Keywords-based Automatic Multimedia Authoring in the Cloud

Abdelkader Outtagarts, Sylvain Squedin and Olivier Martinot
Alcatel-Lucent Bell Labs, Route de Villejust, Nozay, 91620, France

Keywords: Automatic Video Editing, Authoring, Mashups, Speech2text, Annotation, Reasoning, Collaboration, Web 2.0.

Abstract: In this paper, we describe our work on automatic multimedia authoring or editing. Our approach is based on keyword extracted from the audio embedded in videos. A model approach of mashup based on keywords is proposed. A video editing testbed has been designed and implemented. Using speech2text keywords generator component, the audio files uploaded in the video editing testbed are transcribed and analyzed in order to extract keywords. The keywords are used to edit automatically videos in order to produce mashups.

1 INTRODUCTION

Video editing is the process of selecting segments from a set of raw videos and chaining them by adding piece of audio, effects and transition in order to create a new video or mashup. Most available video editing tools are time-consuming and require specific knowledge. There is therefore, an increasing demand for simple, user-friendly online video editing tools. Online video editing tools can afford less costly and more accessible processes that can be completed anytime, anywhere. In this context, this paper presents a new tool to facilitate the creation and handling of video content. What our video editor wants to enable is a keyword-based automatic video editing deployed in the cloud with a previewing web client. Videos are coming from video sharing platforms, directly from the webcam, uploaded video files by users or captured from camera network. They can be composed and viewed on the fly. Those compositions, called video mashups can be shared and modified collaboratively, and their privacy easily is managed. This video editor is also a research testbed for studying automatic video editing and summarization based on text and statistical data and metadata. The paper is organised as follows. In the section 2, we present the video editing model. The video editing testbed is detailed in the section 3. We'll present the current status of our research the section 4. The model and algorithms validation methods are described in the section 5. Conclusions are drawn in Section 6.

Automated video editing and summarisation have gained momentum in recent years. Many algorithms such as shot detection are developed to extract representative key-frames from video. The majority of authors use video analysis or signal processing in order to perform automatic video composition (mashup). Among the authors, Hua et al. (Hua and Zhang, 2009) develop an approach for extracting temporal structure and determining the importance of a video segment in order to facilitate the selection of highlight segments. They also extract temporal structure, beats and tempos from the incidental music. In order to create more professional-looking results, the selected highlight segments satisfy a set of editing rules and are matched to the content of the incidental music. The strategy of (Hua and Zhang, 2009) is to parse the video sequence into hierarchical structures consisting of scenes, shots, and sub-shots. For music, they segment it into clips by strong beats, and for each clip, tempo is estimated, which indicates the speed of the music sub-clips. In their system, the video segment selection is also based the work proposed by Ma et al. (2002). Hua et al. (2009) refine the method by adding an "attention fusion" function, which generates improved results. Müller Arisona et al. (2008) present Odessa framework, which automates the preparation of footage during composition and the editing process. Real-time audio analysis extracts music feature vectors, which are mapped to editing parameters of a video montage engine. The engine can be controlled through high-level editing parameters, such as looping speed or cutting rate. The authors apply a reverse editing of a video. The original video is analysed and re-organised in terms of individual

Outtagarts A., Squedin S. and Martimot O..
Keywords-based Automatic Multimedia Authoring in the Cloud.
DOI: 10.5220/0004073701410147
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications and Wireless Information Networks and Systems* (SIGMAP-2012), pages 141-147
ISBN: 978-989-8565-25-9

scenes, groups, and shots. Odessa implements methods such as audio beat tracking or music similarity analysis to operate in real-time. Takemae et al. (2005) propose an automatic video editing system using stereo-based head tracking in multiparty conversations for conveying the contents of the conversations to the viewers. Their system can automatically detect participants' head 3D position and orientation during a conversation. Based on the detection results, the author's system selects the shot of the participant that most participants' heads are facing. This approach exploits participants gaze behaviour to select the most effective shots of participants. Mudhwuchutyula et al. (2004) propose an automatic mechanism for XML based video metadata editing, in tandem with video editing operations. An implementation framework for editing metadata in accordance with the video editing operations is demonstrated. The video metadata editing mechanism has been implemented in the context of Digital Video Album (DVA) system for editing metadata in accordance with presentation/ summarization operations performed on video. Foote et al. (2002) have presented a music video creation system that can automatically select and align video segments to music. Authors have analyzed both audio and video sequences in order to automatically create music videos for a given sound track.

