



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2017/07.04.19.59-TDI

AN AUTONOMOUS AND DYNAMICAL APPROACH TO SMALL SATELLITE GROUND STATIONS NETWORKS

Jorge Enrique Espindola Diaz

Doctorate Thesis of the Graduate
Course in Space Technology
and Engineering, guided by
Drs. Maurício Gonçalves Vieira
Ferreira, and Walter Abrahão dos
Santos, approved in July 18, 2017.

URL of the original document:

<<http://urlib.net/8JMKD3MGP3W34P/3P83BUE>>

INPE
São José dos Campos
2017

PUBLISHED BY:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

E-mail: pubtc@inpe.br

**COMMISSION OF BOARD OF PUBLISHING AND PRESERVATION
OF INPE INTELLECTUAL PRODUCTION (DE/DIR-544):****Chairperson:**

Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação (CPG)

Members:

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Dr. André de Castro Milone - Coordenação de Ciências Espaciais e Atmosféricas (CEA)

Dra. Carina de Barros Melo - Coordenação de Laboratórios Associados (CTE)

Dr. Evandro Marconi Rocco - Coordenação de Engenharia e Tecnologia Espacial (ETE)

Dr. Hermann Johann Heinrich Kux - Coordenação de Observação da Terra (OBT)

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SID) **DIGITAL LIBRARY:**

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

DOCUMENT REVIEW:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

ELECTRONIC EDITING:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2017/07.04.19.59-TDI

AN AUTONOMOUS AND DYNAMICAL APPROACH TO SMALL SATELLITE GROUND STATIONS NETWORKS

Jorge Enrique Espindola Diaz

Doctorate Thesis of the Graduate
Course in Space Technology
and Engineering, guided by
Drs. Maurício Gonçalves Vieira
Ferreira, and Walter Abrahão dos
Santos, approved in July 18, 2017.

URL of the original document:

<<http://urlib.net/8JMKD3MGP3W34P/3P83BUE>>

INPE
São José dos Campos
2017

Cataloging in Publication Data

Espindola Diaz, Jorge Enrique.
Es65a An autonomous and dynamical approach to small satellite
ground stations networks / Jorge Enrique Espindola Diaz. – São
José dos Campos : INPE, 2017.
xxvi + 184 p. ; (sid.inpe.br/mtc-m21b/2017/07.04.19.59-TDI)

Thesis (Doctorate in Space Technology and Engineering) –
Instituto Nacional de Pesquisas Espaciais, São José dos Campos,
2017.

Guiding : Drs. Maurício Gonçalves Vieira Ferreira, and Walter
Abrahão dos Santos.

1. ADSGS. 2. Artificial intelligence. 3. Ground station
networks. 4. Hungarian algorithm. 5. Software defined radio.
I.Title.

CDU 004.8:629.783



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](#).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#).

Aluno (a): **Jorge Enrique Espindola Diaz**

Título: "AN AUTONOMOUS AND DYNAMICAL APPROACH TO SMALL SATELLITE
GROUND STATIONS NETWORKS".

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Doutor(a)** em
**Engenharia e Tecnologia Espaciais/Eng.
Gerenc. de Sistemas Espaciais**

Dra. **Maria de Fátima Mattiello-Francisco**



Presidente / INPE / São José dos Campos - SP

Dr. **Maurício Gonçalves Vieira Ferreira**




Orientador(a) / INPE / SJCampos - SP

Dr. **Walter Abrahão dos Santos**



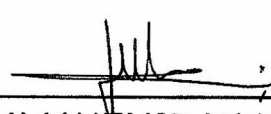
Orientador(a) / INPE / São José dos Campos - SP

Dr. **Fabricio de Novaes Kucinskis**



Membro da Banca / INPE / São José dos Campos - SP

Dr. **Douglas Soares dos Santos**



Convidado(a) / ITA / São José dos Campos - SP

Dr. **Daniel Portnoy**



Convidado(a) / IAI / Israel - ISR

Este trabalho foi aprovado por:

() maioria simples

☒ unanimidade

São José dos Campos, 18 de julho de 2017

“If you have a problem that can be fixed, then there is no use in worrying. If you
have a problem that cannot be fixed, then there is no use in worrying”
Buddhist proverb

“To write a bit you have to read a lot”
Jorge Espíndola

To my children David Santiago, Vivian Andrea, Jorge Ivan and my wife Maria Patricia, as well as Juan Arquímedes and Maria del Carmen† my parents.

