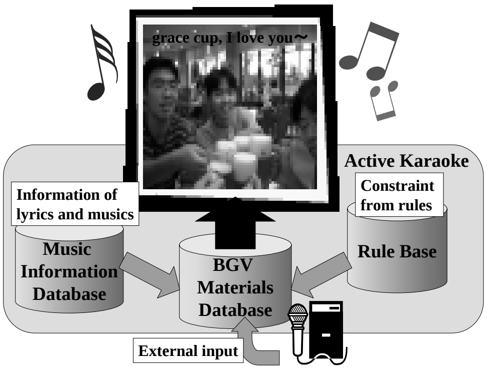


Figure 2: The System Image of the Active Karaoke

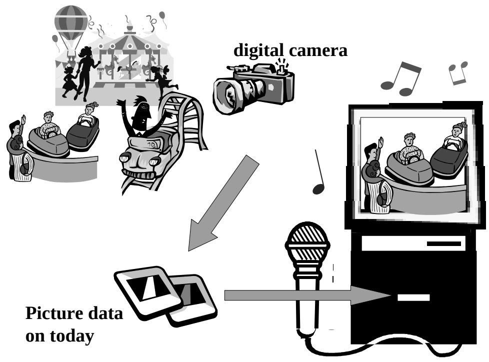


Figure 3: Practical model 1

ing systems. This system generates background scenes dynamically in response to changes of situations.

The remainder of this paper is organized as follows. Section 2 outlines the Active Karaoke. Section 3 explains an active database that forms the base technology of our system, and Section 4 illustrates an implementation of Active Karaoke. Section 5 shows related works, and Section 6 presents conclusions and future works.

2 Active Karaoke

Active Karaoke generates and shows background scenes of karaoke suited to the BGM. This system extracts scenes from databases dynamically and automatically in response to changes in lyrics, melody parts and the environmental mood. The system image is illustrated in Figure 2. In the figure, the BGV (Back Ground Video) materials are a database to store materials for karaoke background scenes. Each data has some information, i.e., key words, file size, etc. The Music Information Database is a database for storing music data, consisting of lyric data, lyric display timing, music genre and others. The Rule Base has rules for deciding what scenes are displayed and what effects are given.

While playing a song, Active Karaoke continues to retrieve and display materials automatically in real time by exploiting information stored in the Music Information Database, the external input, and rules in the Rule Base.

Active Karaoke has the following characteristics:

- **Scene selection keeping up with the progress of the playing music**: Active Karaoke extracts a suitable scene from the database in response to changes in lyrics, melodies and the other environmental mood.
- **Various visual effects**: The system can apply various visual effects to displaying images for exciting karaoke. For example, if a music playing is a *slow ballad*, the system displays scenes with color tones changed to soft colors.
- **Utilization of personal multimedia data**: Users can easily register BGV materials such as private photos. We expect the system to attract a lot of interest with these personal materials as background scenes.

The following models are examples of practical uses of Active Karaoke:

1. **Enjoying Active Karaoke with photos taken at the place where users are staying (Figure 3)**: A user and friends go on holiday to an amusement park with digital cameras and take many pictures there. When users enjoy Active Karaoke in a karaoke room on the way home, pictures that ware taken by the users can be used as background scenes by transferring photo data to the karaoke system. If the digital camera has a function for attaching keywords to pictures, our system can show the picture at applicable times in consideration of the attached information.
2. **Enjoying Active Karaoke with prepared pictures on a network (Figure 4)**: A user prepares pictures with keywords, and uploads these pictures to a WWW site in advance. The user enjoys karaoke with prepared pictures as background scenes by entering the URL of these pictures at the karaoke house. If a user prepares several sets of picture data on Internet, he can choose a picture theme by changing the entered URL.

