The following paper was originally published in the
Proceedings of the Fifth Annual Tcl/Tk Workshop
Boston, Massachusetts, July 1997

A Tcl/Tk-Based Video Annotation Engine

M. Carrer, L. Ligresti, and T.D.C. Little
Multimedia Communications Laboratory
Department of Electrical and Computer Engineering
Boston University

A Tcl/Tk-Based Video Annotation Engine

M. Carrer, L. Ligresti, and T.D.C. Little
Multimedia Communications Laboratory
Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts 02215, USA
tdcl@bu.edu

The population of a video database requires tools for manipulation and annotation of raw video data. Characteristic of this requirement is the need to satisfy many disparate video-based application domains. In this extended abstract we describe the design and development of a video annotation engine called Vane (Fig. 1), intended to address the issue of domain-independent video annotation. Rather than relying on a single, general data model and application interface, we developed a dynamic interface and data model using the Tcl/Tk environment and SGML document type definitions (DTDs). This approach allowed us to implement an intuitive graphical user interface application that is easily portable to different systems. The outcome of our work is a multipurpose, domain-independent video annotation application that has been developed taking advantage of Tcl/Tk features for easy construction and reconfiguring of GUI widgets at execution time. Thereby offering a novel application model appropriate for the domain.

We built Vane so that the DTD can be modified in many of its parts to suit the needs of the annotator and to better describe the current domain under analysis. Opening a new annotation in Vane means identifying its associated DTD. The particular DTD is then parsed by a Tcl procedure, resulting in the loading of its syntax rules into an array in memory. When the annotation of one of the defined SGML elements is requested by the annotator, a new, top-level window is built (Fig. 2). Attributes belonging to these elements are mapped to a Tcl/Tk widget according to their type. These can be pop-up menus, form entries, listboxes, or text areas for attributes such as content transcripts. Defining a new DTD for a new annotation, or changing part of an existing DTD, does not affect the implementation of Vane, rather, these changes are accommodated by the construction of the user windows from the DTD on-the-fly.

Once the syntax of the annotation document has been extracted from a DTD, each of its fields is associated with its semantics from a corresponding SGML document. The latter carries the video information as annotation data. An additional Tcl procedure handles the output produced by an SGML parser by loading the field values into an annotation array. Because the array indices are based on the DTD array, and therefore on the DTD syntax, identification of format mismatches is easily accommodated. The reverse process, writing-out annotation metadata to SGML, is therefore straightforward as well. In this case we generate SGML output by following the syntax rules stored in the DTD array. The result is a "generated" SGML document which is certainly consistent with its own DTD.

With respect to the static user interface of Vane (Fig. 1), the canvas widget and its associated properties were used extensively. Each element appearing in the main window represents a "hot-spot" with an associated Tcl/Tk binding. Because the binding is based on a motion event, a simple dragging of the mouse pointer over an annotation element automatically updates other information boxes in the window. This offers to the annotator a set of easily accessible shortcuts to check the progress of the annotation process.

In summary, we have used Tcl/Tk as a scripting language and toolkit for the implementation of our annotation tool. The outcome of our work is Vane, a multipurpose, domain-independent video annotation application. It has been developed to achieve a dynamic, run-time-reconfigurable user interface by taking advantage of the unique characteristics of Tcl/Tk. The tool is currently in use to annotate a video archive comprised of educational and news video content in the Multimedia Communications Lab.

Additional details of the Vane implementation can be found at the following URL: *http://hulk.bu.edu/pubs/papers/1996/carrer-vane96/TR-08-15-96.html.*

