The Join of a Geographical Situation Display System and a Platform Independent C2 Information System

Gerhard Bühler & Heinz Faßbender
Research Establishment for Applied Sciences
Research Institute for Communication, Information Processing, and Ergonomics
Neuenahrer Straße 20
D-53343 Wachtberg-Werthhoven
Germany
Telephone numbers: (+49) 228/9435 376 & (+49) 228/9435 640
E-mail addresses: buehler@fgan.de & fassbender@fgan.de

Abstract

We describe concepts and experiences we have made in an ongoing project of our institute by joining a geographical situation display system and a platform independent experimental C2 information system. In implementing the joined system, the middleware standard CORBA and the internet programming language Java are applied such that also the joined system can be used totally platform independent and with minimal requirements to the system resources.

1. **Introduction**

By an ongoing project of our institute we join a geographical situation display system called **GSD** and the platform independent experimental C2 information system **INFIS** [Bühler & Faßbender, 1999] in such a way that the join fulfils the following two requirements:

1. The access to the integrated system is totally platform independent

2. with minimal requirements to the system resources.

The two requirements are realized by using

- internet technologies as web browsers and Java applets [Java] and

- middleware technologies as CORBA [CORBA]

Obviously, a system which fulfils these requirements, has maximal flexibility and the expense for applying and maintaining such a system is minimal.

In the following we want to describe the ideas we have produced and the experiences we have made in the described project and furthermore, we develop a general strategy for extending the platform independent C2 information system by additional functionality in such a way that the resulting system can also be used totally platform independently where only a very small amount of system resources is required.

For this purpose, we have structured the paper as follows: We start with an overview over the existing versions of the experimental C2 information system and the geographical situation display system. Than we design an architecture which enables the integration of the two systems into a unique system in such a way that it fulfils the two requirements above. Furthermore, we describe our experiences in implementing this architecture. Finally, we conclude and mention some further research topics.

2. **The Experimental C2 Information System INFIS and GSD**

In this section we present a short overview over the two systems which will be joined, where we only describe the components of the systems which are important for the joining process. We start with the description of the architecture of INFIS.

2.1 ***INFIS***

The experimental system INFIS is a distributed C2 information system. Its purpose is to support the work of head quarters of an army. Furthermore, INFIS serves as platform for flexible command and control information systems. It shall be incrementally extended by additional functionality. The join with GSD is one example for extending the functionality of INFIS and it shall illustrate how INFIS can be simply extended by using standard communcation mechanisms.

INFIS results from reengeneering the former experimental system EIGER which is described in [Bühler, 1998]. 90% of EIGER have been coded in Ada83. The other 10% have been coded in C and handle the communication of the system’s components. In [Bühler & Faßbender, 1999] we have developed the new implementation of EIGER which is called INFIS by applying Ada95 [Barnes, 1995], CORBA, and Java such that the system fulfils the requirements above. The architecture of the resulting system is demonstrated in Figure 1.

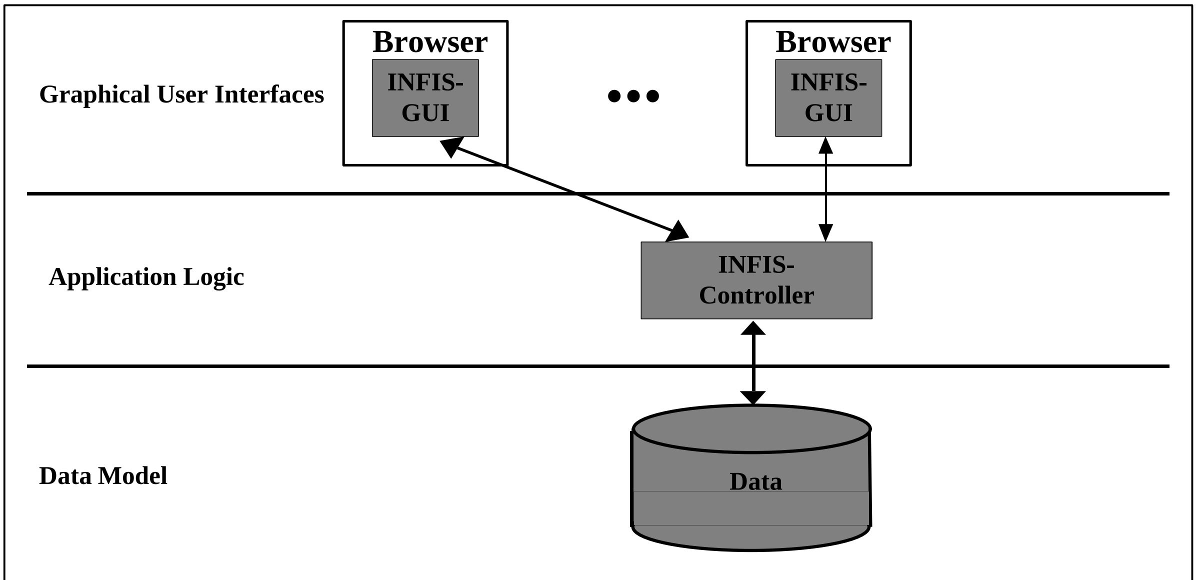

Figure 1: Architecture of INFIS

It is designed as 3-tier architecture corresponding to the model view controller design-pattern [Gamma *et al*., 1994], i.e. INFIS architecture consists of:

1. a *data base system* in the base level which is organized corresponding to the underlying ATCCIS [Wagner and Markmann, 1996] data model. By this fact, it is interoperable to other command and control information systems which are also organized by the ATCCIS data model.
2. a *controller* in the middle level which controls computations and communications of the other system components and to other command and control information systems. Furthermore, the controller implements the complete application logic of the system.
3. and at least one *graphical user interface* GUI in the upper level. The GUIs are implemented as thin clients which supports a simple and cost-efficient maintenance of the clients and thereby of the complete system.

The two requirements mentioned above are fulfilled by implementing the architecture of INFIS by the following two concepts:

1. The graphical user interface is implemented as *Java Applet*. Due to that, the complete system can be applied on every arbitrary system which includes a web browser.

2. The communication between the controller and the graphical user interfaces is handled by a *CORBA* connection. This also amplifies the platform independence of the system.

2.2 ***GSD***

Today the chosen GSD (*xIRIS*, established by Research Institute for Communication, Information Processing, and Ergonomics) is completely implemented in C++ on Windows systems. Its architecture is illustrated in Figure 2. In its original version, also GSD is a 3-tier architecture. But, since we need only the two uppermost levels, we only illustrate these levels in Figure 2. It consists of graphical user interfaces which are controlled by the GSD-Server.

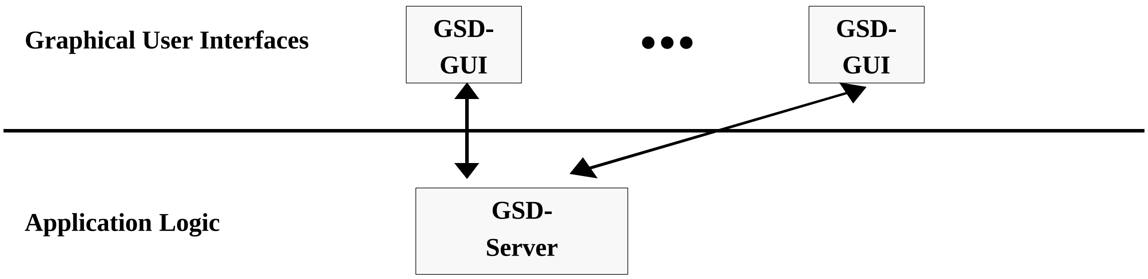

Figure 2: Architecture of GSD

Due to the fact, that the complete system is implemented in C++ on Windows systems, it can only be applied on Windows systems. For fulfilling the two requirements mentioned above, the system has to be reimplemented corresponding to the implementation of the architecture of INFIS by the following two concepts:

1. The GSD-GUIs are reimplemented as *Java Applet*. Then they can also be executed in every Browser.

2. The communication between the GSD-Server and the GSD-GUIs is handled by a *CORBA* connection.

In the following chapter, we describe how GSD will be joined with INFIS.

3. **The Join of INFIS and GSD**

The architecture of the join of the two described implementations of INFIS an GSD is illustrated in Figure 3.

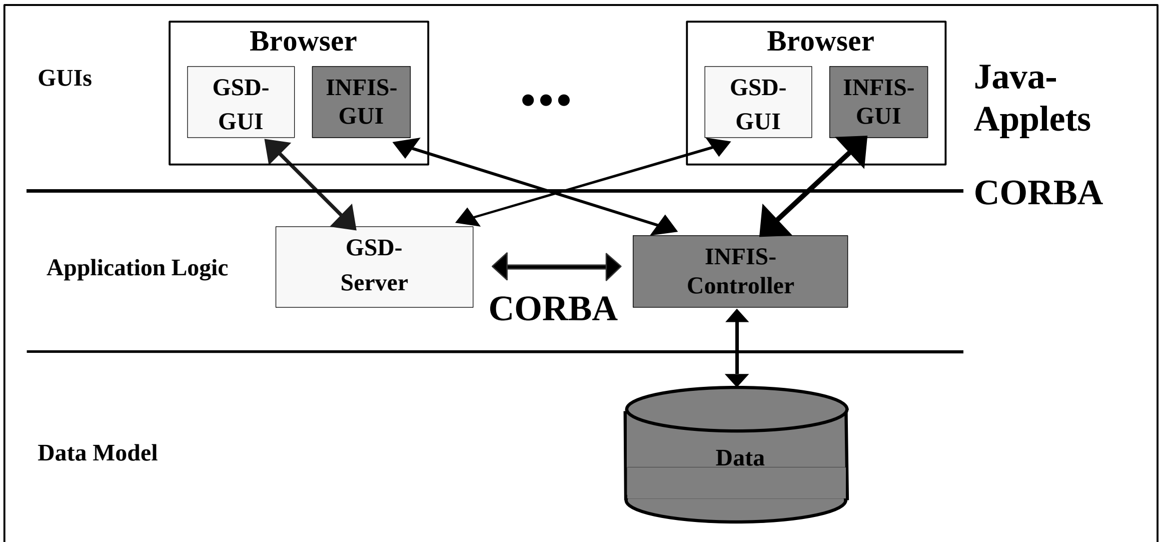

Figure 3: Architecture of the join of INFIS and GSD

Since the system is only an experimental system, the GUIs of GSD and INFIS are executed in parallel in the user's browser. In further developments we will consider an integration of the two GUIs.

