REAL-TIME ARCHITECTURE FOR A LARGE DISTRIBUTED SURVEILLANCE SYSTEM

M. Valera*, S.A. Velastin†

Digital Imaging Research Centre, School of Computing & Information Systems, Kingston University (UK), *m.valera@kingston.ac.uk, †sergio.velastin@kingston.ac.uk, *Fax: +44 208 547 7824

Abstract

The latest generation of surveillance systems can be categorised as concurrent, distributed and large real-time systems. The most common and well-known approach to design these systems is based on Object Oriented technology. In this paper we present another approach to design an intelligent distributed surveillance system. The approach method is known as "real time network approach" or MASCOT, a design method for designing and implementing large real-time concurrent systems. The basic notion is that the flow of data through the system is controlled solely by a set of concurrent processes. The paper introduces the fundamental concepts of MASCOT and its possible contributions to the creation of good designs for such intelligent distributed surveillance systems. This paper finishes with the illustration of a distributed real time surveillance system using this approach.

1 Introduction

The technology of vision surveillance systems has evolved over recent years [17]. "First generation" video-based surveillance systems started with analogue CCTV systems, which consisted of a number of cameras connected to a set of monitors through automated switches. Technological improvement of these systems led to the development of semi-automatic systems, called "second generation" surveillance systems. These systems are able to attract the attention of the human operator by displaying relevant messages. Current research is towards the design of wide-area automatic surveillance systems (third generation). The usual design approach of these advanced vision systems is to distribute sensors over wide areas. This distribution, from the computational point of view, consists of spreading the processing capacities over the computer network and the use of embedded signal processing devices to give the advantages of scalability and robustness offered by distributed systems.

Thus, "third generation" surveillance systems can be categorised as concurrent, distributed, embedded, real time systems. An important aspect of these systems is inherent temporal diversity (heterogeneous timing), arising from the variety of timing requirements from the different response times and processing rates of the functional elements of the systems, and from the parallelisation and distribution in the implementation architectures. Moreover, embedded real-time systems are often naturally asynchronous. However, currently the different parts that compose a complete surveillance system, e.g. computer vision, control, storage and retrieval modules, are usually designed in a sequential and synchronous manner using object-oriented models.

For example, CORBA is the Object Oriented middleware system normally used to deal with the distribution of these systems. The design of processes in a synchronous manner, and the run-time overhead that object oriented and CORBA approaches produce, may cause communication bottlenecks or exhaustion of resources, either from the network communication point of view or caused by unpredictable behaviour of some components of the system.

As system size and diversity grow and consequently complexity increases too, the probability of inconsistency, unreliability and non-responsiveness increases. Therefore the design and implementation of distributed real-time systems present significant challenges to ensure that these complex systems operate as required. In order to understand or implement any complex system it is necessary to decompose it into component parts and functions [4][5][7][8]. Therefore, it is natural to think of distributed systems in terms of independent concurrent activities that need to exchange data in ways that do not undermine the overall predictability and performance of the system.

The paper proposes the use of an alternative approach as a design method for intelligent distributed surveillance systems. The approach is known as MASCOT (Modular Approach to Software Construction Operation and Test) [7][15][16], through which large distributed real-time concurrent systems can be designed and implemented. It is a well-suited method for real-time systems [4][7], since it deals specifically with structuring a system into tasks and defining the interfaces between them. MASCOT is particularly well suited for real-time embedded applications where the software is complex and highly interactive. It uses the concept of a data flow network between concurrent processes as the medium for expressing software structure.

2 A brief review of some Intelligent Distributed Surveillance systems

Current research is centred on automatic surveillance systems that are referred to as a "third generation" or Intelligent

Distributed Surveillance systems. These systems are based on the distribution and separation of the processing tasks into a low level and high level due to the proliferation of devices called DSP (Digital Signal Processors), which allows the building of intelligent (or "smart") cameras with autonomous processing capacities [13]. By distributing the processing capacities over a computer network and using embedded signal processing devices, it is possible to build systems with improvements in scalability and robustness.

Several vision surveillance systems have been designed and developed within the academic and commercial world. One of the classical distinctions between different surveillance applications refer to their use indoors or outdoors. This distinction occurs because of the differences in the design at the architectural and algorithmic implementation level. The topology of indoor environments is also different from that of the outdoor environments. At the algorithm level, the former are usually built in relatively small spaces that are separated by walls and communicate with each other through corridors and doors. The latter are more challenging due to the variability and sometimes poor quality of lighting conditions (e.g. in [6] the system is designed especially for night-time outdoors surveillance or poor light conditions).