As a conclusion of the state of the art, existing video editing solutions are time consuming. To facilitate video edition, many authors have worked in automatic video editing and summarization. The most algorithms developed to extract representative key-frames from video are based on video and/or audio analysis. These solutions require significant processing especially when it must deal with hundreds or thousands of videos.

2 MODEL

A video mashup is a combination of video and audio sequences, effects and transitions. The manual video editing is recommended when a user is working with a small number of media. When the number of media increases, it is necessary to help users by managing automatically the abundance of media. In the example of mashup shown in the figure 1, a video mashup consists of a set of video clips, transitions, titles... The videos are trimed to select only the interested segments or clips to build a small movie or a video mashup. After the trims, the selected clip will correspond to a time Δt_k as shown in the figure 1. The position (start attribut) of the video clip in the mashup and attribute values (length, trimstart, trimend, metadata positions in the mashup,…) are stored in a database using XML format.

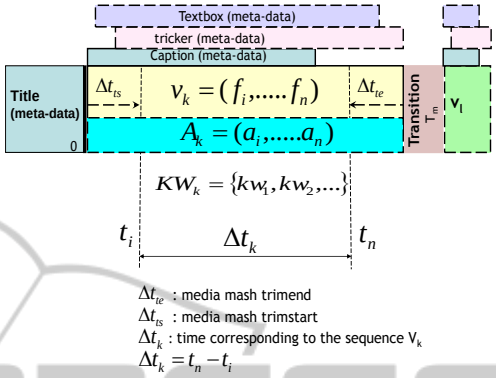

Figure 1: Detailed video mashup.

A video is a sequence of video frames captured by a webcam, phones or other devices during a certain time. A video v_k containing (n-i) frames during the time $t_i<t<t_n$, is given by (Figure 1):

$$v_k = (f_i, ... f_n) \quad (1)$$

The corresponding audio sequence contained in the video is represented by:

$$A_k = (a_i, ... a_n) \quad (2)$$

A_k is an audio sequence containing (n-i) number of audio frames.

Between the times t_i and t_n, the extracted keywords from the corresponding audio of video file is given by the following expression:

$$vKW_k = \{vkw_1, vkw_2, ...\} \quad (3)$$

vKW_k is the keyword list extracted during between t_i and t_n of the video V_k

The video V_k can be represented with both video frames, audio frames and metadata (keywords).

$$V_k = \begin{bmatrix} (f_i, ..., f_n) \\ (a_i, ..., a_n) \\ \{kw_1, kw_2, ...\} \end{bmatrix} \quad (4)$$

A video transition is the visual movements as one picture or video clip inserted between two video sequences. It is the way in which video sequences/shots are joined together. Preferably, a transition must be related to the context of the

created mashup video. We can represent a transition with a matrix containing a suite of video frames and keywords/tags which describes the transition.

$$T_m = \begin{bmatrix} (f_{n+1},\ldots, f_{n+m}) \\ \\ \{tkw_1, tkw_2, \ldots\} \end{bmatrix} \quad (5)$$

Where T_m is a transition and tkw_i keywords/tags corresponding to the transition T_m.

A Mashup M is basically represented by a sequence of non-overlapping sequences from a multiple video:

$$M = \sum_{j=0}^{j=x} V_j \quad (6)$$

x is the number of video sequences. Transitions can be added between video sequences to join them.

A mashup when encoded become a video and identified by a suite of frames of images, audios and keywords.

$$M = \begin{bmatrix} \sum_j f_j \\ \sum_j a_j \\ \{kw_1, kw_2, \ldots\} \end{bmatrix} \quad (7)$$

The challenges in this work is how, from one or more keywords, we can automatically selecting segments/sequences/shots from a set of raw videos and chaining them by adding transition in order to create a new video or mashup.