ACKNOWLEDGMENTS

First of all to God, for the gift of life and health in the development of this work.

My wife Maria Patricia, for strength, dedication, understanding and love. To my children David Santiago, Vivian Andrea and Jorge Ivan daily engine for the realization of this dream.

To my parents, for all the support, love, education and lifelong teaching, being examples of real parents. Like my brothers and relatives including my mother-in-law, who were in Colombia and were support in difficult times.

To my supervisors, Walter Abrahão dos Santos and Mauricio Vieira Ferreira, for their trust and dedication to the work.

To my great friends and colleagues Roberta de Cassia Porto, Jenny Amparo Rosales, Julian Molano Peralta, Jaime Enrique Orduy, Josue Nicolas Pinzon Villamil, Gabriel Coronel and Italo Pinto Rodriguez, for his company and the long years of friendship, as well as for the support in this work.

To the professors and friends Daniel Portnoy, Daniel Choukroun and Edson Pereira for the contribution with their knowledge to this work.

To educational institutions such as the Pedagogical and Technological University of Colombia - UPTC in Colombia, the National Institute for Space Research - INPE in Brazil, Ben Gurion University - BGU in Israel and Coordination for the Improvement of Higher Education Personnel - CAPES, For the academic and financial support they gave me to finish my doctorate.

ABSTRACT

The costs in satellite operations are generally not negligible, especially for long-term missions. An alternative to lower costs is to increase the level of automation in procedures whenever possible. This thesis proposes a dynamic and autonomous approach to small satellite ground station networks aimed at minimizing these operational costs. The proposed solution is called the ADSGS (Autonomous and Dynamic System Ground Station) and is a middleware with hardware and software components to operate in a distributed network environment of ground stations. In this work the SATNet network was considered as currently it lacks an allocation component with the characteristics of dynamism and autonomy. In ADSGS this is provided through the use of artificial intelligence in a rule-based expert system. In the proposal, an ADSGS network agent operates autonomously and dynamically on the SATNet network where the associated station components are managed. The ADSGS agent uses an adapted version of the Hungarian Algorithm for combinatorial optimization of allocation problems and responds to events such as service interruption. The proposed hardware component uses COTS and Software Defined Radio (SDR) technology while the software component uses packages such as Orbitron, ProEst, the SINTA expert system, among others. UML modeling is provided to document the software component. A case study is made to illustrate the main features of the ADSGS consisting of a small simulation in MATLAB with STK (Systems Tool Kit) two scenarios of designation (**1-to-m** and **n-to-m**) of stations in the SATNet network to satellites by the ADSGS agent.

Keywords: Adapted Hungarian Algorithm. ADSGS. Ground Station Networks. Software Defined Radio.

UMA ABORDAGEM DINÂMICA E AUTÓNOMA DE REDES DE ESTAÇÕES TERRENAS PARA PEQUENOS SATÉLITES

RESUMO

Os custos nas operações de satélites geralmente não são desprezíveis, especialmente para missões de longa duração. Uma alternativa para baratear custos é aumentar o nível de automação em procedimentos quando for possível. Este trabalho de tese propõe uma abordagem dinâmica e autônoma para operações espaciais em redes de estações terrenas para pequenos satélites que visa minimizar seus custos operacionais. A solução proposta denomina-se ADSGS (*Autonomous and Dynamic System Ground Station* em Inglês) e é um “middleware” com componentes de hardware e software para atuação em um ambiente distribuído em uma rede de estações terrenas. Neste trabalho foi adotada a rede SATNet que carece de um componente de alocação com as características de dinamismo e autonomia. No ADSGS isto é oferecido mediante o uso da inteligência artificial em um sistema especialista baseado em regras. Na proposta, um agente de rede da ADSGS atua autônoma e dinamicamente na rede SATNet onde se gerenciam componentes das estações associadas. O agente ADSGS utiliza uma versão estendida do Algoritmo Húngaro para otimização combinatória de problemas de alocação e responder a eventos como interrupção de serviços. O componente de hardware proposto utiliza elementos de hardware COTS e tecnologia SDR (*Software Defined Radio*) ao passo que o componente de software utiliza pacotes como Orbitron, ProEst, o sistema especialista SINTA, entre outros. A modelagem UML é oferecida para documentar o componente de software. Um estudo de caso é feito para ilustrar as principais funcionalidades do ADSGS consistindo de uma pequena simulação em MATLAB com STK (*Systems Tool Kit*) e dois cenários de designação (*1-to-m* e *n-to-m*) de estações na rede SATNet a satélites pelo agente ADSGS.

