

# A Learning Environment For Creating Media Processing Systems

Gang Wei, Valery A. Petrushin and Anatole V. Gershman

*Accenture Technology Labs, Accenture*

*161 N. Clark St., Chicago, IL 60601, USA*

*{gang.wei,valery.a.petrushin,anatole.v.gershman}@accenture.com*

## Abstract

*The Community of Multimedia Agents project (COMMA) is devoted to creating an open Web-based environment for developing, testing, learning and prototyping multimedia content analysis and annotation methods. Each method is represented as an agent that can communicate with the other agents registered in the environment using templates that are based on MPEG-7 descriptors and description schemes. The low-level agents can be combined to obtain more sophisticated ones. The paper discusses the educational aspects of the COMMA project. It describes both agent development tools that can be considered as learning environment in the narrow sense and the Webbased community as a collaborative learning environment.*

## 1. Introduction

The explosive growth of digitized image, audio and video data is making the efficient content-based indexing and retrieval increasingly important, which requires the ability to automatically analyze, understand and annotate multimedia content. A large number of approaches have been proposed in this area [1, 2]. These approaches are ranging from cut detection in video and music/speech detectors in audio to complex object tracking in video [3], emotion recognition in audio [4], topic detection and tracking using audio transcripts [5] and automatic summarization of TV programs [1]. However, the capability of the current techniques is still far from the human perception of multimedia content and more work has to be done. Knowledge and experience accumulated in multimedia analysis area made the standardization issues urgent. The Multimedia Content Description Interface (MPEG-7) coming standard [6] promises to provide a unified base for multimedia content description for both producers and consumers.

Agents are defined as active, persistent software components that perceive, reason, act, and communicate

[7]. Agent-based approach proved to be very useful in many applications. We found that the concept of agent is highly valuable for multimedia analysis. Most of the multimedia processing systems uses agents (in the above mentioned sense) implicitly or explicitly [8,9].

## 2. Motivation

Multimedia content analysis requires expertise in a number of fields such as image and video processing, audio processing, speech recognition, linguistics, information retrieval and knowledge management. The range of expertise spans from DSP techniques for feature extraction to methods for knowledge representation, integration and inference. It is unlikely a researcher or a research laboratory can cover the required range of expertise to develop a multimedia analysis system from scratch. Usually, each lab concentrates on its own research agenda using commercial tools (if available) or borrowing some experimental tools from other researchers to develop a rounded-up multimedia analysis prototype. Borrowing from the others is not easy due to variety of platforms, programming languages, data exchange formats and unwillingness of companies to disseminate their intellectual property unprotected. A lucky researcher can get a tool that covers a particular task, for example, face detection; an unlucky researcher has to implement a tool by himself. In any case, the researcher will have only one or two (if any) face detectors, in spite of his awareness that two dozens of such tools exist in the world. This scarcity of media analysis tools and difficulty in finding and learning them motivated our COMMA project. The project's general objective is to create a virtual community of researchers and students, who exchange their multimedia analysis tools and test data. The Community's objective is to consolidate efforts and expedite research and education in multimedia analysis. To facilitate exchanging and combining of media analysis tools, the following requirements are held:

– The Community is located on the World Wide Web

and is accessible from any Internet-enabled workstation.

- The Community provides a library of multimedia analysis agents. Agents are represented in an executable form, thus protecting the proprietary details of agents' design. Any community member can submit and download agents.
- Copyrights belong to the agents' authors or their organizations. The Community members are granted a license for free non-commercial usage of the agents.
- The Community provides templates for agents' outputs that facilitate communication among agents and allow building of agent hierarchies.
- The Community provides open source tools for creating agents and visualizing their performance. These tools can be freely downloaded from the CommunityWeb site.

Currently we foresee the following stages in developing the COMMA project.

**Stage 1. Just Agents.** The objectives of this stage is to:

- Develop tools for creating agents and visualizing their work; develop MPEG-7 based templates for agents' outputs.
- Develop a Web-based portal that contains links to lead laboratories and researchers, tutorials, tools and datasets.
- Accumulate initial "critical mass" of agents, and launch the communityWeb site.

