A Student Project in the Field of Satellite Tracking

M. Barbeau
Département de mathématiques et d'informatique
Université de Sherbrooke
Sherbrooke, Québec, CANADA J1K 2R1
www.dmi.usherb.ca/~barbeau

Abstract

This paper presents a project given to the students of a real-time system course. The project consists of developing a software for tracking in real-time non geostationary satellites. Students have to analyze, design, program, and test their solution. The project is not solely a simulation. There is a satellite tracking lab where students can really experiment and validate their solution. This paper presents the problem given to the students, the hardware platform, and some elements of a typical solution.

1. Introduction

This paper presents a project given to the students of a course entitled Real-Time Systems that has been offered for the first time at the fall 1996 semester at University of Sherbrooke. The course is part of the curriculum of a Master of Software Engineering. This course aims to render the students apt to the realization of real-time systems by using methods of development and tools specific to this type of systems. More specific objectives are to make the students capable of understanding characteristics of real-time systems and to apply a method for real-time system development to a concrete problem. The course lasts one semester and consists of lectures, both given by the professor and students, and a major project. This paper is about the project given to the students.

There are several areas where systems have real-time characteristics. For this course, we chose the exciting one of satellite telecommunications. In their project, students develop a case study in the area of tracking in real-time non geostationary satellites. They perform a full software development project: analysis, design, implementation, and test. Each student develops its own solution.

The suggested development framework is the object-oriented development methodology of Booch [2]. The reason we use Booch is because it explains very well how to tackle the different development phases (in particular analysis, design, and implementation). The expected results of each phase are also very well described. One the flaws of the Booch approach, however, is the non executability of the analysis and design models. Verification through simulation, before implementation, is therefore not possible. Hence, although the Booch methodology is employed, the modeling language used for the project is ROOM [5]. The modeling language of ROOM has been devised specifically for real-time systems. It is very well documented and supported by a tool called ObjecTime. ObjecTime allows editing of models as well as verification through simulation. Automatic code generation is also possible, although not exploited in the course. Rather, students translate their ROOM model into C and C++ and program the implementation for the QNX real-time operating system [4]. The rest of the paper is organized as follow. Section 2 describes the problem given to the students. The hardware platform for the implementation of their solution is described in Section 3. Some elements of a typical solution are given in Section 4. We conclude with Section 5.

2. The Problem

The problem given to the students is in the area of telecommunications satellite tracking. This topic is very actual, relevant, and motivating. Telecommunications satellites can be classified into two categories (see Figure 1) according to whether or not they place the spacecraft in a geostationary orbit. A geostationary orbit is characterized by the fact that the equatorial and orbital planes coincide. Moreover, the satellite describes a circle around the earth at an altitude of approximately 35 680 km. These conditions are such that the satellite performs a complete cycle in every day, as well as the earth. Thus from the point of view of an observer on the ground, at a given location, the satellite is always at the same position in the sky. Tracking of geostationary satellites is trivial.

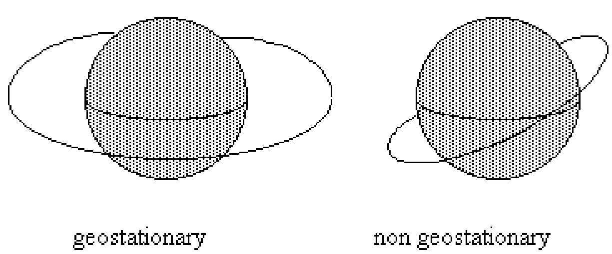

Figure 1: Types of orbits

The satellites in which we are interested in the student project are rather of the non geostationary type. The orbital and equatorial planes do not coincide and the first is inclined with respect to the second. The National Oceanographic and Atmospheric Administration (NOAA) [www.noaa.gov] maintains series of weather satellites on non geostationary orbits. Their altitude is relatively low (e.g.,: 850 km) and weather images can be received with relatively simple equipment [6]. In addition, AMSAT [www.amsat.org] (and other related voluntary organizations) maintains a series of non geostationary satellites, called Orbiting Satellite Carrying Amateur Radio (OSCAR), which can be used as relays in different modes of transmission (analog or digital) [3]. Both reception and transmission are possible through them (an amateur radio license is required for transmission [www.rac.ca, www.arrl.org]).

