Structured software development: case study of a real time communication system

J.P. Santos, F.M.S. Ramos
University of Aveiro/INESC, Department of Electronics and Telecommunications, 3800 Aveiro, Portugal

Abstract

In 1990 INESC (Institute of System Engineering and Computers) was contracted by PT-Portugal Telecom - the Portuguese telecom operator - to develop the prototype of a telesurveilling system supported on ISDN (Integrated Services Digital Network), in the context of the R&D activities promoted by PT concerning the introduction of ISDN in Portugal. This paper includes a brief survey and discussion on the structured techniques evaluated for the system specification, and presents and discusses the analysis and design processes of the software developed to support the real time functionalities of the system (moving images and alarms) focused on the most complex of its components, the Central Telesurveilling Station.

1 Overview of the ISDN based telesurveilling system

The purpose of the telesurveilling system is to enable the visualisation, in a central surveilling station, of images collected from a video camera located on a remote place. The architecture of the telesurveilling system developed is based on two types of stations: central surveilling station (CS) and remote surveilling station (RS); both types of stations are based on standard PC compatible computing platforms up-graded with an image grabber and an ISDN (2B+D) interface card.

The communication between a CS and a RS is supported on a circuit switched ISDN B channel (64kbit/s) used to transfer image data from the RS to the CS, and on the packet switched ISDN D channel (16kbit/s) used to exchange end to end signalling information (UUS-user to user signalling capability), such as alarm indications from a RS to a CS, or RS configuration commands from a CS to a RS.

This system allows the transfer of two types of images between a RS and a CS: 144x180 pixel b&w moving pictures with a rate of 8 images/sec, or

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

288x360 pixel still JPEG colour images, with a maximum transfer rate of 1 image/sec.
The connection between a CS and a RS can be setup by any of stations: if the initiative is from the RS, it results from the activation of one of its alarms (such as fire, water or movement detectors); if the initiative is from the CS it results from a request from the operator.
When a connection between a CS an a RS is setup, the CS must be able to accept alarm indications coming from others RSs: in this case the operator should be informed of the reception of each, and must decide if the current connection should or not be terminated and a new connection, with the RS that generated the new alarm, should be established.
Figure 1 shows the basic components of the architecture of the system described.

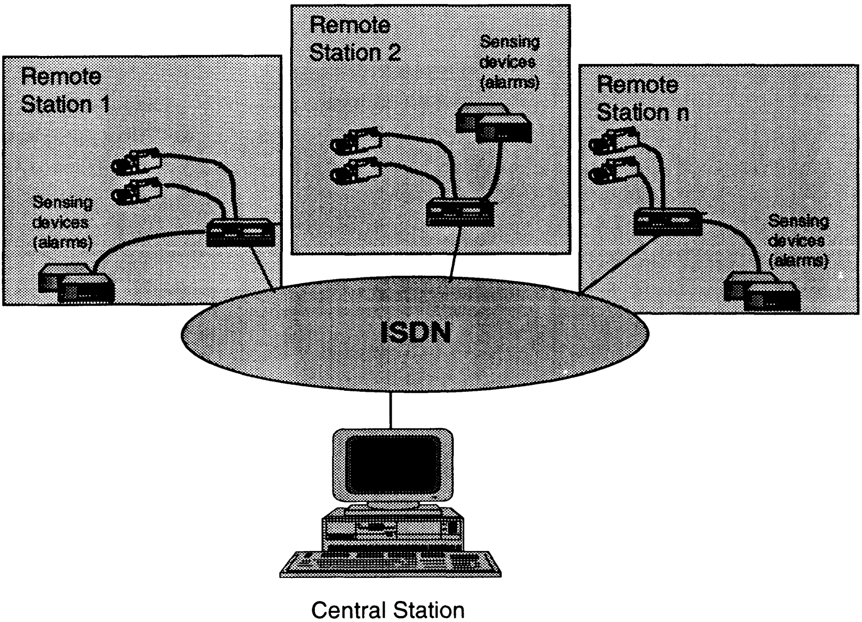

Figure 1: Architecture and basic components of the telesurveilling system.