Palavras-chave: ADSGS. Algoritmo Húngaro Adaptado. Redes de Estação Terrenas. Radio Definido por Software.

LIST OF FIGURES

| | <u>Page</u> |
|--|-------------|
| Figure 2.1 - Modernization Interfaces undergone on the AFSCN network | 9 |
| Figure 2.2 - Block diagram of the Mercury system | 10 |
| Figure 2.3 - Architecture of the system..... | 12 |
| Figure 2.4 - GMS Client access..... | 15 |
| Figure 2.5 - Implementation of the GMS drivers..... | 15 |
| Figure 2.6 - The MC3 rack and its contents | 16 |
| Figure 2.7 - CGA characteristics and capacities..... | 17 |
| Figure 2.8 - The MC3 architecture..... | 18 |
| Figure 2.9 - Architecture of the GENSO network..... | 21 |
| Figure 2.10 - Design of GENSO ground station type VHF/UHF | 22 |
| Figure 2.11 - Hardware design | 23 |
| Figure 2.12 - Functional diagram of the GSS application | 24 |
| Figure 2.13 - Direct Exchange of data in a GENSO network..... | 25 |
| Figure 2.14 - Scheme of uplink sequence | 26 |
| Figure 2.15 - LEO coverage projected for the GENSO network | 26 |
| Figure 2.16 - Global vision of the SatNOGS concept | 29 |
| Figure 2.17- Scheduling diagram of the observation of the network..... | 30 |
| Figure 2.18 - Components and interactions of the Client software | 31 |
| Figure 2.19 - SATNet - Network basic concept | 32 |
| Figure 3.1 - The ADSGS middleware in typical ground station architecture | 39 |
| Figure 3.2 - The ADSGS Ground Station Architecture | 40 |
| Figure 3.3 - ADSGS-adapted Hardware components | 42 |
| Figure 3.4 - The SDR Concept | 47 |
| Figure 3.5 - Simple steps to integrate of nodes into the SATNet network. | 50 |
| Figure 3.6 - Current display of the INPE node in the SATNet network..... | 51 |
| Figure 3.7 - Orbitron - Interface | 52 |
| Figure 3.8 - ProEst Software | 55 |
| Figure 3.9 - SoundModem..... | 56 |

| | |
|--|-----|
| Figure 4.1 - ADSGS Agent | 62 |
| Figure 4.2 - ADSGS - Actors Diagram..... | 65 |
| Figure 4.3 - Expert SINTA - Shell window | 66 |
| Figure 4.4 - Expert SINTA - Production Rule scheme | 67 |
| Figure 4.5 - The “4+1” view model..... | 76 |
| Figure 4.6 - ADSGS - Flow Diagram | 77 |
| Figure 5.1 - Scenario #1 - 1 Satellite to n Ground Stations | 84 |
| Figure 5.2 - Scenario #2 - n Satellite to m Ground Stations | 87 |
| Figure 5.3 - Illustrative simulation scenario using STK..... | 90 |
| Figure 5.4 - Satellite access conflict #1 | 92 |
| Figure 5.5 - Satellite access conflict #4 and #5 | 93 |
| Figure A.1 - VHF-UHF CrossBoom antenna | 113 |
| Figure A.2 - S-band antenna | 114 |
| Figure A.3 - VHF/UHF Preamplifier | 115 |
| Figure A.4 - S-band 2,4 GHz converter | 116 |
| Figure A.5 - IC-7100 Transceiver | 116 |
| Figure A.6 - Yaesu G-5500 Rotor..... | 117 |
| Figure A.7 - GS-232A Control Interface | 118 |
| Figure A.8 - S-band BIG-RAS/HR rotor..... | 119 |
| Figure A.9 - MD-01 Rotor Controller..... | 119 |
| Figure A.10 - CX-600N relay | 120 |
| Figure A.11 - AirSpy SDR..... | 121 |
| Figure B.1 - SDR - Hardware Board..... | 123 |
| Figure C.1 - A classical problem solved by the Hungarian Algorithm | 129 |
| Figure D.1 - Artificial Neuron | 139 |
| Figure D.2 - Activation Functions | 140 |
| Figure D.3 - Rule-Based System..... | 146 |
| Figure E.1 - Expert SINTA Simplified Architecture | 152 |
| Figure E.2 - Tool bar for base-making..... | 153 |
| Figure E.3 - KIB window | 154 |
| Figure E.4 - Tools bar of Expert SINTA..... | 154 |