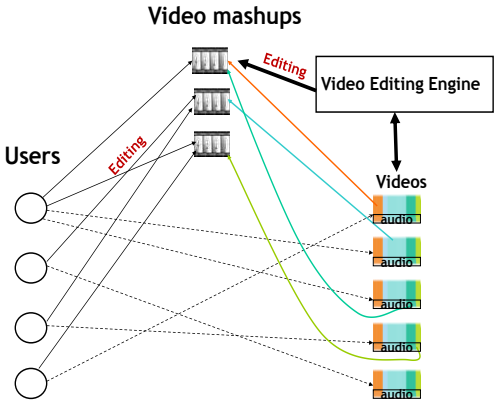

Figure 2: Video mashups.

Automatic video editing helps to manage the abundance of the media by proposing mashups for users. As showed in the figure 2, the feedback of users is necessary to validate the mashups.

We consider two possible users feedbacks which can be introduced in the automatic video editing algorithms:

Implicit: This user feedback is characterized by the number of times the video is viewed as well as modifications done in the proposed mashup.

Explicit: Here the user gives his opinion directly or delete the proposed mashup. These parameters can be added to the model in order to automatically taking into account the user feedback by using machine learning algorithms.

3 VIDEO EDITING TESTBED

3.1 Description

The video editor testbed platform allows online and collaborative video edition. This testbed is composed on six main components: a web-based video editing application, import multimedia, mashup browser and media manager, a database, an XML2Video converter and an automatic video editor engine:

- **Video Editing Application:** It is a web-based video editor flash applet which has not need pre-installation of editing software in client's devices. It allows online video editing or video mashup creation and annotation. Using the video editing application user has possibilities to modify videos using function such as: drag and drop media, trim, cut, adding audio, image, effects and transitions… All the mashups and text data are stored in Mysql database using XML format. The XML descriptor contains metadata such as : trim start/end attributes of videos and audios, videos and mashups length, the beginning and the end of a title, a caption, a textbox…
- **Mashup Browser:** this function allows displaying the private and public video mashup, the contributors for each mashup, links to edit the composition and to export the composition into the streaming server.
- **Mysql Database:** Two types of data are stored in the video editor testbed database: Critical data and statistical data. The critical data include: users, the media used to create compositions and compositions created by users. The statistical data are collected and processed to perform automatic annotation of the media. This data include: specific changes to a composition for each edition, How a user edits in compositions, metadata added by users and the data associated with imported videos.

- **Import Multimedia Features:**
  - ✓ *Import from Tivizio*: Using REST APIs, a user can collect videos from Tivizio, an enterprise video sharing platform.
  - ✓ *Webcam Video Recorder*: this function allows user to record a pitch in order to create a video mashup using other media and metadata.
  - ✓ *Import from camera network*: using Camera Search Engine component, a user navigate in camera network to collect images which are imported into the video editing application using REST APIs in order to be part of a video composition.
  - ✓ *Upload file from the device*: from PC or Android OS devices, a user is able to upload media with standard format.
- **XML2Video Converter:** this component allows converting XML descriptor of a video composition generated by the video editing application component into a file video format. This feature allows also exporting the video to Tivizio or Youtube.
- **Automatic Video Editor Engine:** this function allows for instance, automatic video edition based on keyword and rule.
- **Video Editing Testbed APIs:** REST APIs are provided for uploading videos, setting/getting metadata and composing mashup using keyword.

3.2 Data and Algorithm

Our algorithm of automatic video editing is based on keywords extracted from audio file of the video and other parameters of the media. A media can be a video, an image, an audio file or a mashup. As described in table 1 which shows the media table, the main attribute of a media used the automatic video editing algorithm are:

- The type
- The Meta data
- The mashup descriptor
- The media viewed counter

When uploaded in the video editing testbed, the video frames and audio file are extracted and stored the multimedia database. Other data are stored in MySql database in the media containing attributes described in the table 1.

Two types of data are stored in the database:

a- Critical Data

The first type of data is required to operate the video editing testbed:

- Users,
- The media used to create compositions,
- Compositions created by users.

b- Statistical

The second type of data stored in the database is linked to the use of the video editor. This data is used to compile statistics. Indeed, it is expected in this research project using all data collected to create an engine for implicit automatic annotation of videos. This is roughly able to understand "who publishes what and how»:

- Specific changes to a composition for each edition
- How user edits compositions
- Metadata added by users
- The data associated with imported videos

Table 1: Media table description.