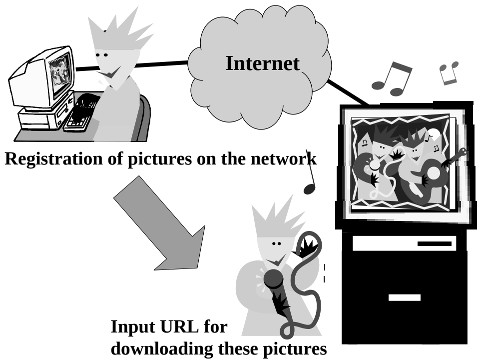


Figure 4: Practical model 2

```
CREATE RULE  Rule name ON  Event name
      [  Definition of variables ]
      [ WHERE  Conditions ]
      THEN DO  Actions
```

Figure 5: Syntax of the ECA rule

Table 1: Events provided by Active Karaoke

Name	Content
SELECT	Data retrieval
INSERT	Data insertion
DELETE	Data deletion
UPDATE	Data update
MUSIC_START	Music starts playing
MUSIC_STOP	Music stops playing
MUSICTYPE_CHANGE	Change of music type
MUSICPART_CHANGE	Change of melody part
FIND_SPECIFIC_WORD	A specific word appears in lyrics
INPUTVOLUME_CHANGE	Change of input volume
RECEIVE	Input from external equipment

3 Active Database

The most important function of Active Karaoke is to carry out various actions in response to changing situations. For example, Active Karaoke has a function for applying effects to a scene in response to changes in lyrics, and to change scenes in response to an external voice input. Especially in the latter case, since an external input is the accidental matter for the system, it is difficult to predict these events. As we mentioned in Section 1, since karaoke systems have been continuously developing, it is necessary that the system is adaptable.

Thus, we constructed Active Karaoke using active database technologies. An active database is a database that processes the prescribed actions in response to the occurrence of an event arising on inside/outside the database[5]. Behaviors of an active database are described by ECA rules. Each ECA rule consists of three parts: Event, Condition, and Action. The Event part describes the occurring event in the system. The Condition part describes the conditions for executing actions, while the Action part describes the operations to be carried out.

Using ECA rules for describing the system functions, we can describe system behaviors in an event-driven manner. Moreover, we can customize system functions by adding/deleting ECA rules to/from the system.

3.1 ECA Rule

In general, conventional active databases are only capable of detecting the events that occur within the database, such as SELECT, INSERT, DELETE, and UPDATE. Correspondingly, active databases can carry out only database operations as actions. In our research, we enhanced the description capability of ECA rules to fulfill the various requirements for karaoke systems. Figure 5 shows the syntax of the ECA rule used for Active Karaoke.

In the figure, the *Event name* describes a name of an event that triggers this rule. The *Definition of variables* defines the local variables used within this rule, while the *Conditions* specifies the condition for executing the following actions. This part is described by the sequence of *<Expression 1>* *<operator>* *<Expression 2>*. *Expressions 1, 2* describe database attributes or constant values, and the *operator* can be '=', '>', '<', '>=', '<=', or '! ='. The *Action* describes operations to be carried out if all conditions are satisfied.

Table 1 shows the events provided by Active Karaoke. The *SELECT, INSERT, DELETE, UPDATE* are events activated when the system retrieves, inserts, deletes, updates data on the database of Active Karaoke. These events are used for ECA rules such as 'When new materials are stored in the database, these materials are displayed at a specific point in the music'. The *MUSIC_START*, *MUSIC_STOP* events are activated when music starts/stops playing. The *MUSICTYPE_CHANGE, MUSICPART_CHANGE* events are activated on the change of the genre (rock, pops..) and the part (bridge, theme) of the song that is playing. The *FIND_SPECIFIC_WORD* event is activated when specific words appear in the lyrics. The *INPUTVOLUME_CHANGE* is an event activated by a significant change of the input volume from microphones. This event is used for ECA rules such as 'When the singer is excited, the system displays specific images'.