From the functionalities of the system results clear that several types of real time events must be handled by a CS, namely bi-directional communications (reception of alarms and images and transmission of commands), alarm processing, image reconstruction and display and interaction with the operator. In a similar way a RS should be able to handle different real time events, including bi-directional communications (transmission of alarms and images, reception of configuration commands) and acquisition and compression of images.

This paper describes and discusses the process of software analysis, design and implementation of the system developed, with emphasis in the software of the CS.

2 Structured methodologies for software specification

The use of formal methodologies in the analysis and design of software is mandatory to enable the rigorous and complete description of the specification requirements, and is particularly useful when, as it is the case under description, the functional requirements include the handling of several different types of real time events.

The analysis phase of a system includes the definition of its interface, that is the listing of all the interactions that may occur, between the system and its external world, at the system's boundaries and the identification of the data and the data formats that are related to each event, that should be processed and/or stored by the system.

Concerning the analysis, three methodologies where evaluated for use in this system: DeMarco [1], Gane-Sarson [2] and Yourdon [3].

DeMarco and Gane-Sarson propose that the first stage of the analysis of a new system is the development of the New Logic Model, that is a description of the system specifications based on the following common set of tools:

Data Flow Diagram (DFD): graphical tool used to describe the flows of data at the system's boundaries and the general processing operations that must be performed with the input data; this diagrams are built using four different types of entities: data flows, processes, stores and external entities.

Data Dictionary (DD): is an organised listing of the data flows, stores and external entities used in a DFD, and provides a comprehensive description of the entities used in a particular DFD.

Process Specification (PS): the process specification is the detailed description of the processing operations that each entity identified in a DFD must perform based on its specific input data; for this specification several types of tools may be used in conjunction, ranging from structured natural language (such as "structured" English) to state transition diagrams.

For the description of the stores and related data structures DeMarco proposes de use of Data Structure Diagrams, and Gane-Sarson suggests the use of the Immediate Access concept and the use of the 3FN-Third Normal Form, the latest resulting in a specification with a higher level of organisation and consistency than the obtained through the use of the Data Structure Diagrams.

In the next step of the analysis process, both DeMarco and Gane-Sarson propose the development of the New Physical Model, that includes the definition of the man machine interface and the identification of the possible physical implementation's models. Following this step, the actual physical model to be used should be elected among those identified, using as main

criteria the cost-benefit analysis and the effort of development required in the implementation.
The DeMarco and Gane-Sarson methodologies are commonly referred as being Classical Structured Analysis methods. These methods are very time consuming namely when the system under specification is an evolution of an existing system, do not include, in a practical way, the capability of description of real time systems and do not have a satisfactory degree of abstraction in the specification of the data structures and databases required for the appropriated description of the system under analysis.
In the 80's, authors such as McMenamim & Palmer [4], Ward & Meller [5] and specially Yourdon [3], introduced news concepts that helped to overcome the disadvantages of the Classical Structured Analysis methods. In this new approach, known as Modern Structured Analysis, the analysis process of a new system is based on the definition of the Essential Model, that is formed by the Environment Model and the Behaviour Model. In the Environment Model it is defined the interface between the system and the external world (events and basic data flow), and in the Behaviour Model it is described the internal behaviour of the system through the use of DFDs, DDs, PSs and a new complimentary tool:

Entity Relationship Diagram (ERD): diagram that provides the specification of the data entities in the system and the relationships among those entities.

Furthermore Yourdon proposes for the PSs the use of State Transition Diagrams and the Pre-post Conditions:

State Transition Diagram (STD): diagram that identifies the different states a system may assume, its sequencing and the conditions that define the changing among the different system's states.

Pre-post Conditions (PPC): define, in a very high level of abstraction, the data flows resulting from the occurrence of the different possible events.