Media	
m_id	Media id
u_user	Username
m_type	video, image, audio, mash
m_label	Media name
m_metadata	Keywords metadata Ex : <speech2keywords begin='00.00.16' end='00.01.15' keywords='service provider | quality service | right combination | irrational |'> </speech2keywords>
m_duration_seconds	duration
m_ source_url	Media url
m_date	Creation date
m_xml_string	Mashup XML descriptor
m_last_update	Update the date
m_view_count	Viewed counter
m_fps	Number of frame per seconds
m_audio	url of the audio file

4 CURRENT STATUS OF OUR RESEARCH

The post-production video editing requires significant processing, when we have to analyze a large number of media, hundreds or thousands of videos, audios and images. That is why we propose a solution based only on text data and metadata. These data and metadata are obtained in three ways:

- Classical video annotation by users,
- Implicit annotation video by inheritance of the text data added by users during the compositions. This annotation approach is described below.

- Extracting the text from the audio: speach2keyword during the media upload process.

The annotations collected by each media are analyzed using text processing and statistical algorithms for improving the media description. The automatic video algorithms use these text data in order to generate an XML descriptor as an output of video composition. An example of the XML mashup descriptor is given in the figure 3. Each segment of a clip is represented as a link to the original one, which maintains associated data in the case of being deleted or moved, as well as some more descriptions. The transitions and effects are either a link with descriptions and media links.

```
< descriptor >
<mashup width="320" height="240" id="c8ddafe1233b29125e4f69be8123f763"
                label="XML descriptor example>
   <clip type="video" id="5eef274620a55a78b6c90a845cfcf307" begin="100"
end="200"
         label=http://www.youtube.com/watch?v=HlxIWmA4pMk" length="100"
         description= " Shot description …. " />
   < clip type="video" id="jgjg274620a55a78b6c90a845dfhf67" begin="0"
end="100"
         label=http://www.youtube.com/watch?v=_ul_4kJCy9E" length="100"
         description= " Shot description …. " />
  <media group="video" type="video" id="5eef274620a55a78b6c90a845cfcf307"
         label="http://www.youtube.com/watch?v=HlxIWmA4pMk"
         source=" http://www.youtube.com/watch?v=HlxIWmA4pMk"
         fps="10" duration="1000"
          description= " General description …. " />
   <media group="video" type="video" id=" jgjg274620a55a78b6c90a845dfhf67 "
         label="http://www.youtube.com/watch?v=_ul_4kJCy9E"
         source=" http://www.youtube.com/watch?v=_ul_4kJCy9E"
         fps="10" duration="800"
          description= " General description …. " />
</ mashup>
</descriptor >
```

Figure 3: Example of XML mashup descriptor.

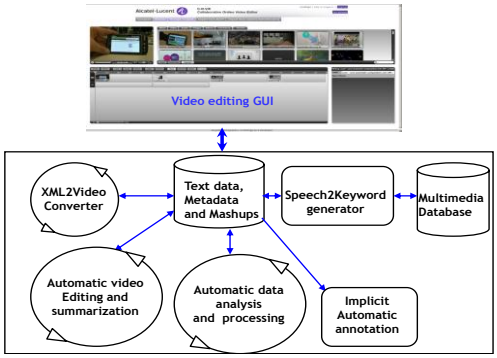

Figure 4: Global architecture of the video editing testbed.

The figure 4 shows the global architecture of the video editing testbed. On this figure main component are detailed in this paper such as:

- Speech2keyword generator,
- Implicit Video Annotations and,
- keyword-based video editing.

4.1 Speech2keyword Generator

Speech-to-text engine like Nuance speech recognition engine, Sphinx or other are really efficient for a limited of vocabulary, but can be less efficient when voice or accent is not trained by the system. Currently we obtain 50 to 80% quality for the text transcription. In some cases we do not need the full text transcription but only an idea of the concepts and subjects addressed by the speaker. The idea is to reduce errors of the Speech-to-text engine with the Speech2keyword generator, that is able to extract the keywords in a time-based manner from the speech transcription in real-time. Next steps are to apply semantic algorithms to improve the consolidation/disambiguation of extracted keywords and reduce the errors.

