# CONTROL CODE GENERATOR USED FOR CONTROL EXPERIMENTS IN SHIP SCALE MODEL

**Polo, O. R. (1), Esteban, S. (2), Maron, A. (3), Grau, L. (4), De la Cruz, J.M. (2)**

*(1) Dept Arquitectura de Computadores y Automatica. Universidad Europea de Madrid.28670 Villaviciosa de Odon, Madrid, Spain.*
*Tf: (34) 91 6647800 (721); e-mail: opolo@darc.esi.uem.es*

*(2) Dept. Arquitectura de Computadores y Automatica, Fac. Fisicas. Universidad Complutense de Madrid. 28040 Madrid, Spain.*

*(3) CEHIPAR, Canal de Experiencias Hidrodinamicas de El Pardo El Pardo. Madrid, Spain*

*(4) Dept. Informatica y Automatica, Fac. Ciencias. UNED 28040 Madrid, Spain*

Abstract: After a study of control design to get a good candidate for testing, it comes a step of experimental confirmation. The general objective of the research is to smooth the vertical motions of a fast ferry. A T-foil and transom flaps are added to a scaled-down replica of the fast ferry. These appendages can move under control. So there is a control system installed on the replica, that moves the appendages using motors, and measures the main variables of the ship and actuators motions. This control system is based on an industrial PC with electronic interfaces for motors and sensors. The control algorithm obtained by the design, must be implemented as real-time control software, to be executed on the industrial PC. For a fast and easy translation from design to real-time application, a new software tool has been developed. This tool generates directly C++ code, easy to compile, from a graphical description of the control. With this tool, the experiments have been achieved in short time. During experiments, several non expected circumstances appear, but this was not a problem: the tool allows for an easy improvement of the original design. The paper describes the tool and its use during experiments. 

Keywords: ship control, rapid prototyping, programming environment, automatic control

## 1. INTRODUCTION

This work is part of the investigation about alleviation of fast ferry vertical accelerations using appendages. Before it, the following stages of the project have been completed: the experiments with the scale model without actuators had been done in CEHIPAR (Canal de Experiencias Hidrodinamicas de El Pardo, Madrid; in English: El Pardo Model Basin) [CEHIPAR, 2000], vertical dynamic models of the ship had been obtained from them, SIMULINK simulation environment had been tested and our work group had started to design the control algorithms.

The next logical step was to apply the control on the scaled down replica with appendages. We decided to employ an industrial PC to test the control algorithms with the scaled down replica. It was important to check the different control strategies, and to be able to do changes quickly in the CEHIPAR installations. This was motivated because the trials had to be performed in a short time without interrupts.

At present, our group is working in the development of a software tool for control code automatic generation. This kind of tools are suitable to solve the design of control software for reactive systems like the control experiments with the replica. The tool runs on MS-Windows and generates C++ code that is supported by a real time operating system called RTKernel. The results, during our experiments, have satisfied the expectations. We have proved the

efficiency of the tool to redesign quickly the control software "in situ". Moreover we always could work with a stable control program that met the specifications.

This tool can be used in other experiments related with ship control. Because of it we consider interesting its presentation. This paper begins with a introduction to ROOM methodology, after that we explain the tool developed and its utilization to solve the control experiments with the replica. Finally we present the conclusions of this work

## 2. ROOM METHODOLOGY

ROOM (Real-Time Object Oriented Modelling) was introduced by Selic in 1994 [Selic, et al., 1994 ]. It is a formalism for modelling real time systems using the object oriented paradigm. This paradigm was introduced in the beginnings of SIMULA and SMALTALK languages and it is characterized to confer behaviour to the objects of computation. In a reactive system, the object of the computation will be actors whose behaviours cooperate to solve the system.

ROOM lets us describe the structure and behaviour of the real time system using diagrams. The main components are actors that communicate between them by message passing. The behaviour of each actor is defined using a kind of state chart, called ROOMCharts, based on the Statecharts introduced by Harel [Harel, 1987]. The received messages lead the trigger of the transitions between the states.

Working with ROOM, the designer of software control system has to define actors, give them a behaviour, and trace connections between them to establish their communications. ROOM also defines the set of scheduling, communication and timing services to satisfy the requirements of the real time software systems. The communication include the definition of communication protocols between actors, the assignment of priorities to the messages and the management of message queues.

