JTelematik

- ein universelles modulares System für die Telemedizin -

H. Brinck, Ch. Bührig, M. Tietz[1], J. Krone, T. Siebenborn[2]

[1]FH Gelsenkirchen - August-Schmid-Ring 10, 45665 Recklinghausen
[2]MFH Iserlohn/Hagen - Frauenstuhlweg 31, 58644 Iserlohn
Email: Heinrich.Brinck@fh-gelsenkirchen.de

Zusammenfassung. Das Telemedizinsystem JTelematic deckt die Tele-Anwendungsgebiete (Video-) Conferencing, Austausch von Messwerten und Bildern bis hin zur Steuerung von medizintechnischen Geräten wie Mikroskopen ab. Es ist lauffähig auf preiswerten Standard-PCs, darüber hinaus aber plattformunabhängig, sehr flexibel einsetzbar und einfach zu bedienen und unabhängig von der Vernetzungstechnologie.

1 Einleitung

Telemikroskopie bedeutet die Übertragung mikroskopischer Bilder über digitale Kommunikationsnetze. Sie erlaubt die Fernsteuerung eines automatisierten Video-Mikroskops und ermöglicht die Kommunikation von Teilnehmern einer Mikroskopiesitzung z.B. über Video-Conferencing. Nicht nur für diesen Anwendungsfall wurden frei konfigurierbare Telematik-Anwendungs-Module geschaffen, die auch für andere Bereiche der Telemedizin, z.B. die Teleradiologie, die bildgestützte Telediagnostik, aber auch für "Tele-Homecare" kombinierbar sind.

2 Das Telemedizin-System JTelematik

Plattformunabhängige Kommunikation zwischen Computern und daran angeschlossenen Geräten bekommt in vielen Bereichen eine immer größere Bedeutung. Hier fehlt es aber größtenteils noch an einer universell einsetzbaren und gut skalierbaren Software. Diese Lücke schließt die Software JTelematic.

Als Programmiersprache wurde Java gewählt, um eine weitgehende Plattformunabhängigkeit zu gewährleisten, d.h. prinzipiell läuft diese Software auf jedem Java fähigen Gerät. Um ebenfalls eine leichte Erweiterbarkeit zu gewährleisten, ist die Software streng modular aufgebaut. Der Kern der Software besteht aus einem Verwaltungsmodul, welches die Konfiguration und das fehlerfreie Zusammenspiel der ausgewählten Funktionsmodule gewährleistet. Die Funktionsmodule verfügen über genau definierte Ein- und Ausgänge und kommunizieren so untereinander und mit der Außenwelt. Die Funktionsmodule werden problemspezifisch in einem Setup zusammengestellt und verbunden.

Die Softwareentwicklung erfolgt komponentenbasiert. Unter einem Modul wird hier eine Komponente verstanden, also ein wieder verwendbares Objekt, das mit anderen Objekten interagieren und externe Ressourcen kontrollieren kann. Jeder Modul ist ein eigenständiger Programmfaden (Thread) mit Javaklassen als Ein-/Ausgänge sowie drei (Standard-) Schnittstellen zur Ablaufkontrolle, zur Einstellung und zur Information. Die Ein-/Ausgabe-Objekte werden in verketteten Listen abgelegt, die Kommunikation zwischen Modulen erfolgt ähnlich dem Event-Handling.