The Accenture Technology Labs has released a first version of the agent development and visualization tools for Windows 2000/XP platform. We are also collaborating with several Universities to create an initial library of agents, which will be released at the Community's Web site. The Community will serve both researchers and students. A researcher can compare his/her approach to the known approaches presented in the agent library, combine agents to create a high-level agent, and do a rapid prototyping of a system that solves a particular problem. A student can learn about various approaches to solve a problem, get a hand-on experience in building media analysis algorithms and systems, and learn up-to-date data representation technologies, such as XML and MPEG-7.

**Stage 2. Intelligent Agents.** The next step is to create more sophisticated agents that are built from low-level agents. We also plan to create a formal description of agent functions and develop tools that can automatically combine (synthesize) agents to solve a specified problem.

**Stage 3. Distributed Agents.** The further step is to develop formal specifications, interfaces and tools that

allow distributed agents to find each other on the Web and to communicate and solve a specified problem. At this stage the Community of researchers and students will be extended to the Community of Multimedia Agents to justify the title of the project. Some steps in this direction have already been made for simple business-oriented agents [10].

### 3. Architecture

The Community Web site provides two components: the agent library and the development environment. The agent library consists of a set of agents in executable form and an agent description file, which describes the set of agents in XML. The development environment is an executable file. The current embodiment of the development environment is a program written in Microsoft C++ for Windows 98/2000/XP platform (see detailed description below).

To start using the software a COMMA member should download and install it on a local computer; then run the COMMA environment to register media files and local agents (if any). Three types of media are allowed: still images, audio files, and video files. Eventually, the user will have a configuration, which consists of the agent library, a collection of media files and a collection of annotations (Figure 1). Each agent produces an annotation or a metadata sheet in MPEG-7 language, and all annotations are grouped together to form a "blackboard" description for media files.

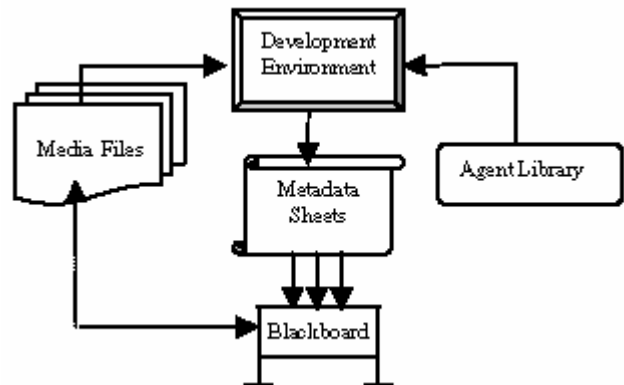


Figure 1. The COMMA Architecture.

### 4. Tools

The development environment provides means for registering media files and agents, and two major tools: a *Workbench* for developing media annotation processes, and a *Blackboard Browser* for visualizing results. The development environment is a learning environment or a cognitive tool, which allows a student to learn the