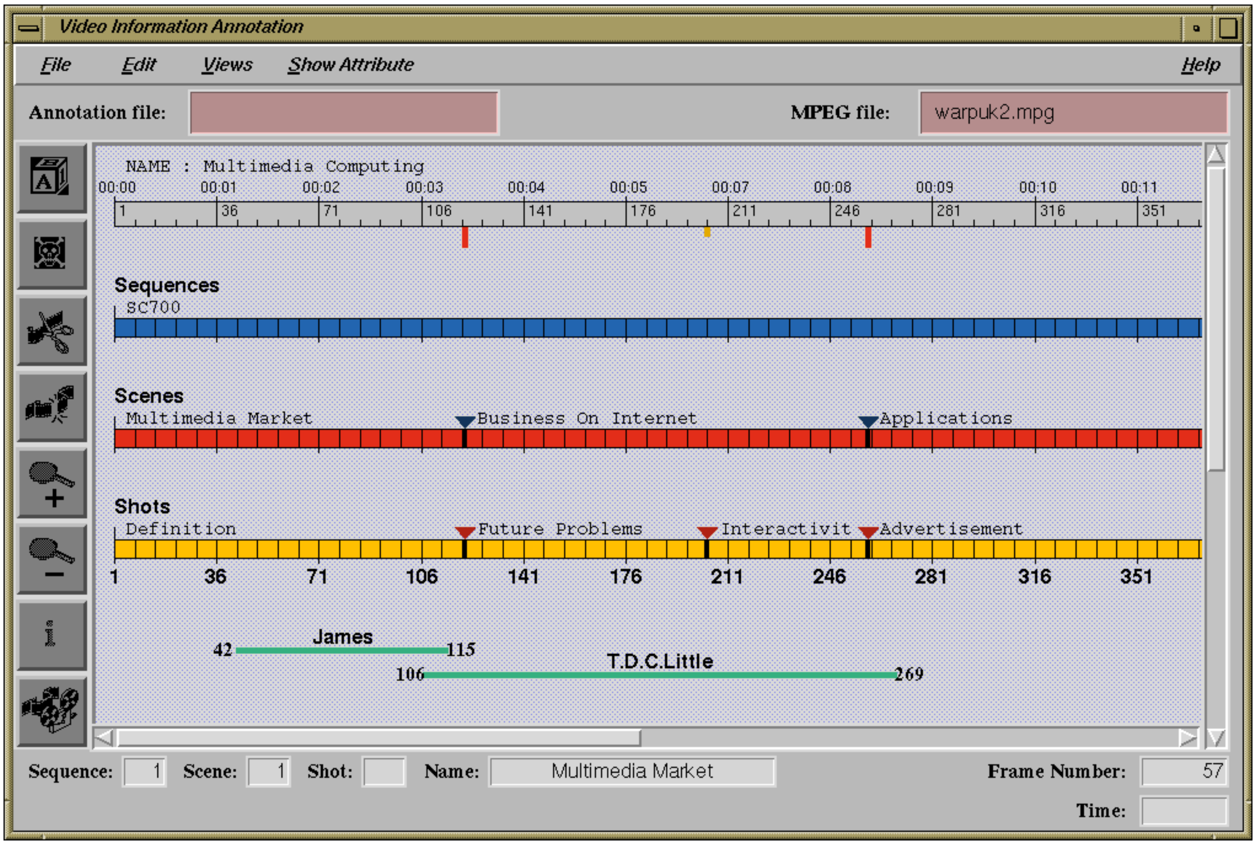

Figure 1: Screenshot of the Vane Workspace Window

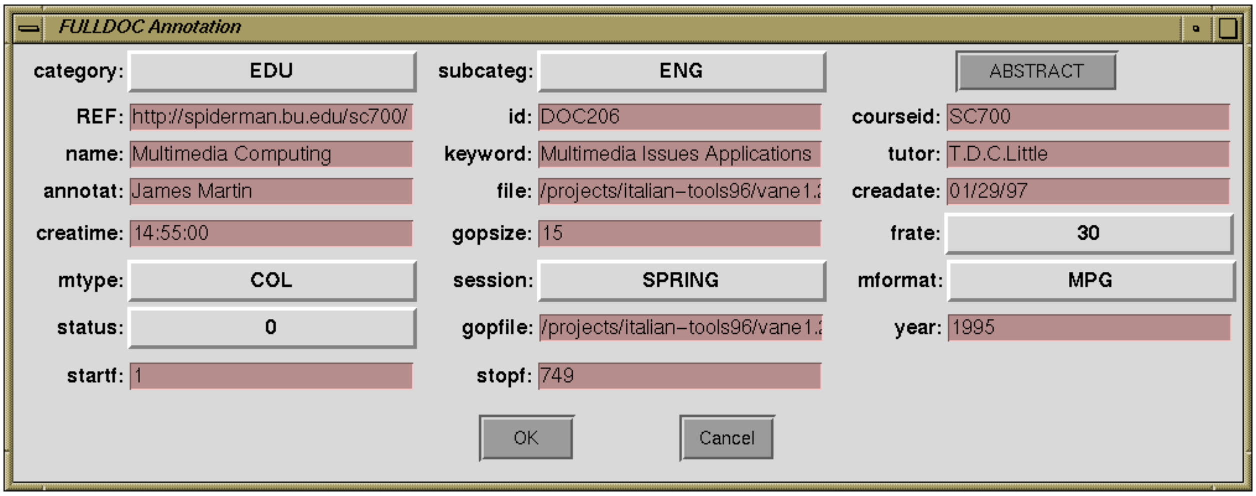

Figure 2: Screenshot of a Vane Annotation Window