The STDs provide a powerful tool for the description of the control processes, and are essential in the specification of a real time and interrupt driven system. The ERDs allow a flexible and very high level definition of the stores, data structures and organisation of the databases within a system, thus being a very useful tool in the analysis process.
Following the definition of the Environment Model, and as a basis for the detail of the system's specification, it is required the definition of the Context Diagram that identifies the part of the system (the whole or a part) that is going to be specified.
The Modern Structured Analysis also includes a new concept in the system's analysis that is the Event Partitioning, alternative to the Top Down approach used in the Classical Structured Analysis methods. Event Partitioning induces the analysis of a system based on the description of the way the system must behave when an event occurs; this approach enables an easier description of the system's kernel than the one provided by the Top Down

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

approach, particularly in the case of real time event (interrupt) driven systems.

Concerning the design phase of a system, Yourdan [3] proposes the creation of the following models: the System Implementation Model, that includes the Processor Model and the Task Model, and the Program Implementation Model.

The Processor Model is a description of the strategy of allocation of the Essential Model, defined in the analysis phase, by the different processors/machines available at the implementation phase, and the Task Model defines for each processor/machine the corresponding tasks (both in single task or multitask operating environments). Each task is then specified in detail in the Program Implementation Model, using a graphical tool named Structure Diagram [6].

3 Case study: application of structured methodologies to the specification of the ISDN based telesurveilling system

Figure 2 presents the Environment Model of the system under analysis in this paper, whose functional specifications where briefly described in paragraph 1:

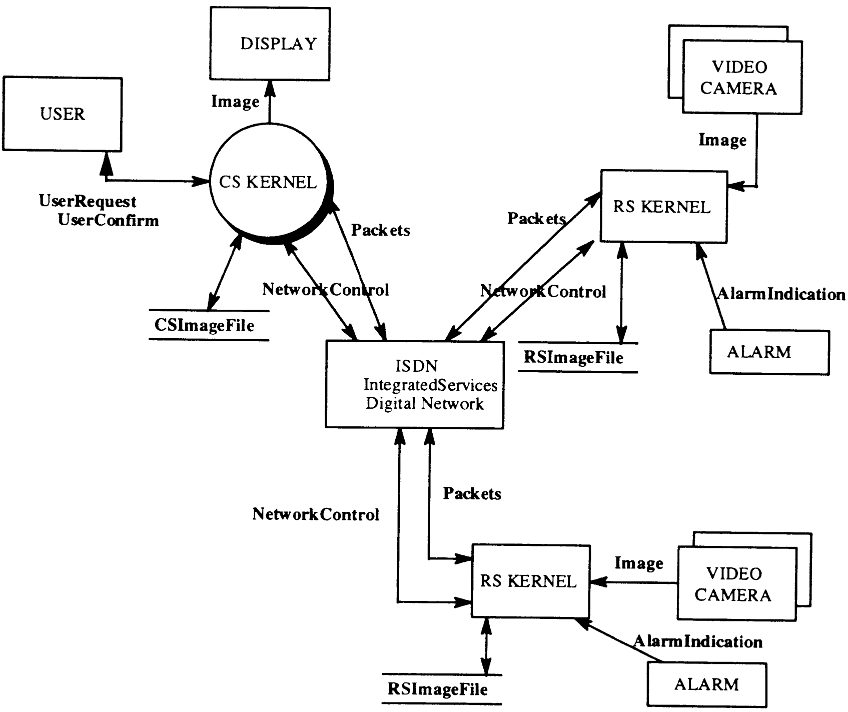

Figure 2: Environment Model of the ISDN based telesurveilling system.