The current implementation includes ASR (Automatic Speech Recognition) of Nuance (Beaufays et al., 200) and semantically methods to extract keywords (Figure 5).

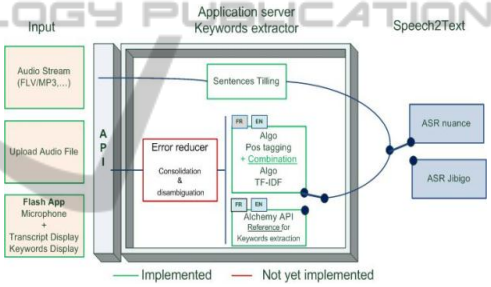

Figure 5: Speech2keyword generator.

The quality of the speech to text is key element in the keyword extractor and when the voice or accent is not trained by the system. The result of the transcription success is about 50 to 80% depending on audio file input.

The cosine similarity computation is used to compare the both extracted keywords is given by :

$$similarity = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (8)$$

Where A and B are vectors of attributes.

To show the speech2keyword generator efficiency with an example, we have converted a text to mp3 file and have used the audio file to extract the transcription and keyword using alchemy APIs. To convert text to mp3, we have used a web-based free access tools text2speech.org (http://www.text2speech.org/). The following

parameters have been used:

Voice type: American Male 1
Volume scale :5

The original text is:
Video editing is the process of selecting segments from a set of raw videos and chaining them by adding piece of audio, effects and transition in order to create a new video or mashup. Most available video editing tools are time-consuming and require specific knowledge. There is therefore, an increasing demand for simple, user-friendly online video editing tools. Online video editing tools can afford less costly and more accessible processes that can be completed anytime, anywhere.

The transcription using speech2keyword generator of the generated audio file by text2speech.org is given below:

<metadata>
<session id="11">
<stream id="mp3 text2speech">
<audio>
<speech2keywords begin="00.00.00" end="00.00.15" sentences="Video editing just the process of selecting segments from a set of ***wrong*** *videos and* ***shaving*** *them by having piece of audio effects and transitions in order to create a video on the* ***sharp*** *most Available video editing tools are time consuming and require specific knowledge there is therefore an increasing demand for simple user friendly online video editing tools online video"*
</speech2keywords>
</audio>
</stream>
</session>
</metadata>

As can be seen, there some difference between the audio transcriptions and the original text. The following table compares the keywords extracted from original text with the keywords extracted by speech2keyword generator using Alchemy algorithm. More than 60% of keywords are identical allowing us to confirm that the quality of the transcript is acceptable. This result is very interesting for automatic video editing to quickly edit a mashup when a user is dealing with a large number of videos.

4.2 Implicit Video Annotations

The proposed approach allows, implicitly, annotation of videos. A user when composing the video adds textual data such in titles, captions, textboxes and tickers in order to enrich his video mashup. A video mashup is composed by video clips. These videos are trimmed by the user to select only the interested segments or clips to build his small movie or a video mashup. The figure 6 shows an example of implicit annotation of a video inherited from mashup user annotations. Text processing and statistics analysis will be performed in the future work in order to obtain a better description of each video shots or segments.

Table 2: Comparing original text with speech2keyword generator.

Original text using http://www.alchemyapi.com/api/demo.html	speech2keyword generator using Alchemy
video editing tools online video editing Most available video user-friendly online video new video raw videos specific knowledge chaining time-consuming	video editing video editing tools user-friendly online video Available video wrong videos audio effects specific knowledge time-consuming online accessible processes tools anywhere

```
<metadata>
  <annotations>
    <annotation>
      <type>textboxeffect</type>
      <text>The buffalo before the attacks of lions</text>
      <start>51</start>
      <duration>9.1</duration>
    </annotation>
    <annotation>
      <type>textboxeffect</type>
      <text>A clan of lions attacking a buffalo</text>
      <start>142</start>
      <duration>71</duration>
    </annotation>
      <type>titletheme</type>
      <text>Wild animals</text>
      <start>0</start>
      <duration>5</duration>
    </annotation>
  </annotations>
  <mashlabel>WildAnimals</mashlabel>
  <othermedia>
    <label>Lions Hunt Buffalo.flv</label>
    <label>Rhinoceros attacking tour bus.flv</label>
  </othermedia>
</metadata>
```

Figure 6: Implicit media annotation.