Table 2 shows the actions provided by Active Karaoke. The *QUERY* action issues a database query for management of the database in response to the change of a situation. The *INSERT_ECA*, *DELETE_ECA* actions insert/delete ECA

Table 2: Actions provided by Active Karaoke

Name	Content
QUERY([expression])	Database operation
INSERT_ECA([description])	Storing an ECA rule
DELETE_ECA([identification])	Deletion of ECA rules
ENABLE_ECA([identification])	Activation of ECA rules
DISABLE_ECA([identification])	Deactivation of ECA rules
APPLY_EFFECT([identification])	Applying visual effects
DISPLAY_GRAPHIC([identification])	Displaying specific images

Table 3: Visual Effects

Name	Type	Content
POS_NORMAL	POS	Normal position (full screen)
CHILD	POS	Dual window (full and small windows)
RANDOM	POS	Displaying 1/4 size random images at four corners by rotation
SMALL	POS	Displaying a 1/4 size image at a random position
EFF_NORMAL	VIS	No visual effect
EFF_GRAY	VIS	Gray scaling
EFF_SEPIA	VIS	Change the color tone of images to sepia tone
EFF_FLASH	VIS	Flash effect
DISP_NORMAL	DISP	No displaying effect
SLIDEIN_LEFT	DISP	Slide-in effect from left side. Same applies to right, top and bottom.
WIPEIN_LEFT	DISP	Wipe-in effect from left side. Same applies to right, top and bottom.
CENTER_ZOOM	DISP	Zoom-in effect from center.

POS: Position VIS: Visual effect DISP: Displaying effect

rules into/from the system. Using these actions, the system can change its functions dynamically. The *ENABLE_ECA*, *DISABLE_ECA* actions enable/disable ECA rules for customizing system behavior without deleting any of the rules. The *APPLY_EFFECT* action applies various visual effects to to the displaying images. Table 3 shows the visual effects that can be used in our system, which can be categorized into three types: position, visual effect, and displaying effect. This action specifies which effects to apply combining effects from each category. Moreover, this action assigns the applicable scope of the effect shown in Table 4. The *DISPLAY_GRAPHIC* action displays specific images. This action also specifies the period of time the image is to be displayed by same method as the *APPLY_EFFECT* action.

Additionally, Active Karaoke provides two system parameters, *NEW data* and *OLD data* for each event. The system sets the information shown in Table 5 to these parameters when the event is activated. These parameters can be used in the condition part or the action part of ECA rules.

Figure 6 shows some examples of ECA rules. In this figure, *RULE1* converts scenes to sepia colors for creating mood. This rule is activated if the genre of the playing music is "Ballad". *RULE2* and *RULE3* are rules that fulfill the demand for special scenes at the climax of a song. *RULE2* initiates the use of a specific image as the background when the music part becomes "Climax," and *RULE3* stops the effect of *RULE2* at the end of the "Climax." *RULE4* is an example of processing in response to external equipment. If the system receives

Table 4: Specification of Applicable Scope

Specification	Content
ALL	Whole span of a song
RANGE([start], [end])	Fixed period of time
UPTO([flag specification])	By the time a specific flag becomes on
ONLY_ONE	Only next single image

Table 5: Contents of NEW data and OLD data

Event	NEW	OLD
SELECT	Retrieved data	–
INSERT	Inserted data	–
DELETE	–	Deleted data
UPDATE	Data after update	Data before update
MUSIC_START	Song information	–
MUSIC_STOP	–	Song information
MUSICTYPE_CHANGE	New type	Old type
MUSICPART_CHANGE	New part	Old part
FIND_SPECIFIC_WORD	position of the word	–
VOLUME_CHANGE	New volume	Old volume
RECEIVE	Received data	–

data from an external equipment named "Maraca," this rule displays a specific image for one second. *RULE5* and *RULE6* are rules for displaying random quarter-sized scenes during musical introductions or interludes because these musical parts have no lyrics as keywords for diciding which scenes to display. *RULE7* is an example of complex effects. When music part becomes a "Bridge," this rule displays a specific image for five seconds with two effects: the *flash effect* and the *zoom-in from center effect*.