The images collected, through an image grabber, from the video cameras located in the RSs are compressed in the corresponding RS, after what they are transmitted to the appropriated CS. The transmission process of each image may include the transmission of one or more packets, and is performed via the ISDN interface card. If a connection does not exist between a RS and a CS, the RS should store the images locally, for eventual future retrieval, in the store *RSImageFile.*
In the CS the image data is receive from the network through the CS ISDN card, and are displayed after being decompressed. The user may additionally request the storage of the images in the CS, what is done in the *CSImageFile* store.
CSs and RSs may also exchange others types of information, such as configuration commands and indication of alarms.
Due to the space available, the following paragraphs are focused on the CS analysis and design processes. This is expressed in the Context Diagram shown in figure 3, that highlights the part of the Environment Model that is going to be subject of detailed analysis. At this stage it must be referred that all the libraries related with the hardware components (communications, display and decompression) of the system where already available at the beginning of this project, so the analysis basically refers to the kernel of the system.

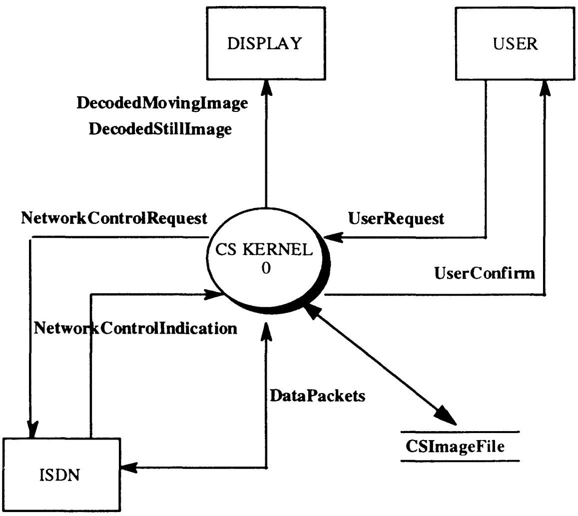

Figure 3: Context Diagram of the CS.

The input messages and its corresponding answers associated with the Context Diagram are described in the DD. Those messages can be classified as events generated by the user (CS operator) and received from a RS. The events generated by the user may be of four different types: CS configuration request, RS configuration request, operation and display. The events received

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

from a RS may be of two different types: indication of an alarm or reception of an image data packet. Furthermore the Environment Model also includes the list of the events of the system.
The Behaviour Model describes the internal behaviour of the system based on the list of events; for each event an intermediate DFD was constructed (Event Partitioning) to identify the required data flows and the way the data should be processed and/or stored. For this purpose tools such as DFDs, DDs and PSs were used. Figure 4 shows the DFD (DFD 0) constructed to describe the EC KERNEL identified in the Context Diagram of the CS, that was built based on the intermediate level DFDs. In this DFD there is a detailled description of the control entities required, whose need resulted from the identification of the events that may occur.

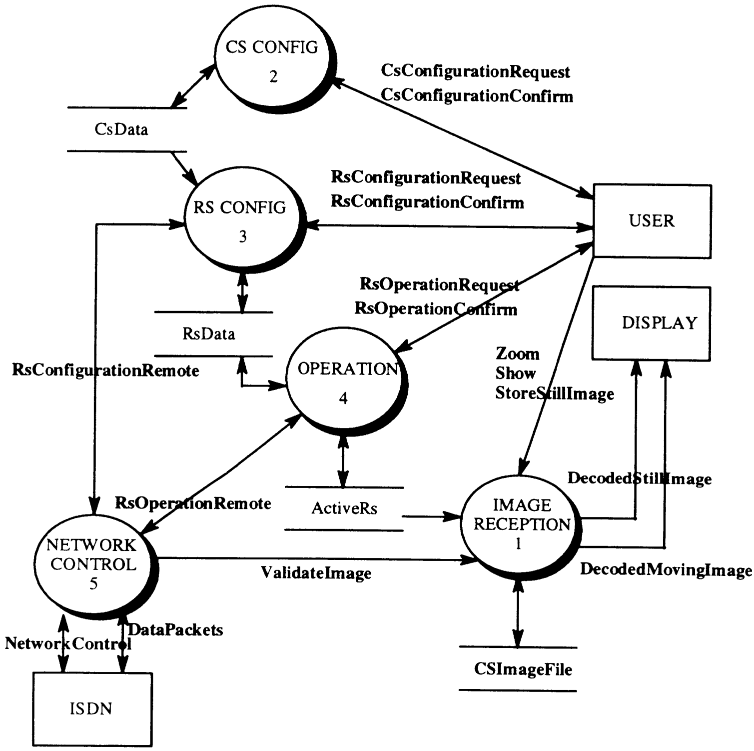

Figure 4: DFD 0 - CS KERNEL.

