OPENNESS IN (TELE-) RADIOLOGY WORKSTATIONS: THE CHILI® PLUGIN CONCEPT

Uwe Engelmann, Andre Schröter, Ulrike Baur, Markus Schwab,
Oliver Werner, Manuela Makabe, Hans-Peter Meinzer

Deutsches Krebsforschungszentrum, Abteilung Medizinische und Biologische Informatik,
Im Neuenheimer Feld 280, D-69120 Heidelberg, Germany
email: U.Engelmann@DKFZ-Heidelberg.de

in cooperation with
Steinbeis-Transferzentrum Medizinische Informatik, Heidelberg, Germany

1. INTRODUCTION

Several problems can be observed today in the field of computer assisted radiology. Nearly every application in the radiology department needs it's own computer: Console computers are necessary to operate the imaging modalities. Viewing stations are used for reporting or further analysis of the image data. Terminals of the radiology informations system RIS are used to manage and organize the patient information flow. Additional personal computers are used for word processing (e.g. reports, publications). New computers for teleradiology are now entering the radiology department.

Developers of new medial image analysis applications have the problem, that the current computer applications (e.g. viewing stations) are not open enough. Every developer has to care about the connection to the modality or viewing station, file formats, data storage of files on the disk or in a data base system, and the display of 12-bit images on 8-bit screens. Developing a new method for the radiologist means to invent many wheels a second or third time before the real problems can be addressed. Most of the manufacturers of the existing viewing stations do not allow the installation of additional software. Usually, the customers do not get the root password of the systems. An additional computer with different user interfaces and additional interfaces to the rest of the world has to be introduced therefore to get the new functionality to the users.

The result of the depicted problems is that it needs a lot of effort for developers to implement and introduce new methods and that it is costly for the end user.

This paper describes how the (tele-) radiology workstation CHILI avoids a number of this problems.

2. DIFFERENT CONCEPTS OF "OPENNESS"

Openness in most of the cases means that a system can exchange data with other systems with standardized protocols and formats. For a radiology workstations this means that images

can be received via the DICOM protocol as C-Store Provider. Images can be exported to other PACS components, such as viewing stations or digital archives by the DICOM protocol (C-Store User). An alternative to the DICOM protocol is the exchange of data as files in standard formats in the file system of the computer. Other open interfaces can be realized with the HL7 protocol for the communication with the RIS or HIS systems.

We are introducing now the new concept *functional openness*. This means that the functionality of an application can be extended after it's completion by the developers at runtime even by the users. Two requirements can be identified:

- Integration of *existing applications* (in executable form; without change of the source code) into a system.
- Integration of *additional functional modules* into the target system which are able to communicate with other modules of the application, like the data base system.
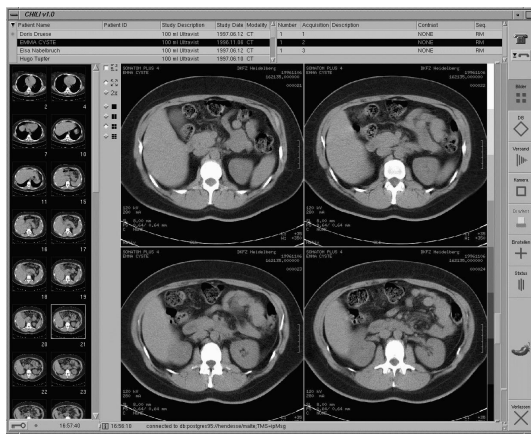

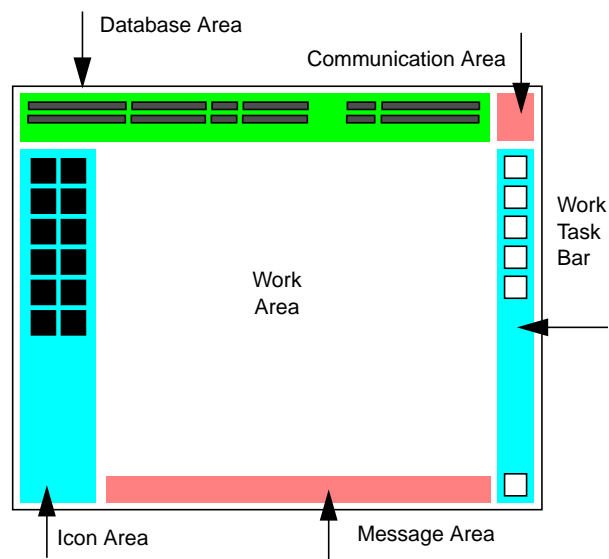Fig. 1.   The user graphical interface of CHILI