Similarly to the functional blocks hierarchy employed in SIMULINK, in ROOM one actor can be construct from other actors. This fact lets define several levels in the structure of the system. The ROOM formalism can be implemented in different ways and its adoption is very interesting to develop a clear and structured real time control code.

## 3. THE ENVIRONMENT FOR FAST CREATION OF REAL TIME CONTROL SYSTEMS

As a result of the investigation performed by our group, a CASE tool called EdROOM has been development. This tool is capable to generate automatically real time control code starting from a ROOM model of the system. It includes a graphic editor to define the actor structure of the model and the behaviour of its actors. The figure 1 shows part of the structure of a model, made with EdROOM, that includes 5 actors and their interconnections. The figure 2 shows the behaviour of one of this actors edited with EdROOM. The tool also lets define the communication protocols, and the petitions to the communication, timing and scheduling services.

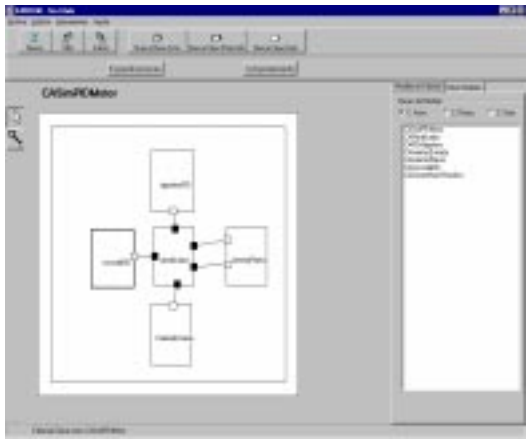

Figure 1: Actors of a ROOM model edited with EdROOM.

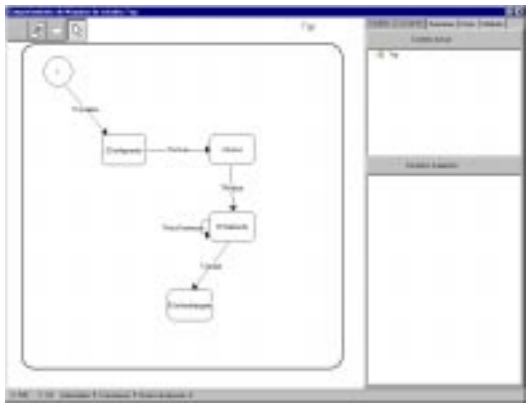

Figure 2: ROOMChart that describes the behaviour of a ROOM actor.

EdROOM runs on Ms-Windows, but it generates C++ code supported by RTKernel [3]. RTKernel is a multitasking real time kernel which uses pre-emptitive scheduling. It uses Ms-DOS to provide some services as file management but works under it and manages its own scheduler to assure the necessary

determinism for working in a real time conditions. Ms-DOS is not a real time system and it does not offer multitasking therefore we avoid it.

After we have constructed the graphic model of the system and we have defined a few functions integrated with it, we can generate C++ control code using a specific utility of EdROOM. This source code is compiled and linked with a library called mv_rtk.lib which implements the communication, timing and scheduling services in RTKernel. The result is a compact executable file which can run on a any PC with Ms-DOS. The implementation of mv_rtk.lib in other real time operating systems lets us construct portable and highly reusable code using EdROOM.

## 4. REPLICA CONTROL HARDWARE

The Replica constructed by CEHIPAR is 4.5 meters long. It has incorporated a T-foil near the bow and two transom flaps. There is a step motor to move the T-foil wings and other to move the Flaps. We have used an industrial PC to control them. Figure 3 shows a picture of the replica with its appendages. Figure 4 shows a picture of one of the step motors.

Figure 3: A view of the replica wit the T-foil and the flaps.

Figure 4: A view of the step motor.

The step motors have a 0.18º precision per pulse and both included encoders to measure their position. The TE5312 card is connected to the PC bus to read the encoders value.

In accordance with the limits of the real ship with respect to the wings rotation speed provided by the hydraulic cylinders, the maximum rotation speed of the motors is 67.5 º/ seg. This value corresponds with a 13.5 º/ seg speed in the real ship.