There will be no immediate connection between the two corresponding GUIs. The only way of communication of the joined systems is the CORBA connection between the GSD-Server and the INFIS-Controller, i.e., if the user chooses the application of situation display in his INFIS-GUI, then a message is sent from the INFIS-GUI to the INFIS-Controller. The INFIS-Controller prepares the data which are needed to display the current situation. For this purpose, the INFIS-Controller consults its data base. Then it prepares the data in such a way that it can be sent to the GSD-Server by using services which are implemented by the GSD-Server. These services are specified as interface MapServer in CORBA’s Interface Definition Language (IDL) [CORBA].

The important components of the interface are specified in Figure 4.

```
interface MapServer      {

string LoadMap      (    in double left,
                         in double right,
                         in double top,
                         in double bottom,
                         in string description);

string InsertMilSymbol ( in string oid,
                         in MilSymbol milsymb);

string UpdateMilSymbol ( in string oid,
                         in MilSymbol milsymb);

string DeleteMilSymbol ( in string oid);

                         };
```

Figure 4: Interface MapServer in IDL

The interface MapServer specifies four functions which are implemented by the GSD-Server in C++ and which are called by the INFIS-Controller in Ada95. The two different programming languages at the ends of the CORBA connection are one important reason for defining a CORBA connection instead of e.g. an RMI connection. The four functions of the interface are implemented by the GSD-Server where in its turn it uses services of the GSD-GUI which are also defined as an interface in IDL. In the following we will describe the semantics of the four functions in the interface MapServer:

- LoadMap displays a map which can be specified either by the coordinates of its corners which are sent to the GSD-Server by the parameters left, right, top, bottom, or by the name or a description of the map in the fifth parameter description.

- InsertMilSymbol displays a military symbol as a unit, a feature, or a facility on the displayed map. This symbol is referenced by its unique object identification oid which is sent as first parameter of the function to the GSD-Server. The complete information which is needed to display the symbol, is sent in the second parameter milsymb. For example, this parameter contains information about the position and the type of a unit and display informations.

- `UpdateMilSymbol` updates a former displayed symbol. For example, if the position of a unit or feature has changed, then the function is called with the `oid` and with complete information for updating its presentation on the map in the parameter `milsymb`.

- `DeleteMilSymbol` deletes a former displayed symbol from the current map. For example, if a unit is no longer in the area of the map, then the symbol is deleted. The only information that is needed, is the unique `oid`.

Each of the four functions yields a string as return value. By this string exceptions or successful results in computing the function are sent to the INFIS-Controller.

4. **Conclusions and Further Work**

We have developed an architecture for joining a geographical situation display system and a platform independent experimental C2 information system. By this experiment, we have shown that the two systems can be joined and the whole system can be used totally platform independent with minimal requirements to the system resources. Furthermore, the experiment describes the following general approach for joining systems which are implemented in different languages and on different platforms:

Identify the system which is the server for the other one and describe an interface in CORBA's IDL that specifies the services which are implemented by the server and called from the other system.

In further research activities we want to integrate the two GUIs immediately into one single GUI in order to increase the comfort of the system handling. Nevertheless, we will not integrate the GSD-Server and the INFIS-Controller, since we favour a well-distributed architecture. Furthermore, for increasing the efficiency of the system, the GSD-Server and the INFIS-Controller may run on the same computer.

Additionally, we will checked the validity of the presented general approach by extending the joined system by further functionality, e.g., managing unstructured information like office applications (power point presentations, word texts, ...).

5. **References**

[Barnes, 1995] John Barnes . *Programming in Ada°95.* Addison-Wesley, 1995.

[Bühler, 1998] Gerhard Bühler. *Einsatz von Ada im Experimentellen Führungsinformationssystem EIGER.* in Workshop „Entwicklung von Software-Systemen mit Ada“, Bremen, Germany, Ada Germany, 1998.

[Bühler & Faßbender, 1999] Gerhard Bühler & Heinz Faßbender. *Making a C2 Information System Platform Independent by Using Internet and Middleware Technologies.* in: Proceedings of the Command & Control Research & Technology Symposium 1999, Naval War College, Rhode Island, USA, Vol.2, pp. 1098-1108, 1999.

[CORBA, 1999] see http://www.omg.org.

[Gamma *et al*., 1994] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley, 1994.

[Java, 1999]   see http://www.javasoft.com

[Wagner and Markmann, 1996] Karlheinz Wagner & Günther Markmann. *Interoperability Aspects of Command & Control Information Systems with Respect to International Standards and Emphasis on ATCCIS.* Report No. 469 of the Forschungsinstitut für Funk und Mathematik, Wachtberg-Werthhoven, Germany, 1996.