| | |
|--|-----|
| Figure E.5 - Expert SINTA Simplified Architecture | 155 |
| Figure E.6 - Objectives | 156 |
| Figure E.7 - New rule creation | 156 |
| Figure E.8 - Rule Edition Window | 157 |
| Figure E.9 - Rule window | 157 |
| Figure E.10 - Interface window | 158 |
| Figure E.11 - Sample question | 159 |
| Figure E.12 - Results window | 160 |
| Figure E.12 - Record window | 160 |
| Figure E.14 - All results | 161 |
| Figure E.15 - The Expert System | 162 |
| Figure F.1 - ADSGS - Class Diagram | 164 |
| Figure F.2 - ADSGS - Entity-Relationship Model Diagram | 165 |
| Figure F.3 - ADSGS - Main Parameters Sequence Diagram | 166 |
| Figure F.4 - ADSGS - Execution Client Sequence Diagram | 167 |
| Figure F.5 - ADSGS - Activity View Diagram | 169 |
| Figure F.6 - ADSGS - Component Diagram | 170 |
| Figure F.7 - ADSGS - Deployment Diagram | 172 |

LIST OF TABLES

| | <u>Page</u> |
|---|-------------|
| Table 2.1 - MC3 Radios and Antennas | 19 |
| Table 2.2 - Comparison of the different ground stations..... | 35 |
| Table 2.3 - Mission, advantages, disadvantaged, and frequency bands | 36 |
| Table 3.1 - Summary of ADSGS-Adapted Hardware specifications | 43 |
| Table 3.2 - Commercial companies that offer solutions GS | 44 |
| Table 4.1 - Questions/Answer | 70 |
| Table 4.2 - Rule 1 | 71 |
| Table 4.3 - Rule 2..... | 72 |
| Table 4.4 - Rule 3..... | 72 |
| Table 4.5 - Rule 4..... | 72 |
| Table 4.6 - Rule 5..... | 73 |
| Table 4.7 - Rule 6..... | 73 |
| Table 4.8 - Rule 7 | 73 |
| Table 4.9 - Rule 8..... | 74 |
| Table 4.10 - Rule 9..... | 74 |
| Table 4.11 - Rule 10..... | 74 |
| Table 4.12 - Rule 11 | 74 |
| Table 4.13 - Rule 12..... | 75 |
| Table 4.14 - Rule 13..... | 75 |
| Table 4.15 - Rule 14..... | 75 |
| Table 5.1 - Extended-visibility matrix for: <i>1-satellite to n-ground stations</i> | 85 |
| Table 5.2 - Preferences to a n-satellite m-ground stations | 87 |
| Table 5.3 - Allocation result from the Adapted Hungarian Algorithm | 88 |
| Table 5.4 - Ground Stations | 89 |
| Table 5.5 - Satellites..... | 90 |
| Table 5.6 - Satellite access conflicts identified during the STK simulation | 91 |
| Table A.1 - Element values for AMSAT-BR..... | 122 |
| Table B.1 - Comparisons with other common SDR | 124 |

| | |
|--|-----|
| Table B.2 - Comparisons with other common General Purpose RTL-SDR | 127 |
| Table C.1 - Original matrix..... | 131 |
| Table C.2 - Subtract row minima | 132 |
| Table C.3 - Subtract column minima | 132 |
| Table C.4 - Cover all zeros..... | 133 |
| Table C.5 - Create additional zeros (a) | 133 |
| Table C.6 - Cover all zeros with a minimum number of lines (a) | 134 |
| Table C.7 - Create additional zeros (b) | 134 |
| Table C.8 - Cover all zeros with a minimum number of lines (b) | 135 |
| Table C.9 - Create additional zeros (c)..... | 135 |
| Table C.10 - Cover all zeros with a minimum number of lines (c) | 135 |
| Table C.11 - The optimal assignment..... | 136 |
| Table C.12 - The optimal assignment, original matrix | 136 |
| Table D.1 - Artificial Intelligence Techniques | 147 |