4.3 Keyword-based Automatic Video Editing

In our research, the video compositions are based on text data added by users directly of implicitly and

meta-data extracted by media analyzers (speech2keyword generator). The text data will be continually processed using text processing and statistical algorithms in order to better describing shot, sequences or all media file. For instance, the video composition allows automatic video composition by keyword (Figure 7). First a user or an application performs an HTTP request with a "keyword", a user and the name of the rule as parameters (1). The keyword is processed (2) and a search request is sent to the database (3). Using rules, the automatic video editor (4) compose an XML descriptor which chain different video shots matching with the keyword. Audio file can be added in the video composition if the keyword matches with auto description. The XML descriptor of the composition is stored in the database (5). An end user can perform the following function using the web based video editing application: play the composition (6 and 7), modify the video composition (8) and render the composition by converting the XML descriptor on video format.

5 MODEL AND ALGORITHM VALIDATIONS

The video editing testbed has been built for studying and testing automatic video editing based on text data. To experiment the automatic video editing algorithms, we will use a video dataset with multiple semantic concepts in using video. We propose two methods to experimentally validating models and algorithms of automatic video editing: Mashup validation by users and comparison of the mashups to one or more reference mashups (depending on the context). For the first methods, the algorithms will use the user profiles. The user feedbacks analysis information will be processed and injected in the algorithms in order to improve future compositions. For the second, representative users will create reference mashups in different domains or subjects. The algorithms will be tested the mashup compared to reference mashups.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed an approach of automatic video editing which is derived using algorithm and rules based on keyword. The text data is collected in three ways: direct user annotations, implicit annotation during the video edition and by extracting keywords from video analysis. For the future we plan to create more complex models and algorithms of video composition to allow composing video from a sentence.

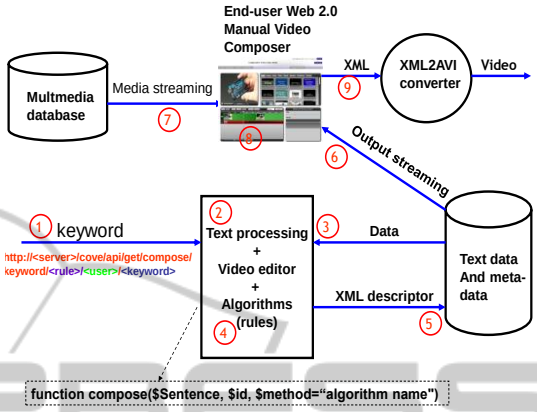

Figure 7: Automatic video editing engine.

REFERENCES

Hua, X.-S., Zhang, H.-J., 2009. Automatic Home Video Editing, Signals and Communication Technology, *Springer*, 353-386.

Hua, X.-S., Zhang, H.-J., 2003. AVE - Automated Home Video Editing, *ACM MM*.

Ma, Y. F., Lu, L., Zhang, H. J., Li, M. J., 2002. A User Attention Model for Video Summarization. *ACM MM*, 533–542.

Müller Arisona, S., Müller, P., Schubiger-Banz, S., Specht, M., 2008. Computer-Assisted Content Editing Techniques for Live Multimedia Performance, *R. Adams, S. Gibson, and S. Müller Arisona (Eds.)*: DAW/IF, CCIS 7, 199–212.

Takemae, Y., Otsuka, K., Yamato, J., 2005. Automatic Video Editing System Using Stereo-Based Head Tracking for Multiparty Conversation, *CHI 2005*, 1817-1820.

Takemae, Y., Otsuka, K., Yamato, J., 2005. Development of Automatic Video Editing System Based on Stereo-Based Head Tracking for Multiparty Conversations, *IEEE*.

Mudhwuchutyula, C. L., Kankunhalli, M. S., Mulhem, P., 2004. Content Based Editing of Semantic Video Metadata, *IEEE International Conference on Multimedia and Expo*.

Foote, J., Cooper, M., and Girgensohn, A., 2002. Creating Music Videos Using Automatic Media Analysis, *ACM MM*.

Beaufays, F., Sankar, A., Williams, S., Weintraub, M., 2003. Learning Name Pronunciations in Automatic Speech Recognition Systems, *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence.*