problem domain of media processing and annotation and

clicking on their descriptions in the agent tree. Each agent

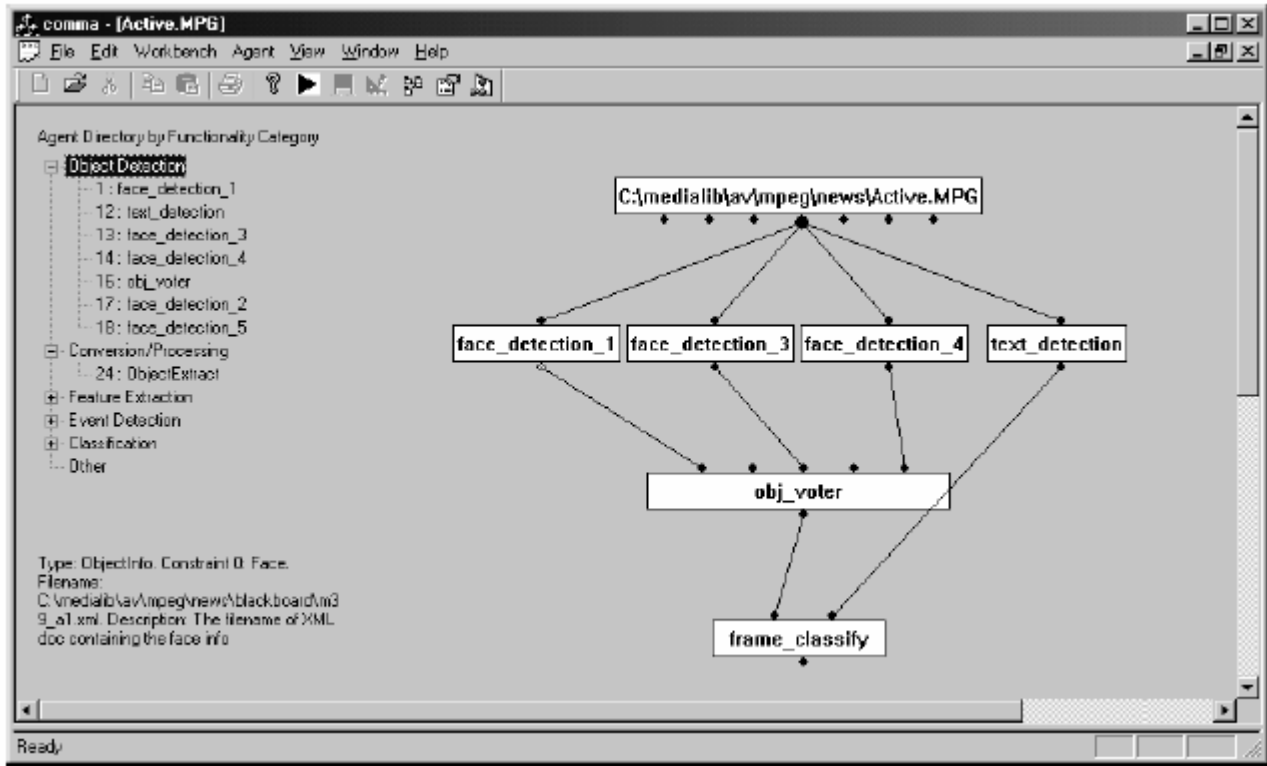


Figure 2. Working in a workbench window.

express his or her knowledge and skills by solving problems. A teacher can evaluate student knowledge in-depth by running student’s agents and comparing results to the “ground truth” results or the results of the well-known agents. The teacher can also examine and evaluate the structure of the student’s system. The environment can be used both as a tool for collaboration among members of a team and as a tool for competition among teams or individual students.

#### 4.1. Workbench

The Workbench allows a user to select and combine existing agents as building blocks to construct media annotation processes. The user starts by selecting a media file, then Workbench displays agents that can process the media of this type and format. The agents are organized by their functionality in a tree structure (Figure 2). When an agent is selected, the Workbench displays a short agent description in the text box below the agent tree area. In the working area, the target media is represented as a rectangle with a number of output pins at the bottom. The largest pin corresponds to the raw data and the others to the results produced by agents. Those processing results can be used as inputs to agents to reduce processing time. The user can load the agents to the working space by

is represented as a rectangle with input and output pins.

For example, Figure 2 shows a video file in mpeg format that is processing by four agents – three face detection agents and a text detection agent, which detect faces and superimposed text in each I-frame correspondingly. The output pins of the face detection agents are connected to inputs of a voting agent. This agent can serve as an example of a high-order agent that uses the results of other agents. It can accept up to five inputs and has a parameter (tuner) that specifies the mode of “voting”, which could be “or” (a frame has a face detected if at least one of the agents detects a face), “and” (if all agents detect a face) or “majority” (if the majority of agents detect a face). Another example of a high-level agent is a frame classification agent. It uses the outputs of the voting agent and the text detection agents and outputs a higher-level classification of frames and sequences of frames. The user can save the system composed of agents as a script and later load it as a “macroagent”. By clicking the triangle button on the toolbox bar the user can run the script. The Workbench coordinates the script execution and manages the annotations. Currently the tool does not support parallel or real-time execution of agents.

The user can use any programming language to develop an agent if the final result is an executable module. The system also allows using any interpretive

language for agent development, but installation of the frame can be categorized as “no faces detected” (white



**Figure 3. Blackboard Browser window.**

interpreting program should be done separately. To facilitate agent creation we provide tutorials for the following development languages: C++, Java, and Perl. We plan to include MATLAB to the list.