The sensors located in the replica measure the following variables: heave, pitch, the height of the arriving wave, the drag forces ( starboard and port ) and the accelerations in several points of the replica.

The replica is moved by a carriage which has a complete installation of data acquisition devices and video cameras. The control is performed by the industrial PC fixed to the carriage and located near the replica. The PC includes the Advantech PCL812PG data acquisition card connected to its bus. Six of its sixteen analog inputs channels are used by the control program to perform the periodic sampling and four of its sixteen digital outputs manage the control of the motors. Depends on the control algorithm it is necessary to sample a different set of signals. For example the first control studies have been done using the WVA acceleration.

The industrial PC has also a console in which we can observe the values of the main variables. This has been very useful in the calibration tasks and to complete the basic control tests (moving by hand the replica and checking if the value of the motor rotation signs truly counteract the wave effects).

## 5. EDROOM APPLIED TO THE CONTROL OF THE REPLICA

The model of the replica control includes 5 actors connected between them. The figure 5 shows this aspect of the model structure.

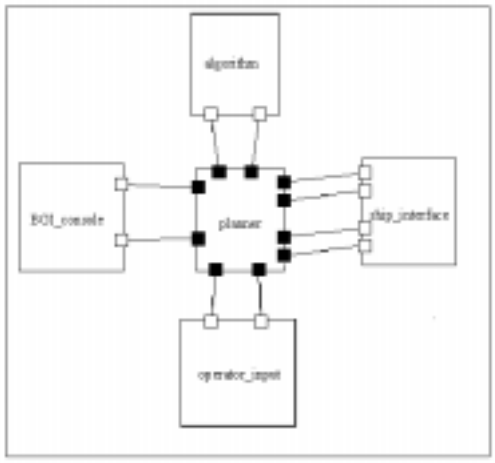


Figure 5: The five main actors of the ROOM model of control program.

The function of these actors are the followings:

-"ship_interface" is employed to provide the interface with the step motors and the sensors.

-"BGI_console" displays the main variables in the console.

-"operator_input" is employed to catch the operator commands given to the system using the keyboard.

-"algorithm" executes the control algorithm.

-"planner" takes charge of coordinate the work of the rest of the actors in each sampling period. It manages the start-up and termination of the whole system too.

The "planner" has three main states in its behaviour which implements the start-up, control execution and termination. Figure 2 shows the top state chart of the "planner" actor and the dynamics of the three behaviour states are showed in the figures 6, 7 and 8.