DETER [9] (Detection of Events for Threat Evaluation and Recognition) is an example of an outdoors surveillance systems. DETER is meant to report unusual moving patterns by pedestrians and vehicles. In order to do this, the system fuses the view of multiple cameras into one view and then performs the tracking of objects. Other examples of outdoors surveillance systems can be found in [14][10][11].

CCN [5] (Cooperative Camera Network) is an indoor application surveillance system that consists of a network of nodes. Each node is composed of a PTZ (pan-tilt-zoom) camera connected to a PC and a central console to be used by the human operator. The system reports the presence of a visually tagged individual inside the building. Its purpose is to monitor potential shoplifters in department stores using the assumption made that human traffic is sparse. However, this design night not work for crowded stores (e.g. during sales seasons). CCN and DETER are systems developed by Honeywell Laboratories.

Another distributed surveillance system is that developed under an EU-funded research project called PRISMATICA [12][1] (Pro-active Integrated Systems for Security Management by Technological Institutional and Communication Assistance).

The main goal of PRISMATICA is to detect certain types of behaviours, which are defined from public transport management requirements. It is not only a wide-area video-based distributed system like ADVISOR [1][11], but it is also a wide-area multi-sensor distributed system; receiving input from CCTV, smart cards, local wireless networks and audio sensors. PRISMATICA connects all the inputs extracted from the different sensor processors through a network (wide bandwidth distribution network) to a main computer system for system coordination, user interface and recording of events that affect safety and security in a public transport infrastructure. Thus, it handles global information coming from different kinds of devices. The CCTV and wireless cameras are connected to a video matrix and to a framer server (video matrix), which is connected to the main computer (the server) and the intelligent camera systems. The server then routes the video signals to the intelligent camera systems. The server also controls the whole system and acts as a user front-end. Similarly to ADVISOR (see below), PRISMATICA is a modular and scalable architecture approach using standard commercial hardware.

ADVISOR (Annotated Digital Video for Intelligent Surveillance and Optimised Retrieval) assists human operators by automatic selection, recording and annotating images with interesting events in it. In other words, ADVISOR interprets shapes and movements in scenes being obtained by the CCTV in order to build up a picture of the behaviour of people in the scene. ADVISOR stores all the video output from cameras. In parallel with recording video information, the archive function stores commentary of associated sequences (annotations). The archive can search for video sequences, which match keywords in the annotation data, or according to specific times. Retrieval of video sequences can take place alongside continuous recording. ADVISOR is an open and scalable architecture approach and is implemented using standard commercial hardware with an interface to a wide-bandwidth video distribution network.

Although both systems are classified as distributed architectures, they have a significant main difference in that PRISMATICA employs a centralised approach whereas ADVISOR can be considered as a semi-distributed architecture. PRISMATICA is built with the concept of a main or central computer that controls and supervises the whole system. ADVISOR can be seen as a network of independent dedicated processor nodes, avoiding a single point-of-failure at first sight. Nevertheless, in each node there is a central computer, which controls the whole node. Therefore, there is a single point-of-failure within each node. The number of CPUs in each node is directly proportional to the number of existing image processing modules or tasks.

In [3] the authors report a surveillance system with no server to avoid this centralisation, making all the independent subsystems completely self-contained, and then set up all these nodes to communicate with each other without having a mutually shared communication point. This approach avoids the disadvantages of the centralised server, and moves all the processes directly to the camera making the system a group of smart cameras connected across the network.

3 MASCOT

Going through the literature it is possible to see that there is much of research and work done on new specific vision algorithms, i.e. tracking, detection or segmentation. In terms of a surveillance system as a whole (video and/or multi-sensor) we have mentioned some examples that use different architectural approaches: central, semi-distributed or distributed. In general, these are still relatively small systems. To successfully build a large real-time system requires robust design methods capable of encapsulating the different levels of abstraction that need to be handled (from a global view of the system to the detailed implementation aspects).

There are several design methods for real-time systems such as structured design methods (SD), Jackson System Design (JSD), MASCOT or Object Oriented design (ODD). Each of these design methods emphasises one set of criteria to characterise the components of the systems [5] e.g. procedural modules in structured design, concurrent tasks in MASCOT, or objects in OOD.