Figure 2 illustrates the coverage of a typical non geostationary weather satellite. The area in which the satellite is visible from the earth is delimited by the ovoid figure. The latter is called the acquisition circle. Most of these satellites are at a relatively low altitude. It means that the acquisition circles of satellites of this type are of small dimension. The small cross indicates the place on the ground directly under the satellite. This point is called the subsatellite point (ssp). Because of the fact that the earth rotates on itself (and other phenomena), given a latitude and direction in which the satellite moves (up or down), the longitude of the ssp is shifted to the west from one cycle to another. It allows the satellite to cover all the surface of the earth during a certain period of time (e.g., a day). In Figure 2, the curved line indicates the future trace of the ssp for a cycle beginning from the current ssp.

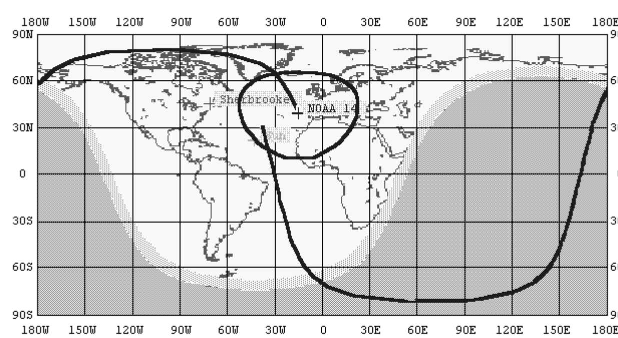

Figure 2: Orbit of a non geostationary satellite

To be able to communicate with a non geostationary satellite, it is necessary to determine at what time our position on the ground will be part of an acquisition circle. Directional antennas equipped with motors to adjust in real time their orientation may be required for communicating because the distance may be to great or the signal may be to weak. If directional antennas are necessary, we have to determine in real time in which direction they have to be oriented in terms of azimuth (an angle between zero and 360 degrees) and elevation (an angle between zero and 180 degrees). To make these calculations, a mathematical model of satellite orbits has to be implemented. A two-body mathematical model is well documented in Refs. [3] and [1]. Besides, the mathematical model has to be fed with data, called orbital elements. The mathematical model predicts the position of a satellite at a given time from coordinates observed at a time of reference. In order for this prediction to be the most accurate as possible, orbital elements have to be the most recent as possible. One can obtain them thanks to the Internet on the Web site of AMSAT (www.amsat.org) or the FTP site of Air Force Institute of Technology (archive.afit.af.thousand/pub/space/tle.new).

The problem given to students is the development of software for tracking in real time non geostationary satellites. Coding techniques and frequencies of NOAA and OSCAR spacecrafts are of the public domain. They can be worked out with relatively modest radio equipment. They are therefore really great for experimenting satellite communications. Students are challenged to experiment and validate their satellite tracking software with NOAA and OSCAR satellites. Development of a real-time satellite tracking software makes a great course project because it requires quite deep analysis and design. The size of the program, in terms of lines of code, is relatively modest, and the problem has soft real-time as well as hard real-time facets. In addition, validation of the result is possible with a not costly to set up satellite tracking laboratory described in the next section.

3. Hardware Platform

This section describes the hardware platform on which students develop and test their satellite tracking software. The communication equipment must be capable of communicating in the VHF (2 m) and UHF (70 cm) radio bands.