3.2 System Structure

Figure 7 illustrates the system structure of Active Karaoke. The *Active Database Part* realizes functions of active database operations such as ECA rules processing and scene retrieval. The *Event detection mechanism* detects a situation change inside/outside the system. The *ECA rule retrieving mechanism* retrieves ECA rules for an activated event from an ECA rule base. If matched rules exist, The *ECA rule processing mechanism* executes them.

The *Interface Part* provides graphical user interfaces to select, play and stop songs.

4 Implementation

We implemented a prototype Active Karaoke system using *Microsoft Visual C++ 6.0* on a *DynaBookSS3380 (Mobile PentiumII 400Mhz, 128MByte memory)*. In addition, about 20 set of music data are stored in the Music database, and we use MP3 and MIDI as the format of storing music data. We also stored approximately 1000 scenes in BGV Material database, where each scene is a photo image in the JPEG format. Figure 8 shows snapshots of the system with ECA rules

```
CREATE RULE  RULE1 ON MUSICTYPE_CHANGE
WHERE NEW.TYPE = 'Ballad'
THEN DO
  APPLY_EFFECT( ALL, EFF_SEPIA, SYS_NOWSONG );
```

```
CREATE RULE  RULE2 ON MUSICPART_CHANGE
WHERE NEW.TYPE = 'Climax'
THEN DO
  DISPLAY_GRAPHIC( UPTO(FLAG(0)),
                'C:/graphics/climax.jpg');

CREATE RULE  RULE3 ON MUSICPART_CHANGE
WHERE
  OLD.TYPE = "Climax"
THEN DO
  FLAG_ON(0);
```

```
CREATE RULE  RULE4 ON RECEIVE
WHERE
  NEW.TYPE = "maraca"
THEN DO
  DISPLAY_GRAPHIC( RANGE(SYS_NOWTIMING,1),
                  'C:/graphics/maracas.jpg' );
```

```
CREATE RULE  RULE5 ON MUSICPART_CHANGE
WHERE
  NEW.TYPE = "Interlude"
THEN DO
  APPLY_EFFECT( UPTO(FLAG(0)),
          SMALL, SYS_NOWSONG );

CREATE RULE  RULE6 ON MUSICPART_CHANGE
WHERE
  OLD.TYPE = "Interlude"
THEN DO
  FLAG_ON( 0 );
```

```
CREATE RULE  RULE7 ON MUSICPART_CHANGE
WHERE
  NEW.TYPE = 'Bridge'
THEN DO
  DISPLAY_GRAPHIC( RANGE(SYS_NOWTIMING,5),
                    'C:/climax.jpg' );
  APPLY_EFFECT( RANGE(SYS_NOWTIMING,5),
                EFF_FLASH, SYS_NOWSONG );
  APPLY_EFFECT( RANGE(SYS_NOWTIMING,5),
               CENTER_ZOOM, SYS_NOWSONG );
```

Figure 6: Examples of ECA rules

shown in Figure 6. Snapshot 1 is a snapshot displayed at the opening of a song. RULE5 orders the display of quarter-sized random images transformed into sepia color by RULE1. In Snapshot 2, since there is no rule activated in this situation, the system displays an image which is suited to the lyrics (in this snapshot, the image is extracted by a keyword 'Summer'). Snapshot 3 is a snapshot during the bridge of a song. In this situation, RULE7 displays a specific image (climax.jpg) with some effects (FLASH, CENTER_ZOOM). In Snapshot 4, RULE4 displays a specific image (maracas.jpg) because the system receives a data packet from an external maraca device.

With regards to the throughput of our system, the scene retrieval process by keywords becomes the bottleneck. This retrieval process consists of two steps: In the first step, the system produces related keywords from a keyword using the related-term dictionary. It then retrieves scenes that are appropriate to these keywords. In this research, to resolve this bottleneck, the system preassigns keywords with related keywords beforehand, and stores information in the *keyword to keywords database*. The bottleneck is broken off to some degree by using this method. However, the system has to remap each time it updates the related-term database. We evaluated the performance of our system with about 1000 scenes. As a result, the retrieval time of each scene was less than 0.1 seconds.