There are four important objectives that design methods for real-time systems should accomplish: to be able to structure the system in concurrent tasks, the capability of developing reusable software through information hiding, to be able to define the behavioural aspects of the system and be able to analyse the performance of the design by determining its performance and the fulfilment of requirements. MASCOT imposes a disciplined approach to design, achieving these objectives in an efficient way.

Jackson and Simpson originated the essential concepts of MASCOT at the UK Royal Signals and Radar Establishment (RSRE) during the period 1971-5. MASCOT has been an evolving method and the latest form is MASCOT-3 [2]. MASCOT is neither a language nor an operating system although it includes elements that are related to both of them [7]. MASCOT brings together a co-ordinated set of tools for dealing with the design, the construction or system building, the operation or run-time execution and testing software.

The basic notion of MASCOT is that the flow of data through the system is controlled solely by a set of concurrent software processes [2][5][16][15]. These processes are known as ACTIVITIES. The data is moved and transformed by activities. Consequently, MASCOT activities need to co-operate with each other by passing data but without direct communication. The communication is provided by special modules called 'Intercommunication Data Areas' or IDAs. IDAs are passive components, which exist only to satisfy the intercommunication requirements of the active components (activities). Therefore, the designer can design in terms of concurrent tasks (activities), which are purely sequential, and IDAs, which even though they are passive components, encapsulate the interactions between activities.

Designs in MASCOT are expressed in a hierarchical manner rather than in terms of a flat network. Therefore a MASCOT system consists of a network of activities and IDAs. A "system" is the outermost level of the network design, which encompasses the whole of the application. A system differs from a subsystem only in having, by definition, no external dependencies other than those that may be satisfied during system building [16][17]. The main consideration in decomposing a software system into concurrent tasks or activities relates to the asynchronous nature of the functions within system.

The MASCOT method is formal enough to give the visibility necessary to support management and control of the design during development and subsequent maintenance. This visibility can be achieved by the use of CASE tools to process the design, supported by a database to hold the design details, so it provides the status progression feature of MASCOT.

4 MASCOT to design Video-based surveillance systems

The MASCOT method provides a design language or textual form and a graphical annotation or ACP (Activities Channel Pool) diagram or MASCOT network diagram. Both forms can be used to control and evolve design structure. In MASCOT the designer should aim to get the MASCOT network diagram (i.e. the system representation) right and then subsequently to modify or derive any less-detailed system description. In this section we present an example of a real-time surveillance system designed using MASCOT. The example system, called CSS (Control Surveillance System), consists of a data processing module called DPM and a control module called CM (see Figure 1).

In this example we consider that the DPU is a group of image processing functions that deals with image data or information related to image data. These functions represent the common basic image processing functions that are found in surveillance systems like ADVISOR, PRISMATICA [12][1]. The functions are: monitor inputs (capture of the information from sensors), control analysis data (e.g. segmentation, motion detection, tracking, semantic interpretations of the low-level data) and database (storage). The CM subsystem consists in two main functions: the control function of the DPM (control unit) and the monitor output, which interacts with the user, see Figure 2 and Figure 3. At this point, it is interesting to point out that even though we consider in this paper a CSS as an example of a small real-time surveillance system, it is possible to think of the CSS as a self-contained or independent node of a hierarchical network infrastructure, which makes up a large-scale surveillance system.

A design approach that has been chosen is to consider the functions that exist in a CSS system in terms of IDAs, subsystems and activities, see Figure 4. These activities are represented as circles and the subsystems as rounded squares, communicating between each other through IDAs. There are three different kinds of IDAs used in the design example: pool, channel and signal. The *pool* represents the concept of reference data, *channel* represents the concept of message data, and *signal* signifies event data (see Table 1).

By using MASCOT concepts, the design of the distribution solution is simple and straightforward, because in MASCOT the distribution problem is a matter of physical mapping phase to hardware but it is not a matter of design. That means, the decision of the distribution of the tasks in the same processor or in a different one, does not correspond to the design phase in MASCOT but to the implementation and test phases.