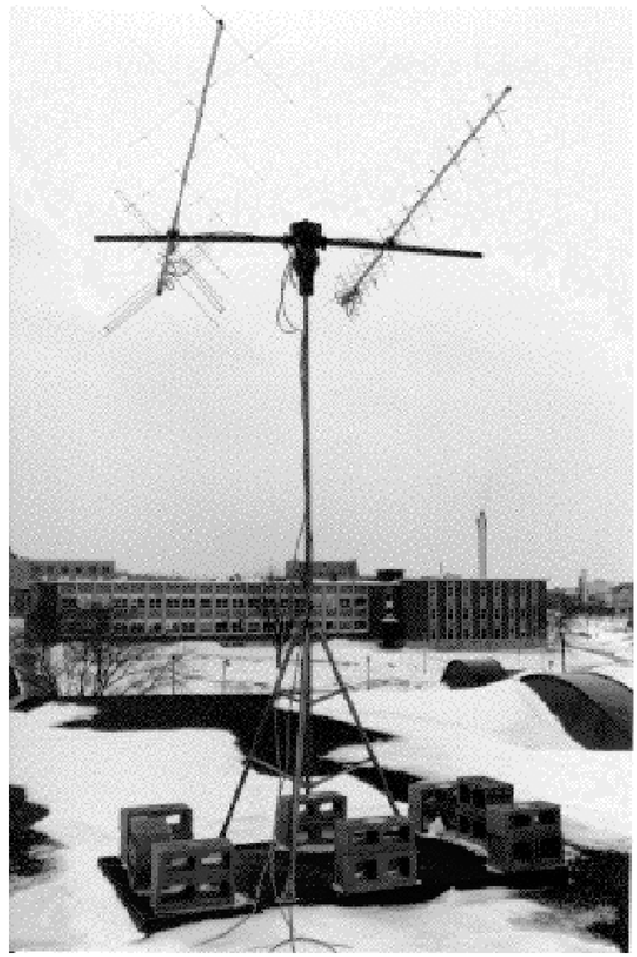

Figure 3: Crossed yagi antennas

Directional antennas equipped with rotors, to adjust in real time their orientation, are required to communicate. An antenna for the 2 m band and another for the 70 cm band are necessary. The polarity of antennas of satellite is circular. Antennas on the ground can be of the type yagi (similar to broadcast television reception antennas) but those of type crossed yagi give best results. A crossed yagi possesses both a series of vertical elements and a series horizontal elements. Their gain is three decibels higher than a non crossed yagi because their polarity is equally circular. Antennas are mounted side by side on a horizontal cross boom. Figure 3 presents a system of antennas for the 2 m and 70 cm bands. It comprises crossed yagis for the 2 m and 70 cm bands. Their length is approximately 10 feet. Rotors are necessary to vary the azimuth and elevation of antennas in order to maintain their orientation in the direction of a satellite. The rotor system, pictured in Figure 3, is computer controlled. Antennas are mounted on a tripod installed on a flat roof of a building. Their height is not critical. It is, however, necessary that the path of transmission-reception be the most clear as possible.

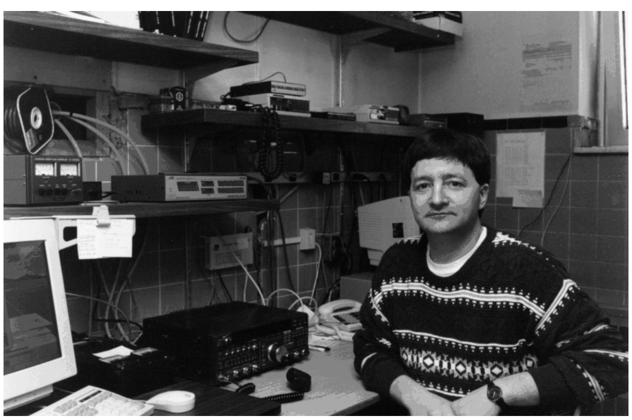

Figure 4: Radio equipment

Preamplifiers make possible communication with antennas of shorter length. They have to be as close as possible to the antennas. Ideally on the mast in a case that is going to protect them from moisture. It is, however, necessary to keep in mind the fact that a preamplifier increases not only the amplitude of the signal but equally the noise. Its efficiency can be reduced considerably in urban areas. Thus, when these antennas are installed in an electromagnecticly noisy environment, they have to be as long as possible (e.g., 15 feet). In contrast, for an installation in an environment where the level of the noise is moderate, antennas can be shorter (e.g., five to eight feet) and the signal can be strengthened by means preamplifiers. There exist radios specially devised for satellite communication in the 2 m and 70 cm bands. The most popular models are the ICOM IC-821, Kenwood TS-790, and Yaesu FT-736. These radios have several features that facilitate the task of the operator. Figure 4 presents the equipment of our satellite communication station. There is a Yaesu FT-736R radio covering the 2 m, 70 cm, and 23 cm bands. The station comprises also an AEA DSP-2232 digital signal processor to encode and decode the different signals employed by the satellites. Excluding the computer, we invested about 4 800$ US for the equipment (radio 2 000$ US, DSP 1 000$ US, antenna system 1 300$ US, and various harware pieces 500$ US).

4. Elements of a Solution

In this section, we discuss the highlights of a typical solution to the problem given to the students. A good design consists of a top level object with three sub objects (actors, in the language of ROOM). The top level object models the control flow of the application. The three sub objects are responsible of determining the ssp, calculating the azimuth and elevation of the satellite, and driving the antennas in the right direction. They are respectively called the groundTracker, azElProvider, and driver. The behavior of the top level actor is modeled by a ROOMchart (an extended state-transition diagram) triggered on a periodic basis by the expiration of a timer. It is initialized with fresh orbital elements of a satellite. When its behavior is triggered, it polls the groundTracker to obtain the current ssp and altitude of the satellite. Both values are passed to the azElProvider which returns the current azimuth and elevation of the satellite. Finally, the azimuth and elevation are transmitted to the driver that starts or stops the motors of the antennas in accordance to the difference between the current position and the target.

Separation of control, in the top level object, and functional aspects, in the sub objects, results in low coupling between the functional elements. Therefore, its makes them more reusable, in other contexts, and substitutable, by other functionally equivalent elements.