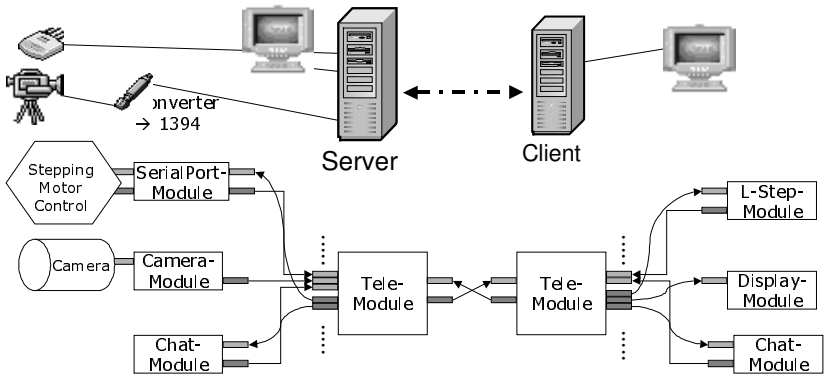

Abb.1 Modulare zusammengestelltes Telematik-System für die Telemikroskopie

3 Der Anwendungsfall Telemikroskopie

Als Beispiel für eine Telematik-Anwendung wurde ein Telemikroskopie-System aufgebaut. Die Aufgabe besteht darin, ein Video-Mikroskop mit Mikroskopsteuerung remote zu steuern. Dazu müssen ein Livebild übertragen und Mikroskop-Steuerbefehle zurückgegeben werden. Die Kommunikation zwischen Teilnehmern erfolg durch Chatten (Abb.1).

An einem Server-Rechner ist unter dem Betriebssystem Windows 2000 eine Bildverarbeitungskamera und zusätzlich über die RS232-Schnittstelle eine Mikroskoptisch-Steuerung angeschlossen. Folgende Module werden auf dem Server verbunden:

- Telemodul: stellt die Netzwerkverbindung über das Internet/LAN o.ä. her.
- Serial-Port-Modul: stellt die Kommunikation mit externen Geräten über den RS-232-Port her.
- L-Step-Modul: erzeugt Steuerbefehle für die Schrittmotoren im Mikroskoptisch.
- Conference-Modul: ermöglicht die Kommunikation zwischen dem Operator vor Ort und dem Nutzer des Client-Rechners.
- Video-Modul: dient zur Aufnahme von Bildern über eine beliebige Kamera.
- Display-Modul: dient zur Anzeige der Bilder.

Mit diesem Server-Rechner ist über ein IP-Netzwerk ein entfernter Client-Rechner verbunden. Auch hier läuft die Software JTelematic, wobei an diesen Rechner keine

besonderen Anforderungen hinsichtlich Hardware oder Betriebssystem gestellt werden, da hier das Serial-Port- und das Video-Modul nicht geladen werden (Abb.2).

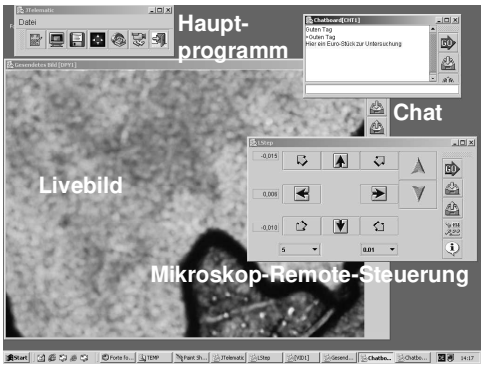

Abb.2 Telemikroskopie-Client: Hauptprogramm, Livebild, Mikroskop-Remote-Steuerung, Chat

4 Die Frame-Grabber-unabhängige flexible Bilderfassung

Das Videosignal liefert 30 Bilder/s der Größe 768 x 576 Pixel. Davon sollen im obigen Beispiel 5 Bilder/s gegrabbed und zum Client übertragen werden. In der Vergangenheit basierte die Bilderfassung in der Regel auf Hardware-Frame-Grabber, die jeweils eine spezielle Software-Schnittstelle, im Windows-Fall in Form einer DLL besaßen. Die Betriebssysteme ab Windows 20000 basieren auf dem Windows-Treiber-Modell (WDM – Windows Driver Model) [1]. Diese Softwareschicht deckt alle Devices ab (z.B. Mouse, Audio-Devices, Video-Devices, ...). Als Hardware-Schnittstelle für Video-Devices hoher Bandbreite ist IEEE 1394 zu empfehlen.