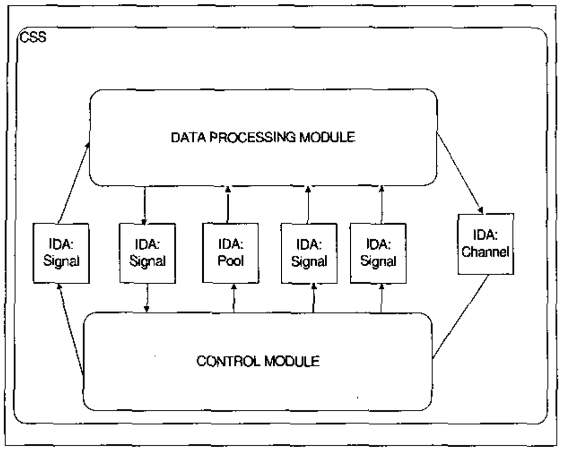
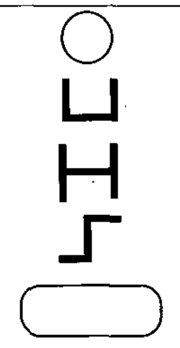


Figure 1: MASCOT Network diagram. It consists of subsystems and IDAs and it represents the outermost level of the system.



	Activities
	Pool: reference data
	Channel: message data
	Signal: event data
	Subsystems

Table 1: Symbols for the components used on the designs

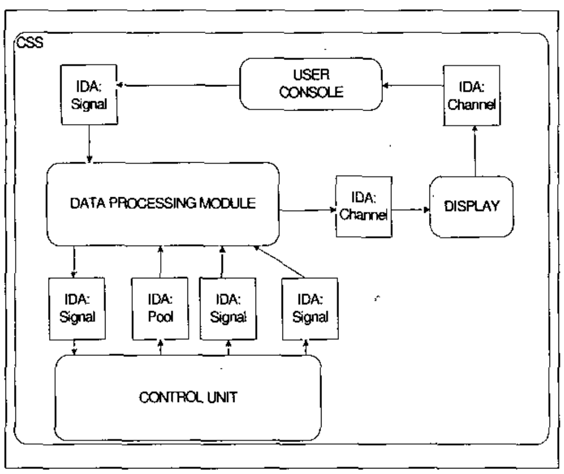


Figure 2: Network diagram representing the decomposition of the CM.

MASCOT annotation gives a good "visibility" of the design by providing a clear picture of the system to be built. Therefore, it provides an effective focal point for discussions within the design team during the initial stages of the design. In our example, it is possible to comprehend at first sight, which is the control process task (see Figure 3) and which is the image process node or module (see Figure 2), and how these two important process units (subsystems) interact between them (see Figure 1).

It is possible to see inside the image-processing module or subsystem, how many initially subsystems and activities there are (see Figure 3 and Figure 4). It is feasible to see how these activities communicate between them through IDAs, giving the possibility of control the timing interaction between the communication of the activities by using different kinds of protocols, which are defined in MASCOT (see Figure 4). Also, it is possible to see how the data flows between the activities and what type of data flows between them through the definition of these hiding modules called interfaces.

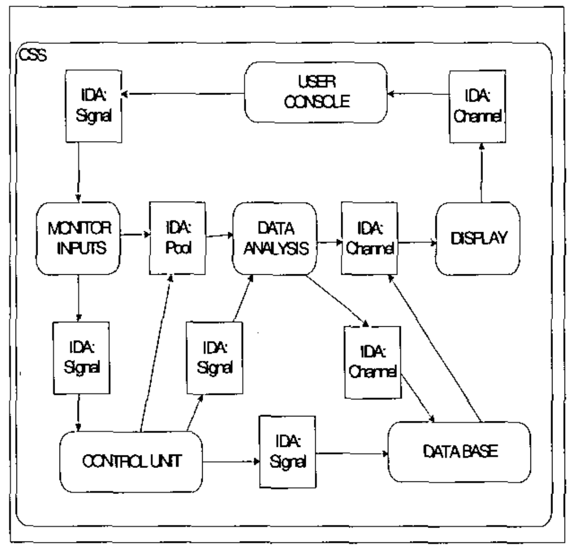


Figure 3: Functional description of the CSS system.