Fig. 2.   The areas and components of the graphical user interface of CHILI

3. THE CHILI® PLUGIN CONCEPT

CHILI® is a (tele-) radiology workstation which has been developed in a cooperation of the German Cancer Research Center and the technology transfer company Steinbeis-Transferzentrum Medizinische Informatik, both located in Heidelberg, Germany. CHILI receives medical images via the DICOM protocol or propriety solutions. The images can be transferred to other workstations over the Internet or ISDN lines. Two users at different locations can view and manipulate the images cooperatively in synchronized teleconferences where they can additionally talk to each other. Twenty three systems are currently in clinical use (March '98) in Germany and more than 90 thousand images have been processed with it (and it's prototype MEDICUS) [1]. The system has advanced teleradiology functions but it is in fact a general purpose radiology workstation which is connected to the other systems in the radiology department or hospital. CHILI is an open system with respect to data input and output in a standardized manner as described above. The exchange of images with workstations of other vendors is

possible via the DICOM protocol.

As the developers of CHILI come from the scientific image analysis field they were aware of the problems of "functional openness" and thus they designed and developed the CHILI PlugIn Concept. Plug-ins are software programs that extend the capabilities of existing programs in a specific way. The plug-in concept of the Netscape Navigator is a well known example for that [2]. It is giving us, for example, the ability to play audio samples or view video movies from within Navigator. Software companies are developing plug-ins at a phenomenal rate and extent the functionality of the Netscape Communicator for specific needs.

3.1 Integration of existing applications

Existing applications can be integrated into CHILI as an own work task which can switched on and off via the graphical user interface management functions. A small interface program has to be written which enables the CHILI application to launch the external program and to "catch" it's main window in the X Window hierarchy. The interface program manages the communication between both stand-alone programs.

3.2 Integration of additional functional modules

Additional functional modules, PlugIns in the CHILI terminology, can be added to CHILI as external linkable objects (dynamic shared objects). The programmer is supported by a set of functions in a CHILI PlugIn Library of the Software Developers Kit.

Figure 1 and figure 2 show the principle layout of the user interface with its different static components: Database Area, Communication Area, Icon Area, Work Space, Work Task Bar and Status Area. The layout of the graphical user interface cannot be changed. Only the size of the areas can be manipulated. The content of the center area, the Work Space is specific for the different work tasks of the program, like Viewing, Data management, Image Transfer, Printing, Configuration, or Status Display. The user can switch between the work tasks by pressing a button at the Work Task Bar. Only one work task is visible at the time, but they all exist in parallel and do not change their state when the user switches from one to another. Figure 1 shows the work task "viewing".

The components of a work task PlugIn

CHILI PlugIns can be realized as additional work tasks. Therefore, they have
  - a button in the work task area,
  - a work space where the core functionality of the PlugIn is performed, and
  - a configuration card in the configuration work task.
This are the visible parts for the user. But additional interfaces to the main program exist.

The interfaces of the PlugIn

The PlugIn developers API (application programming interface) includes interfaces to the following areas:
  - *Graphical User Interface:* The PlugIn can display its graphical interface in a parent window of the main application. It is managed by the main loop of the latter. The user is supported for the programming of the visible part with the PIC widget (see below).
  - The *Icon Area* is also connected to the plugin: Keyboard and mouse events in the icon area (e.g. press on an icon) can be passed to the PlugIn. The PlugIn can also add icons to the icon area.
  - *Database:* Programmers have full access to the local CHILI databases. The database func-

tions hide the implementation details. Functions for read, write, delete, query, and retrieve are available.