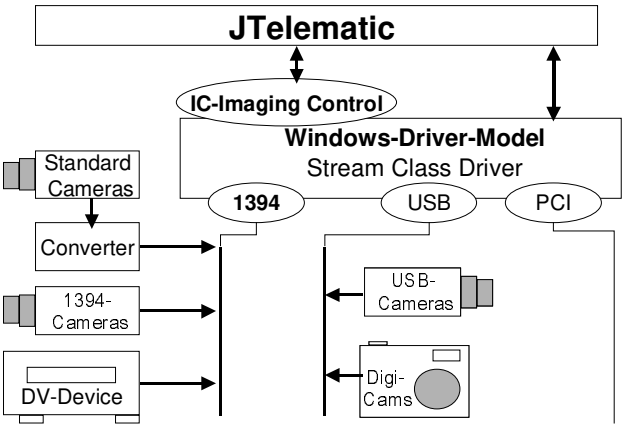

Abb.3 Bilderfassung über das Windows-Treiber Modell ohne proprietäre Frame-Grabber

Wir arbeiten mit dem WDM über das IC Imaging Control [2]. Dieses Control wird über eine Windows-Funktionsbibliothek (DLL) angesteuert. Um diese zu nutzen, wird unter Verwendung der Java-Schnittstelle JNI eine externe, von uns erstellte DLL angesprochen, die die Daten ausliest (Abb.3). So können wir unmittelbar Firewire-Kameras, USB-Kameras, DV-Kameras und viele PCI-Frame-Grabber und Video-To-Firewire-Konverter (Abb.1) unterstützen. Fazit: Es können Bilder von allen Quellen gegrabbed werden, die konform der WDM-Stream-Class-Treiber-Spezifikation sind. Proprietäre Frame-Grabber Lösungen werden so ersetzt. Diese Lösung skaliert von einfachen USB-Web-Cams bis zu High-End-Digitalkameras.

5 Diskussion und Ausblick

Ein wichtiges Problem bei der Bilderfassung, der Bildübertragung und dem Bildaufbau mit Java ist die Performance. Java hatte bei der zuerst bei uns verwendeten Version 1.3 gerade beim Bildaufbau eine derartig niedrige Geschwindigkeit (700 ms pro 640x480 Bild), dass selbst die Netzwerkbandbreite untertroffen wurde. Dies verbot den Einsatz einer Videokonferenz. In der neuen Merlin-Version 1.4. bietet Java neben einem sehr schnellen JIT-Compiler "Hot Spot" einen neuen Grafikmodus mit Direktzugriff auf den Videospeicher. Erste Tests zeigen einen schnelleren Bildaufbau. Um systemunabhängig zu sein, haben wir vor dem Einsatz von WDM die von Sun angebotene Erweiterung Java Media Frame (JMF) [3] zur Bilderfassung benutzt. Es zeigte sich aber, dass das JMF die auf unserer Arbeitsplattform Windows standardisierte Unterstützung "Video für Windows" (VfW) nur sehr eingeschränkt verwenden kann, da unterschiedliche korrekt installierte Treiber vom JMF nicht erkannt wurden.

Da JTelematic modular aufgebaut ist, kann leicht neue Funktionalität hinzugefügt werden. Es werden jeweils nur die konkret benötigten Module geladen. So ist immer eine einfache und applikationsbezogene Bedienung gegeben. Durch Module zur Ansteuerung weiterer medizinischer Geräte können weitere Telematik-Anwendungen entwickelt werden bis hin zur Patientenüberwachung zu Hause über Tele-Messwert-Erfassung. In Arbeit sind Module für die Bildkompression, die Verschlüsselung, die Übertragung von Live-Bild und hochaufgelöstem Einzelbild und das Abspeichern der Bilder z.B. in einer Bilddatenbank. XML-Konfigurations-/Setup-Dateien und ein Bildverarbeitungsmodul werden das Telepaket abrunden.

7 Literatur

1. WDM: Introduction to Windows Driver Model
   http://microsoft.com/hwdev/driver/wdm/wdm.asp
2. IC Imaging Control - http://www.imagingcontrol.com
3. Java Media Framework - http://java.sun.com/products/java-media/jmf/index.html

Dieses Projekt wurde unterstützt vom Innovationsprogramm Forschung des Ministeriums für Schule, Wissenschaft und Forschung des Landes Nordrhein-Westfalen.