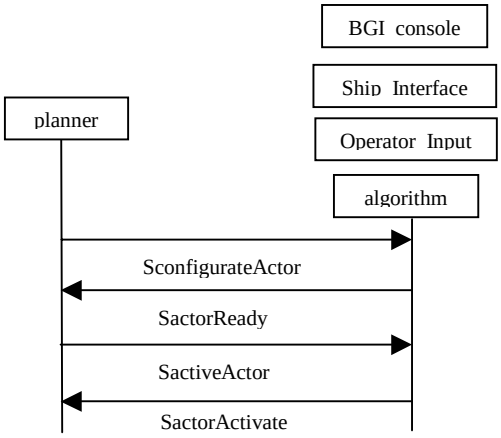


Figure 6: Sequence of messages of the start-up

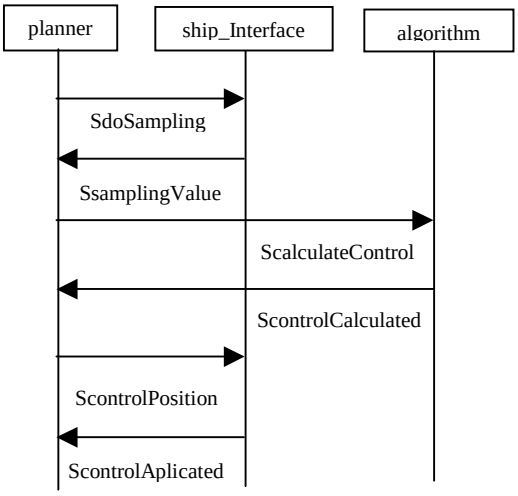


Figure 7: Sequence of messages of the control execution

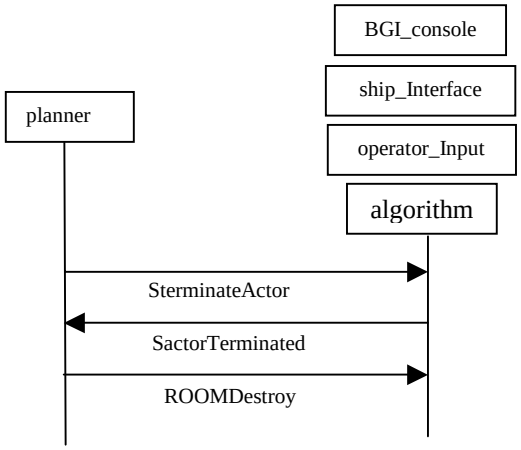


Figure 8: Sequence of messages of the termination

The "planner" also contains other actor, called "periodic_Sampling", which manages the sampling timing. The "ship_Interface" has three actors inside as it is showed in figure 9.

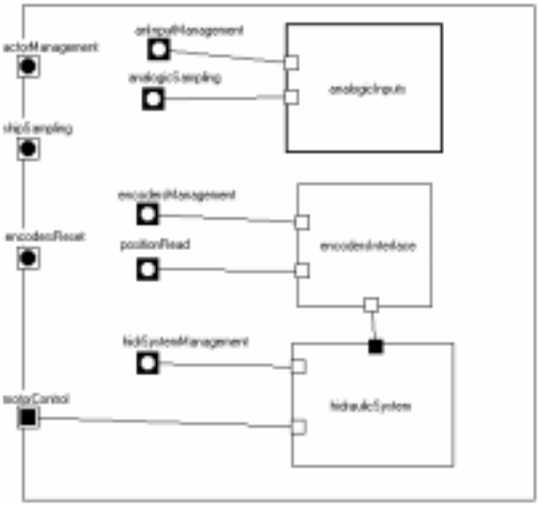

Figure 9. Internal structure of the "ship_Interface" actor

One of them performs analog measurements sampling with the PCL812PG card. Other provides access to the TE5312 card and lets read the motor encoders value. The third is devoted to move the appendages in accordance with the calculated commands. This includes also two actors more in its internal structure. One of them generates the pulses to move the T-foil and the other moves the flaps.

To accomplish the primary tests of the control system, the "ship_Interface" actor was modified to take the values of the analog inputs from a file instead from the replica. The file contained data of a previous experiment. The modification of the “shipInterface” actor was simply made by a substitution of the actor “analogInputs” with

another actor “fileAnalogInputs”. This enabled us to do trials of the program in our laboratory, using two motors similar to those mounted in the replica, before to go to the CEHIPAR.

## 6. "IN SITU" REDESIGN EXPERIENCES

One of the advantages that we wanted to obtain with the use of EdROOM was to make easy the redesign of the control program. The experience shows that the design of this kind of system consists of an iterative process of test and error that it is convenient to be clear and agile. The experiments with the replica have been a good occasion to check the efficiency and suitable characteristics of this tool.

For example, after being ready the automatic control of the replica based in the algorithm, it was also considered opportune adding a manual control to perform calibration tasks and to do initial tests before the experiments. This redesign modified some of the actor's behaviour but it was ready in less than a day. The new model continued to be clear and the program generated needed little debugging.

Besides, during the experiments with waves, we realized that the motors did a considerable noise which affected to the sensors measurements. Then, it was considered suitable to use a software filter to decrease this noise. This decision modified only the actor "periodic_Sampling" dedicated to perform the samples with a period of 2 ms and, after filtering, send them to the "planner" each 10 ms as happened before. Again, redesign could be ready in less than a day and the improving obtained, with respect to the control capabilities, has been notable.

That fact is also the key point of the reusability of the ROOM actors and it let us say that actors work like “plug and play” software components. As the result of all these aspects the development of real-time systems can be affronted like the evolution of a basic prototype that runs properly in each stage.

The code size of the control program is less than 400 kbytes ( including the graphic display of the variables) and the PC used has been a Pentium at 200 MHz. As a whole we have obtained a low cost control system which meet the specifications. The figure 10 shows the console of the control program generated with EdROOM. The graph on top represents the values of the accelerations and the graph below represents the positions of T-foil and flaps.