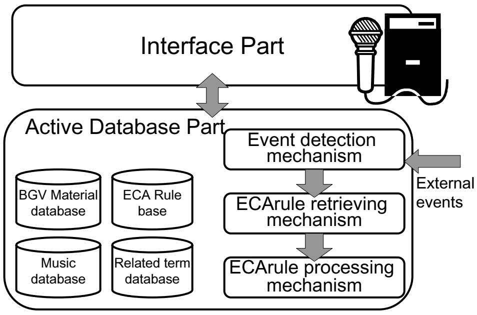


Figure 7: The structure of Active Karaoke

The validity of retrieved results depends on keywords attached to each scene and the related term dictionary. In our prototype system, we use simple keyword matching as the retrieving method, therefore, the system cannot always retrieve the best image when considering its meanings. We can improve the method of retrieval using other methods. However, it may be interesting when the system displays unexpected types of images. Consequently, we have to enhance the retrieving method without losing the element of surprise.

5 Related Works

To take example studies of enhanced active database systems, there are research works on AMDS (Active Mobile Database System)[2], SADB (Super Active Database System)[3]. AMDS and SADB are the enhanced active database systems for mobile computing environments and broadcasting environments. These systems have to process emergent events such as connection/disconnection of the mobile host in AMDS, receiving broadcast data in SADB. The concept of active databases is effective in such environments. Accordingly, Active Karaoke has to manage various emergent events such as changes in musical parts or voice inputs by a singer. Therefore, it is also effective to develop our system based on active database technologies.

6 Conclusion

In this paper, we proposed Active Karaoke, which is a system for generating background scenes for karaoke using active database technologies. Active Karaoke has various functions for enhancing the enjoyment of karaoke.

In the future, we will propose an advanced mechanism for scene retrieval using new ideas such as retrieval that considers previous scenes. Moreover, we will provide more events and actions in the ECA rule format for more visual effects and advanced retrieval methods.

Acknowledgements

This research was supported in part by the Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology, Japan and Grant-in-Aids for Scientific Research numbered 13780331 from Japan Society for the Promotion of Science.

References

[1] DKKaraoke Homepage: http://www.dkkaraoke.co.jp/.

[2] T. Murase, M. Tsukamoto, and S. Nishio: "A System Platform for Mobile Computing based on an Active Database," *in Proc. International Symposium on Cooperative Database Systems for Advanced Applications*, vol. 2, pp. 424–427 (1996).

[3] S. Sanguantrakul, T. Terada, M. Tsukamoto, S. Nishio, K. Miura, S. Matsuura, T. Imanaka: "A User Customized Selection and Categorization for Broadcast Data," *in Proc. of 1999 IEEE International Workshops on Multimedia Network Systems(MMNS)*, pp. 596–601 (1999).

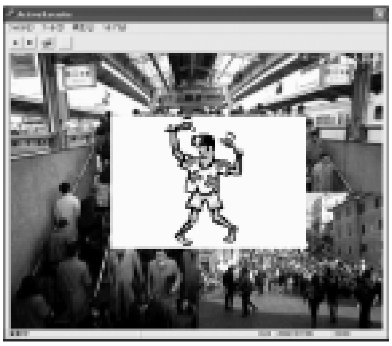
[4] All-Japan Karaoke Industrialist Association Homepage: http://www.japan-karaoke.com/.

[5] J. Widom, R.J. Cochrane, and B.G. Lindsay: "Implementing Set-oriented Production Rules as an Extension to Starburst," *in Proc. of the Seventeenth International Conference on Very Large Data Bases*, pp. 275–285 (1991).

Snapshot 1: Opening

Snapshot 2: Routine Play

Snapshot 3: Bridge



Snapshot 4: Maraca Input

Figure 8: An example of Rule Executions