TiCoLi: An Open Software Infrastructure for Device Integration in the Digital OR

Michael Gessat[1,2], Stefan Bohn[3], Arun Voruganti[3], Stefan Franke[3], Oliver Burgert[3]
[1] Swiss Federal Institute of Technology (ETH) Zürich, Computer Vision Laboratory, Sternwartstrasse 7, 8092 Zürich, Switzerland
[2] University Hospital Zürich, Division of Cardiovascular Surgery, Rämistrasse 100, 8091 Zürich, Switzerland
[3] Universität Leipzig, Innovation Center Computer Assisted Surgery (ICCAS), Semmelweisstraße 14, 04103 Leipzig, Germany

Keywords: Surgical Plug-and-Play, Digital Operating Room, Systems Interconnectivity, Operating Room Integration

Purpose

Modern surgery relies on computers in many aspects ranging from department and patient data management to preoperative planning and intraoperative assistance. During the last 15 to 30 years, rapid progress was made in the development of Computer Assisted Surgery (CAS) systems. During the highly innovative phase of CAS technology and techniques, only little attention was spent on the development of technical infrastructures for the integration of CAS systems into the Operating Room (OR) and for data exchange between the OR and external information systems. CAS is dominated by "stand-alone island of information" [1] with no connection to other devices within the OR, let alone any hospital-wide information systems. Several interdisciplinary workshops [e.g. 1,2] and journal publications [e.g. 3,4] made clear, that the integration of technical systems inside the OR into a local area network and, on a higher level, the connection of this network with hospital-wide information systems, is one of the keys to a better pervasion of CAS. Their recommendations aim at the evolution from monolithic systems for specific applications to distributed systems created by combination of the functionalities of networked modules. In this paper, we present an open source software library which can be used as an application-level transport layer between software systems in experimental CAS setups.

Methods

Development of a software interface for application-layer systems connectivity was driven by requirements derived from the literature [e.g. 1–4]. The TiCoLi library implements the following functionalities:

- Service Discovery and Session Initialization: In order to facilitate dynamic setups of devices, the TiCoLi utilizes the zeroConf protocol for automatic exchange of network configurations and device identifications.
- Message exchange: Data packages of different kinds, ranging from light-weight status or control messages to large data messages which can contain several hundreds of megabytes can be exchanged.
- Continuous data exchange: For continuously acquired measurements, such as audio, video or other sensor data, a service is offered which assures delivery of frames at regular time intervals.
- Read- and write-access to device parameters: The TiCoLi enables a system to grant peer systems read-only or read-write access to its internal parameters.
- Access to methods: Through the TiCoLi, one system can call a method or function in another system and will receive the results of the method or function.

Results

The internal structure of the c++ class library TiCoLi is presented in Figure 1. The principal design is based on the mediator design pattern with the `TiCoLiCore` class as the central mediator entity. It governs the communication between the `TiCoLiAPI` and the five

`Manager` objects which implement the different functionalities of the TiCoLi. The `TiCoLiAPI` is a static proxy object which acts as an interface for an application to use the TiCoLi services.
The viability of the specifications and the implementation has been proven in a modular setup for model guided neurosurgery for intraoperative acquisition of a 3D map of neurophysiological measurements on the neurocortex. A system sketch and screenshot of the intraoperative visualization are presented in Figure 2. The system utilizes the TiCoLi for auto-configuration, service access, and data exchange.
The efficiency of data exchange between networked modules is an important parameter, especially when it comes to the exchange of status and control messages as well as to continuous data exchange through the streaming service. A series of experiments was conducted to quantify the performance of the TiCoLi under different network load conditions. Up to a network load of 80% (on a 100 MBit/s network), no significant delays were identified for streams and messages. Problems arose, though, when large data messages were sent through the network while it was used to transmit streams. In such situations, the data transmission would “steal” bandwidth required for the streams, resulting in a decline of the frame delivery rate. We are currently evaluating different approaches to load-balancing for integration into the library. With such a mechanism, it would be possible to reserve bandwidth for streams and thereby to protect streams against data messages or other streams.