- *Status Area:* This part of the screen is used for the information of the user. Messages, warnings and errors are presented here. The area can be shrinked to invisible when no warnings and errors are present. This area is automatically expanded when a warning or error messages arrives. The PlugIn can write messages to this area.
- *Work Task Bar:* The programmer defines an icon for the PlugIn which is displayed on a button here. The system switches to the Plugin work task when it's button is pressed.
- *PIC Widget:* This is an image display tool written in OSF/Motif. It displays a variety of medical images and has many intrinsic functions for image manipulations, measurements and annotations.
- *PIC Format Routines* are invisible for the user. The internal file format of CHILI is PIC which is a powerful extension to DICOM [3]. The PIC routines read and write data from disk, screen, memory or network.
- The *Configuration Card* is used to set specific parameters to the PlugIn. These cards are parts of the Configuration Work Task. PlugIns can be dynamically loaded and removed at runtime.
- *The CHILI Messaging System:* CHILI is a collection of many independent software components which communicate via a middleware systems. The middleware layer is realized with the message passing method. PlugIns are allowed to register for specific message classes, to read and write messages for or from other software components, such as import service, transfer service, receive service, database service, or multiplexer.
- *Events* of the CHILI system (e.g. work task visible or not, iconification of the system, kill signals) can also be passed to the PlugIn.
- *CHILI Style Guide and Documentation Guidelines* are under construction to enable the programmer of the PlugIn to create the same "look and feel" as the main program.

3.3 Certification of PlugIns

As CHILI processes sensitive patient data it has a strong security concept [4]. Therefore, we developed a procedure to certify PlugIns by a trusted party after a careful analysis of the source code. The certificate is a digital signature at the end of the PlugIn which proves that it has not been modified since inspection. This integrity check is realized by checksums and the digital signature [5], before the PlugIn is loaded.

4. EXAMPLES OF CHILI PLUGINS

Several *existing applications* have successfully been integrated into CHILI as useful additions to the main system. One example (figure 3) is a terminal window (xterm) which is connected to a Radiology Information System RIS (RADOS-M). This enables the user to see both the image data and RIS information on the same screen. Another example (figure 4) of such a simple PlugIn is a WWW browser, which allows the user to explore for example reference cases in the intra- or internet. The user manual of the system is integrated with the same mechanism. It is also possible to integrate computer games as new work tasks. This proved to be useful for users during boring night shifts who want to relax but have to remain in the teleradiology application for emergency cases.
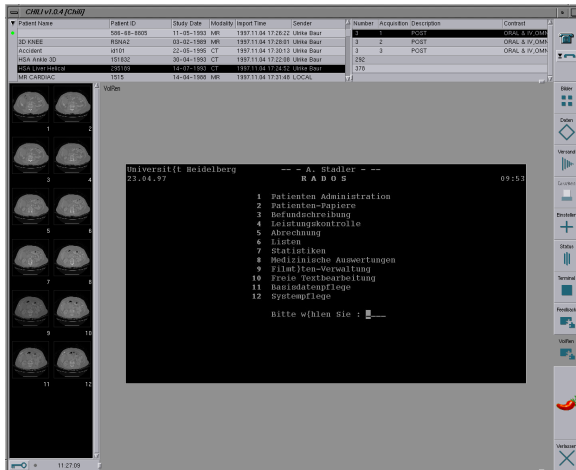
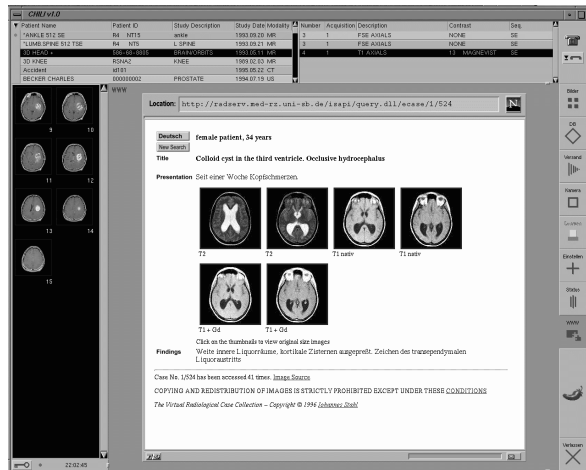Fig. 3.   A Radiology Information System as a CHILI PlugIn