#### 4.2. Blackboard Browser

The results of an agent’s work are represented as MPEG-7 descriptions. Each agent can generate one or more MPEG- 7 annotation files. The user can assign a visualization mode to each annotation. Currently several visualization modes are implemented such as categorical color bar, time series graph, thumbnail picture strip, histogram, scatter plot, etc. The Blackboard Browser serves for visualizing the results based on their visualization modes and type of the media. Figure 3 shows a Blackboard window for a video file. It contains video browser on the right side, a current frame image on the left side that presents agents’ results, and a summary of agents’ findings for the current frame in the middle of the screen. The user can watch the results for any frame using the navigation buttons. Below the frame and time scales are the summaries of agents’ findings for the whole clip. For example, the summary of a face detection agent is presented in a form of the categorical color bar. Each

color), “one face detected” (blue color), and “multiple faces detected” (red color). The same color code is used for the text detection agent’s results. A user can explore how a particular detection agent works by clicking on the agent’s summarization strip and watch the results represented on the current frame picture as a rectangular that frames a detected face or text. Or by clicking on the time scale the user can watch the results of all agents simultaneously on the same picture.

#### 5. Comma as community of learners

One of the COMMA project main objectives is to create a community of researcher and students in the multimedia processing problem domain. This social aspect of the project is very important for its success. The environment should encourage people to interact, exchange agents and ideas, discuss topics of interest, and advertise relevant events, such as workshops, conferences, training sessions that target both academic and business research and development. That is why we are paying a great attention to information that is provided by the COMMA Web site.

This information includes related business and

academic news, overviews of achievements of lead laboratories and researchers, event and job announcements, book and paper recommendations, tutorials, and glossary of specialized terms. It also includes a directory of community member e-mail addresses and chat rooms for real-time discussions.

Altogether the tools and information form a sociotechnical learning environment that could be beneficial for researchers, teachers and students.

## 6. Summary and future work

The Community of Multimedia Agents is a community of researchers and students, and an open learning environment that allows researchers to share their achievements in multimedia annotation field while protecting their intellectual property. Currently, the Community provides a community Web site packed with relevant information, an agent library, MPEG-7 based templates for agent outputs and tools for creating agents and visualizing their work.

In the future the Community will provide tools for creating more intelligent and distributed agent that will be able provide services on the World Wide Web.

## 7. References

- [1] M.T. Maybury (Ed.) *Intelligent Multimedia Information Retrieval*, AAAI Press/MIT Press, Menlo Park, CA / Cambridge, MA, 1997.
- [2] Yao Wang, Zhu Liu, and Jin-Cheng Huang, "Multimedia Content Analysis Using both Audio and Video Clues", *IEEE Signal Processing Magazine*, IEEE Inc., New York, NY, pp. 12- 36, vol. 17, No 6, November 2000.
- [3] N. Dimitrova, L. Agnihotri, and Gang Wei, Video Classification using Object Tracking, *International Journal of Image and Graphics*. Vol. 1, No. 3 (2001).
- [4] V.A. Petrushin, .Emotion Recognition in Speech Signal: Experimental Study, Development, and Application, In *Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, 2000.
- [5] O.V. Ibrahimov, I.K. Sethi, and N. Dimitrova Clustering of Imperfect Transcripts using a Novel Similarity Measure, In Coden A.R., Brown E.W. and Srinivasan S. (Eds.), *Information Retrieval: Techniques for Speech Applications*, LNCS vol. 2273, Springer-Verlag, 2002, pp. 23-35.
- [6] José M. Martínez, Overview of the MPEG-7 Standard, <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm>
- [7] M.N. Huhns and M.P. Singh, "Agents and Multiagent Systems: Themes, Approaches, and Challenges", In Huhns M.N. and Singh M.P. (Eds.), *Readings in Agents*, Morgan Kaufman, San Francisco, CA, 1998.
- [8] A.J. Hauptmann and M.J. Witbrock, "InforMedia: News-on- Demand Multimedia Information Acquisition and Retrieval", In [1], pp. 215-239.
- [9] B. Merialdo and F. Dubois, "An Agent-based Architecture for Content-Based Multimedia Browsing", In [1], pp. 281-294.
- [10] J. Heflin and J. Hendler, "A Portrait of the Semantic Web in Action", *IEEE Intelligent Systems*, vol. 16, No. 2, pp. 54-59, March/April 2001.