Conclusion
The TiCoLi is the first open source software library for medical device integration which combines the presented functionalities under one API. It complies with the requirements to an OR infrastructure which are listed in several workshop reports. In a practical scenario, the suitability of the library in a clinical research setting for integration of a CAS prototype was shown.
Since 2011, the TiCoLi is officially an open source project. The authors would appreciate any feedback and contribution as well as seeing the TiCoLi being used in practical projects at other centers.

References
[1] Cleary KR, Kinsella A (2004) OR2020 the operating room of the future – workshop report. ISIS Center, Department of Radiology, Georgetown University Medical Center
[2] Jolesz FA, Shtern F (1992) The operating room of the future. Report of the National Cancer Institute Workshop, Imaging-Guided Stereotactic Tumor Diagnosis and Treatment. Investigative Radiology 27(2): 326–328
[3] Deinhardt M (2003) Manipulators and integrated or systems – requirements and solutions. Minimally Invasive Therapy and Allied Technologies 12(6):284 – 292
[4] Lemke HU, Vannier M (2006) The operating room and the need for an IT infrastructure and standards. International Journal of Computer Assisted Radiology and Surgery 1:117–121

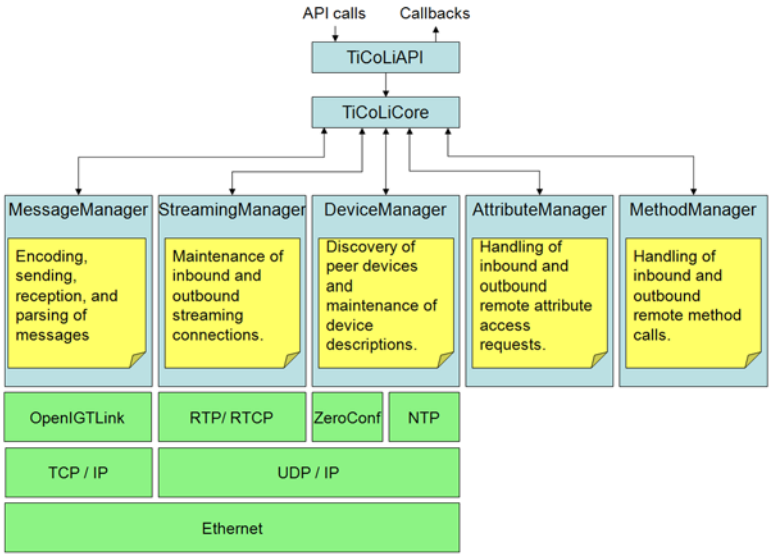

Fig. 1. Principal Architecture of the TiCoLi: implementation is hidden behind an API; the TiCoLiCore acts as a mediator between the Manager classes, which implement the functionalities.

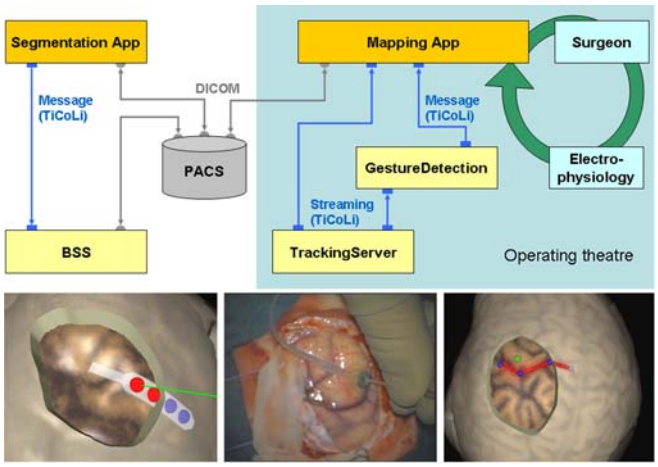

Fig. 2.
Top: System sketch of the Neuromapper system.
Bottom: Intraoperative tracking of probe on the neurocortex and visualization of 3D map.