Fig. 4.   Netscape (the "father" of the plug-in concept as a CHILI PlugIn

Additional functional modules for CHILI are currently under construction in several image processing research groups. One example is a 3D visualization tool which allows the rendering of 3D views of image sequences and the volumetric analysis [6]. Another group of researchers is developing a PlugIn for computer aided planning of liver tumour resections [7].
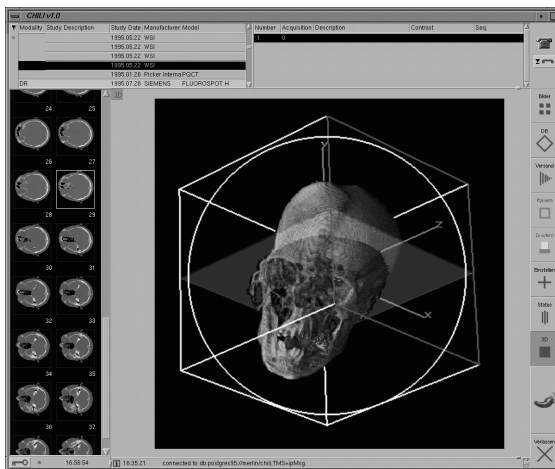

Fig. 5.   3D Reconstruction PlugIn

Fig. 6.   Liver resection planning PlugIn; Top row: 3D reconstructed liver, bottom row: tumor.

5. DISCUSSION

The first method, the integration of existing applications, has the drawback that no real data communication can be realized. The advantage of this method is that legacy programs can be integrated easily. Computer monitors can be saved and data of different applications can be viewed in *one* application on the same screen.The programming effort to integrate an existing application is about half an hour for an experienced programmer.

The advantage of the PlugIn mechanism is that the existing CHILI system has not to be

changed. New modules can be added even by end users. The programmer has a well documented interface which is supported by the product. The PlugIn can communicate with the other software components of the system and therefore is as powerful as the original software modules.

6. CONCLUSION

The CHILI PlugIn concept has advantages for developers as well as for end users. The first save a lot of development effort. The latter gain advantages as many different software components can be used in one environment. New components can be integrated into the existing context without the need to buy new computers. End user can even extent the system on their own. New image analysis methods can be brought easier and faster into clinical use.

7. ACKNOWLEDGEMENT

REFERENCES

[1]   Engelmann U. Schröter A, Baur U, Werner O, Schwab M, Müller H, Bahner M, Meinzer HP. Second Generation Teleradiology. In: Lemke HU, Vannier MW, Inamura K (eds): Computer Assisted Radiology and Surgery. Amsterdam: Elsevier (1997) 632-637.

[2]   Netscape Communications Corporation. Inline Plug-ins: http://home.netscape.com/comprod/products/navigator/version_2.0/plugins/index.html.

[3]   Schröter A, Engelmann U. ipPic Library. A portable library for the PIC image file format. Technical Report No. 75. Heidelberg: Deutsches Krebsforschungszentrum, abteilung Medizinische und Biologische Informatik (1995)

[4]   Baur HJ, Engelmann U, Saurbier F, Schröter A, Baur U, Meinzer HP. How to deal with Security and Privacy Issues in Teleradiology. Computer Methods and Programs in Biomedicine, 53, 1 (1997) 1-8.

[5]   Garfinkel S. PGP: Pretty Good Privacy. Sebastopol: O'Reilly & Associates Inc, 1995.

[6]   Evers H, Mayer A, Engelmann U, Schröter A, Baur U, Wolsiffer K, Meinzer HP. Volume visualization and interactive tools plugged into a teleradiology system. Proceedings of SPIE Conf. on Medical Imaging 1998. San Diego: SPIE (1998) 20.

[7]   Glombitza G, Lamadé W, Demiris AM, Göpfert MR, Richter G, Otto G, Lehnert Th, Bahner ML, Meinzer HP, Herfarth Ch. Computer aided planning of liver tumour resection. Caramella: 15th International EuroPACS Meeting, Pisa, 1997. Brussels: EuroPACS Association (1997) 215-218.