Figure 5 shows an example of entry of the DD related to the DFD 0. The DD includes similar entries for all the data structures existent in this system.

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517


```
ValidateImage: reception of image from the RS.
        ValidateImage= 5555
                + CameraNumber = {byte}1        = 0x01-0x08
                + ImagePartNumber = {byte}1     = 0x00-0xff
                + Image                         = 1{byte}26000
```

Figure 5: An example of a DD entry.

The Process Specification techniques used included the Pre-post Conditions and "structured" English. Figure 6 shows an example of PS using these techniques applied to the process 1 - Image Reception presented in DFD 0 - CS Kernel (figure 4).

```
BEGIN
    IF StoreStillImage
        Copy image from file (default.jpg) to filename specified
    ELSE
    BEGIN
        IF ValidateImage WITH  CameraNumber=ActiveCameraNumber
                IF ActiveCameraNumber= 1 or 2
                        Transfer compressed still image to the image card
                        Store image into file (default.jpg)
                ELSE (* 2 < ActiveCameraNumber < 9 *)
                        Transfer compressed moving image to the image card
                ENDIF
        ENDIF

        IF Show
                Open file
                Transfer compressed image to image card
        ENDIF

        Start decompression of image
        DO WHILE (descompression in progress)
        ENDDO
        Transfer image from the image card  to memory
        Display image
    END
END
```

Figure 6: Example of Process Specification using "structured" English.

After the analysis process was completed it was started the design process. Because the CS was supposed to be implemented in a single processor/single task machine, there was no need to define the System Implementation Model. Figure 7 shows a simplified version of the initial Structure Diagram of the Program Implementation Model of the CS kernel.

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

In this diagram five main modules were considered that directly map to the DFD 0 - CS kernel (figure 4).