Design of the object groundTracker requires a deep understanding of a mathematical model of orbits. It is one of the difficult parts of the project. In the course, we spent two three hours lectures on that topic. There are several models but a simple two-body model (a model that takes into account solely the influence of the earth on one orbiting spacecraft) is popularized in Ref. [3]. More details can also be found in Ref. [1]. There are two levels of difficulties: designing a software that implements a mathematical model of circular orbits and designing one that implements a model of elliptical orbits. The former is relatively simpler than the latter. Both, however, require understating of tracking terms (e.g., inclination), spherical trigonometry, and astronomy concepts (e.g., sideral time). Implementation of an elliptical model is more challenging because the equations are more complex and computation of one of the intermediate values requires implementation of the Newton-Raphson method. In both cases, however, there is a challenge in the understanding of the mathematical model. The resulting program is quite small: a couple of pages. All students implemented a model limited to circular orbits.

Design of the azElProvider is more straightforward. This object determines where to point the antennas according the location of the ground station, ssp, and altitude of the satellite. Equations required for this part of the problem are documented in Ref. [3].

Design of the driver needs understanding of the hardware interface with antenna rotors. There are a couple of interface models available on the market. One of the difficulties we encountered in the project was to find a product with the companion documentation required for understanding programming of the card. The software has soft as well as hard real-time aspects. On the one hand, the top level object is triggered on periodic basis (e.g., every couple of seconds). Although, delays in providing the azimuth and elevation are not so critical since the antennas have a certain beam width. On the other hand, rotors of antennas strike a knock right after 360 degrees of azimuth or 180 degrees of elevation. When these positions are reached the motors do not stop by themselves and they may burn after a while. So care must be taken in the design of the driver to handle this situation. To diminish the risk of burning the motors, we developed an electronic box that simulates the rotors. For the initial test, we plug into the rotor interface card this box instead of the real rotors. This kind of hardware simulator is mandatory when such equipment is used in turn by several students. The software developed by students has to run on the QNX platform. QNX has several real-time features like fast context switching of processes and fast interprocess communication. There was, however, a lack of use by the students of features specific to QNX, like multiprocess implementation. Students developed their program on Sun work stations, with which they are already very familiar, then transfer it on QNX for the final testing.

5. Conclusion

This paper has presented a student project for a real-time system course. The problem consists of developing a software for tracking in real time non geostationary satellites. The development of a solution is a quite substantial piece of work and keeps students busy for a whole semester. At the beginning of the course, we assume that they already know C and C++ as well as a little about software engineering. In addition, they have to read about the Booch method, understand the ROOM model, learn the ObjecTime tool, acquire a deep understanding of a two-body satellite orbit model, understand the hardware, and finally they need to become familiar with QNX. We believe that acquisition of knowledge about so many things is typical of a real-time system development project. Not all students succeed to do the whole thing but most of them are highly motivated.

More information about ROOM and ObjecTime is available at the following URL www.objectime.on.ca. More information about QNX is available at the URL www.qnx.ca. AMSAT maintains a Web site containing orbital elements, technical descriptions of OSCAR satellites, and examples of tracking software (www.amsat.org). Information about states of OSCAR satellites, and other related spacecrafts, are available and updated on a weekly basis at the Web site www.cs.indiana.edu (look for file finger/pilot.njin.net/magliaco/w).

Acknowledgments

The author would like to thank Marc Normandeau and Gaston Sirois for their participation to the setting up of the satellite communication station of the University of Sherbrooke. Thanks also to Jean Bédard, who has taken the pictures, and Aubin Roy, that has developed the electronic simulator of the antenna rotors. Part of the ideas that are presented in this paper emerged from a research project, joint with Centre de recherche informatique de Montréal (CRIM), financially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and Bombardier Inc.

References

[1] R. Bate, D. Mueller, and J. White, *Fundamentals of Astrodynamics*, Dover Publications Inc., New York, 1971.

[2] G. Booch, *Object-Oriented Analysis and Design with Applications*, Second Edition, Benjamin/Cummings, 1994.

[3] M. Davidoff, *The Satellite Experimenters Handbook*, The American Radio Relay League, 2nd Edition, 1994 (ISBN: 0- 87259-318-5).

[4] QNX Software Systems Ltd, *QNX 4 Operating System - System Architecture*, 1993.

[5] B. Selic, G. Gullekson, and P. T. Ward, *Real-time Object-oriented Modeling*, John Wiley & Sons Inc. 1994 (ISBN: 0-471-59917-4).

[6] R. E. Taggart, *Weather Satellite Handbook, The American Radio Relay League*, 5th Edition, 1994 (ISBN: 0-87259-448-3).