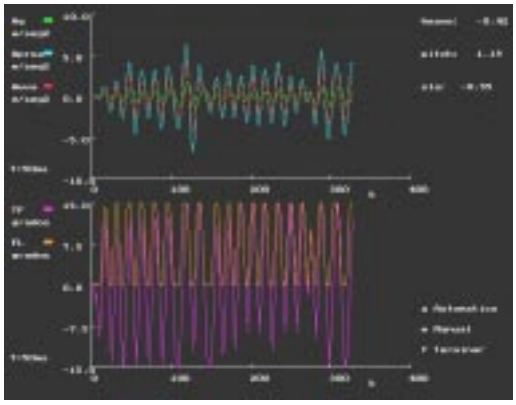

Figure 10: Graphic display of the variables on the generated control program.

## 7. CONCLUSIONS

A visual CASE tool for real time automatic control code generation has been created. The tool, called EdROOM, runs under Ms-Windows. It follows an object oriented methodology of software engineering, called ROOM, based in the use of actors. The use of ROOM guarantees a structured and clear design of control programs. The executable code is generated quickly and can be used in Intel PC. This code uses a low cost real time kernel called RTKernel which works in a stable way together with Ms-DOS. The library mv_rtk.lib implements the communication, timing and scheduling services using RTKernel primitives necessary to execute the ROOM model. This library is linked with the C++ code generated to create the executable. The figure 10 shows the relationships between C`++ code generated, mv_rtk.lib library, Ms-DOS and RTKernel.

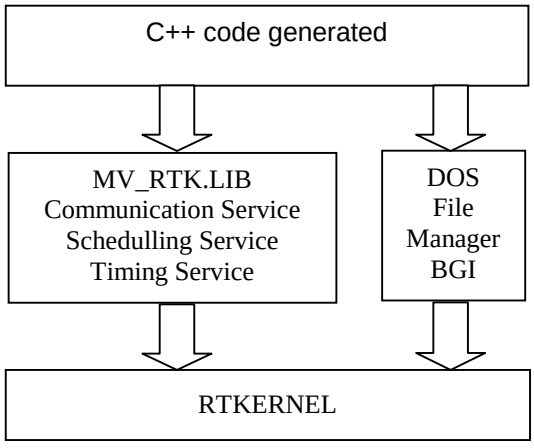


Figure 11: Relation between C`++ code generated, mv_rtk.lib library, Ms-DOS and RTKernel.

EdROOM has been employed to perform the control of a fast ferry replica within the experiments in the CEHIPAR. The control moves a T-foil near the bow and two transom flaps to alleviate, in the best possible way, the vertical accelerations of the ship. Thanks to EdROOM the experiments have been done quickly and the debug time has been reduced considerably. Besides, it has been able to test most of the control program in laboratory conditions, avoiding unnecessary experiments in CEHIPAR.

One of the advantages of using EdROOM have been its capability to make easy the redesign of the control program. During the experiments there have been several situations in that this redesign have been necessary. It has been proved EdROOM can solve efficiently this sort of problems.

In comparison with other alternatives, like the RealTime Workshop (MATLAB), our approach is more in the spirit of software engineering methods for a modular, clear development of potentially big applications. Besides EdROOM can be used for several RTOS platforms, with no changes in the design.

The generality of the EdROOM approach let us tackle easily other experimental problems like the test with other replicas and several control strategies. For example, it is foreseen one set of experiments in multivariable control based on heave and pitch values.

It is also possible to develop other versions of the mv_rtk.lib library, which supports the code generated by EdROOM, to allow for the generation of code for other hardware targets with other operating systems. The only requirement is the existence of a C++ compiler.

## 8. ACKNOWLEDGEMENTS

The authors want to thank the support of the CICYT Spanish Committee (project TAP97-0607-C03-01), and the collaboration of the BAZAN and CEHIPAR staff.

## 9. REFERENCES

CEHIPAR (2000): www.cehipar.es

Harel, David. (July 1987), Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming 8 :231-274.

RTKernel 4.0 and RTKernel 4.5 Real-Time Multitasking kernel for C/C++. User's Manual.

Selic, Bran, Gulleckson, Garth, and Ward, Paul T. (1994), Real-Time Object Oriented Modelling. NewYork, John Wiley and Sons.