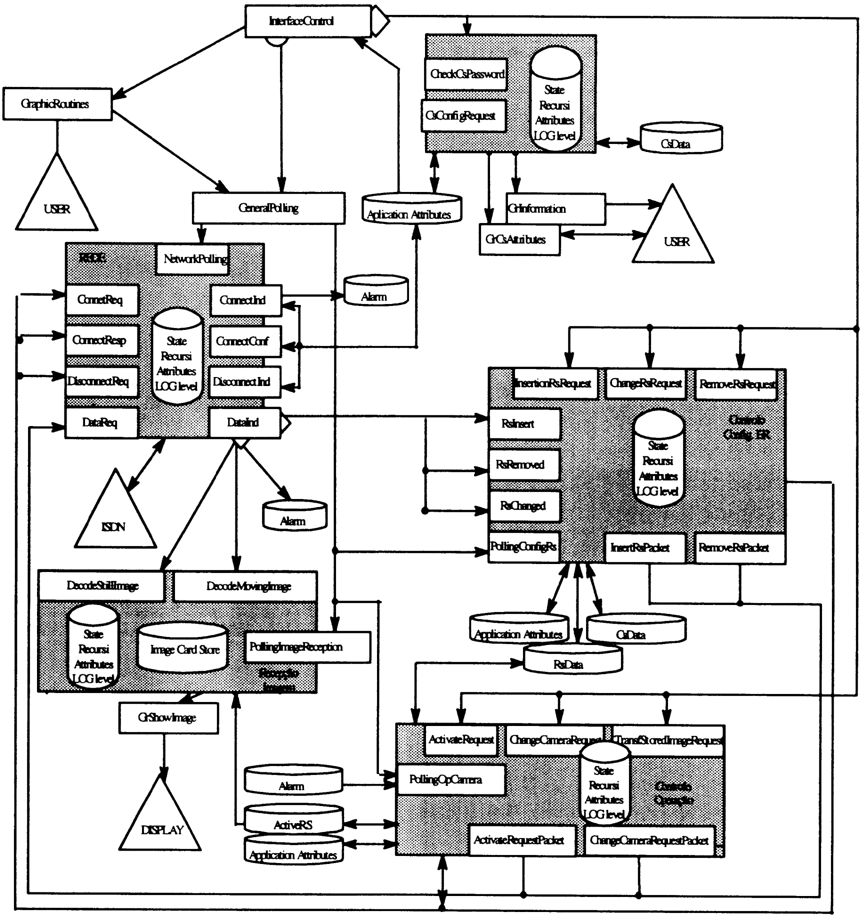

Figure 7: Initial Structure Diagram of the Program Implementation Model of the CS kernel.

In this figure aditional new modules, sub-modules and stores are shown that are not present in the Context Diagram: they resulted from the need to construct the synchronous model required for the implementation from the asynchronous model of the analysis.
For each of the modules identified further refinement was produced, using the Structure Diagram technique exemplified above.

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

4 Conclusions

From the experience of application of structured techniques to the specification of an ISDN based telesurveilling system, experience that was very briefly described in this paper, many conclusions concerning the advantages and the disadvantages of its use may be listed. Let us focus on the most important ones.

First conclusion, perhaps obvious but that we think that must be emphasised, is that the use of structured techniques is absolutely mandatory in the different phases of the specification process of a system. This is particularly important in systems with significative concurrence of events, where avoidance of loss of data and time constraints are definitive requirements. The use of such techniques is a condition to the minimisation of the effort required to produce, maintain and disseminate specification documents.

A very difficult issue to consider in the application of these techniques is how deep in the system's analysis and design they should be used. We do not have a definitive answer to this question, but our experience tells us that the use of these techniques is specially useful in the high level specification of the system. From a time vs. effectiveness point of view, and this is particularly true in the PS activity, the further we proceeded in the specification process the less we considered them appropriated. It should be referred that the use of the PPCs revealed to be a very well balanced solution to this tradeoff.

One essential aspect of the specification process is the rigorous definition of the data structures, stores and organisation of the databases. For this purposes the DDs were essential, because they provide a complete reference of all the data entities within a system, and are very useful particularly when the system is being specified and implemented by a team of experts.

Although the Modern Structured Analysis techniques applied provided an efficient mean of specification, they revealled some limitations concerning the description of complex real time aspects of the behaviour of the system. This was particularly true in the description of complex interactions among the control processes and the external entities, such as those with recursivity problems, that required a very significative amount of specification effort.

Concerning the design phase we found the structure diagrams very convenient and relatively easy to produce and maintain. These diagrams are very useful in the implementation phase, because they induced the description of the system in a way very close to the future implementation. Furthermore they provided a very convenient mean to describe particular aspects not completely covered in the analysis process, such as recovery procedures, exception handling and platform dependent implementation issues.

Transactions on Information and Communications Technologies vol 11, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

References

1. DeMarco, T. *Structured Analysis and System Specification*, Prentice-Hall, Englewood Cliffs, N.J., U.S.A., 1979.
2. Gane, C., Sarson, T., *Structured System Analysis: Tools and Techniques*, Prentice-Hall, Englewood Cliffs, N.J., U.S.A., 1979.
3. Yourdon, E., *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, N.J., U.S.A., 1989.
4. McMenamim, S., Palmer, J., *Essential System Analysis*, Yourdon Press, 1984.
5. Ward, P., Meller, S., *Structured Developemnt for Real-Time Systems*, vol. I, II and III, Yourdon Press, 1985.
6. Martin, J., MacClure, C., *Structured Techniques: Basis for CASE*, Prentice-Hall, Englewood Cliffs, N.J., U.S.A., 1988.