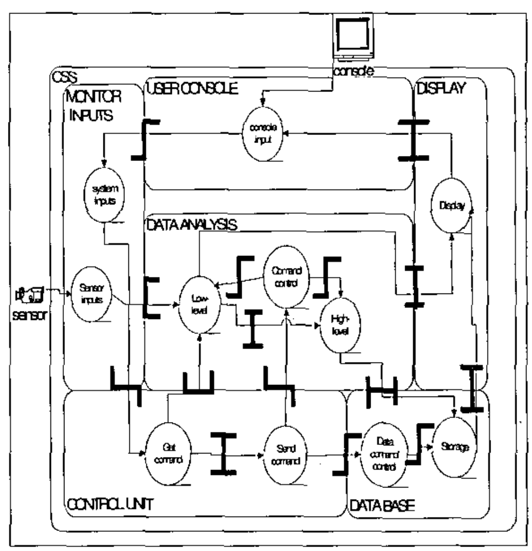


Figure 4: ACP diagram of the CSS system

5 Conclusions

In this paper we pointed out the need to apply design methods for designing intelligent distributed multi-sensor wide-area surveillance systems. One of the main reason for this is because surveillance systems can be categorised as large-scale real-time concurrent distributed systems which need to be built using a real-time design method for their great complexity and high interaction.

In this paper we do not intend to present a comparison between different design methods, i.e. MASCOT versus Object Oriented Design or versus other real-time design methods. On the contrary, we present MASCOT as a possible method for these surveillance systems, because it can make a contribution to the real design of these systems. Structuring the design of the software as concurrent tasks orthogonal to the module hiding structure provides a complete visibility of the system, giving the possibility of a scheduling control over the tasks and therefore more control of the performance of the system.

Moreover, MASCOT brings the possibility to use certain communication protocols. These protocols permit asynchronous communication, which avoids the tight interlocked timing relationships implicit in synchronous communication. Asynchronous communication also reduces the risk of deadlock and severe performance degradation at run time, which in systems such as a distributed real-time visual surveillance system may be important to avoid.

Acknowledgements

This on going work is part of an EPSRC-funded project called COHERENT (Computational Heterogeneously Timed Networks), grant number is GR/R32895. The goal of this project is the modelling, design and verification of concurrent, distributed, real time systems, with potential applications in control and data processing (http://async.org.uk/coherent/).

References

[1] ADVISOR specification documents (internal classification 2001).

[2] G. Bate, "Mascot 3: an informal introductory tutorial", *Software Engineering Journal*, **Vol 1**, pp. 95-102, (1986

[3] M. Christensen and R. Alblas, "V^2- Design issues in distributed video surveillance systems", pp. 1-86, (2000).

[4] H. Gomaa, "A software design method for real-time systems", *Communications of the ACM,* **Vol 27**, pp. 938-949, (1984).

[5] H. Gomaa, "Software design methods for the design of large-scale real-time systems", *Journal Systems Software*, **Vol 25**, pp. 127-146, (1994).

[6] I. Haritaoglu, D.Harwood, and L.Davis, "Who, when, where, what: A real time system for detecting and tracking people", In Proceedings of the Third Face and Gesture Recognition Conference, pages 222--227, (1998).

[7] Joint Iecca and Muff committee on MASCOT. "The official handbook of Mascot", version 2, Issue 2, (1983).

[8] D. Parnas, "On the criteria to be used in decomposing systems into modules", *Communications of the ACM,* **Vol 15**, pp. 1053-1058, (1972).

[9] I. Pavlidis, M. Vassilios,, " Advanced video-based surveillance systems", Kluwer Academic Publishers, Boston, USA.,Chapter 3, pp. 39-50, (2001)..

[10] M.Pellegrini, P. Tonani, "Advanced video-based surveillance systems", Kluwer Academic Publishers, Boston, USA, Chapter 1.3, pp. 27-33, (2001).

[11] A. Pozzobon, G. Sciutto, "Advanced video-based surveillance systems", Kluwer Academic Publishers, Boston, USA, Chapter 1.2, pp. 19-26, (2001).

[12] http:\\www.prismatica.com

[13] P. Remagnino, G. Jones,. N. Paragios and C. Regazzoni, "VIDEO BASED SURVEILLANCE SYSTEMS Computer Vision and Distributed Processing", Kluwer Academic Publishers, Boston, USA. Chapter 21, pp. 251-265, (2002).

[14] N..Ronetti, C.Dambra, "Multimedia video-based surveillance systems", Kluwer Academic Publishers, Boston, USA, Chapter 1.2, pp. 13-21, (2000).

[15] H. Simpson, "The MASCOT method". *Software Engineering Journal*, **Vol 1**, pp. 103-120, (1986).

[16] "The official Handbook of MASCOT", version 3.1, Issue 1, (1987).

[17] M. Valera, S. A. Velastin. "An Approach for designing a real-time intelligent distributed surveillance system", *IEE Symposium Intelligent Distributed Surveillance Systems*, (2003).