Component-based software for dynamic configuration and control of computer assisted intervention systems

Release 1.0

Peter Kazanzides, Anton Deguet, Balazs Vagvolgyi, Marcin Balicki, Min Yang Jung, and Russell H. Taylor

June 15, 2011

Johns Hopkins University, Baltimore, MD

Abstract

This paper will present the rational for the use of a component-based framework for development of computer-assisted intervention (CAI) systems. This rationale includes common arguments for component-based software, such as the ability to reuse components and to easily develop distributed systems. We introduce three additional capabilities, however, that we believe are especially important for research and development of CAI systems. The first is the ability to deploy components among different processes (as conventionally done) or within the same process (for optimal real-time performance), *without requiring source-level modifications to the component*. This is particularly useful for real-time video processing, where the use of multiple processes would cause perceptible delays in the video stream. The second key feature is the ability to dynamically reconfigure the system, which is especially useful during experimentation with complex CAI systems. For example, in a system composed of multiple processes on multiple computers, dynamic reconfiguration allows one process to be restarted (e.g., after correcting a problem) and reconnected to the rest of the system. This is more convenient than having to restart the entire distributed application, which can be detrimental in time-critical situations such as in-vivo experiments. More generally, dynamic reconfiguration enables better fault detection and recovery. The third key feature is the availability of run-time tools for data collection, interactive control,and introspection, and offline tools for data analysis and playback. For example, it may be necessary to collect a large and diverse set of data (e.g., video, audio, robot positions, sensor readings) during an in-vivo experiment, leading to the need for an offline tool to manage this data; conceptually, this can be considered a multi-media editing tool (as opposed to a standard video editing tool that handles only video and audio). The above features are provided by the *cisst* software package, which forms the basis for the Surgical Assistant Workstation (SAW) framework. A complex computer-assisted intervention system for retinal microsurgery is presented as an example that relies on all of the above features. This system integrates robotics, stereo video (from a microscope), force sensing, and optical coherence tomography (OCT) imaging to transcend the current limitations of vitreoretinal surgery.

Contents