LIST OF ABBREVIATIONS

ADSGS - Autonomous and Dynamic System Ground Station
AFSCN - Air Force Satellite Control Network
AI - Artificial Intelligence
AMSAT - Radio Amateur Satellite Corporation
ASCII - American Standard Code for Information Interchange
AUS - Authentication Server
CCSDS - Consultative Committee for Space Data Systems
CGA - Common Ground Architecture
COTS - Commercial Off-the-Shelf
CSA - Canadian Space Agency
DARPA - Defense Advanced Research Projects Agency
DoD - Department of Defense
DSN - Deep Space Network
DSP - Digital Signal Processing
ESA - European Space Agency
FCFS - First-Come-First-Served
FOP - Flight Operational Plans
GENSO - Global Educational Network for Satellite Operations
GMS - Ground Station Management Server
GPS - Global Positioning System
GROWS - Ground Station Operation Web Service
GSML - Ground Station Markup Language
GSN - Ground Station Network
GSS - Ground Station Server
GUI - Graphics User Interface
HAL - Hardware Abstraction Layer
HF - High Frequency
HTTP - Hyper Text Transfer Protocol

IARU - International Amateur Radio Union
INPE - Instituto Nacional de Pesquisas Espaciais
IP - Internet Protocol
ISEB - International Space Education Board
ISIS - Small Satellite Ground Station
ISS - International Space Station
JAXA - Japan Aerospace Exploration Agency
JPL - Jet Propulsion Laboratory
KBS - Knowledge Based Systems
LEO - Low Earth Orbit
MATLAB - Matrix Laboratory
MC3 - Mobile CubeSat Command and Control
MCC - Mission Control Client
MGSN - Mercury Ground Station Network
NASA - National Aeronautics and Space Administration
NL - Network Level
NORAD - North American Aerospace Defense Command
NPS - Naval Postgraduate School
NRL - Naval Research Laboratory
OSAGS - Open System of Agile Ground System
OSCAR - Orbital Satellite Carrying Amateur Radio
SAFB - Schriever Air Force Base
SATNet - Satellite Network
SATNOGS - Satellite Networked Open Ground Station
SDR - Software Defined Radio
SGLS - Space Ground Link System
SL - Session Level
SSDL - Space Systems Development Laboratory
STAC - Satellite Tracking and Command Station
STK - System Tool Kit
TCP - Transfer Control Protocol

TCP/IP - Transfer Control Protocol/Internet Protocol
TDI - Teses e Dissertações Internas
TLE - Two-Line Element
TNC - Terminal Node Controller
TT&C - Telemetry, Tracking, and Command
UHF - Ultra High Frequency
UML - Unified Modeling Language
UNISEC - University Space Engineering Consortium
VHF - Very High Frequency
VHL - Virtual Hardware Level
VPN - Virtual Private Network
XML - eXtensible Markup Language

CONTENTS

| | <u>Page</u> |
|--|-------------|
| 1. INTRODUCTION | 1 |
| 1.1. Motivation and Problem Definition | 1 |
| 1.2. Objectives and Expected Outcomes | 2 |
| 1.3. Solution Outline..... | 3 |
| 1.4. Thesis Structure..... | 4 |
| 2. BACKGROUND AND LITERATURE REVIEW | 7 |
| 2.1. Deep Space Network (DSN) | 7 |
| 2.2. The Air Force Satellite Control Network (AFSCN) | 8 |
| 2.3. Mercury System (MERCURY)..... | 10 |
| 2.4. The Ground Station Network (GSN) | 13 |
| 2.5. The Mobile CubeSat Command and Control (MC3) | 16 |
| 2.6. Global Educational Network for Satellite Operations (GENSO) | 19 |
| 2.7. The Satellite Networked Open Ground Station (SatNOGS)..... | 27 |
| 2.8. Satellite Network (SATNet) | 31 |
| 3. THE ADSGS ARCHITECTURAL AND MODELING SOLUTION | 39 |
| 3.1. Hardware and Software Components ADSGS | 40 |
| 3.1.1. The ADSGS-Adapted Ground Station Hardware..... | 41 |
| 3.1.2. The ADSGS-Adapted Ground Station Software | 45 |
| 3.1.2.1. Software Defined Radio | 45 |
| 3.2. External Components | 47 |
| 3.2.1. SATNet..... | 48 |
| 3.2.2. Orbitron | 52 |
| 3.2.3. ProEst | 54 |
| 3.2.4. Modem Software | 55 |
| 3.2.5. Flight Operational Plan - FOP | 57 |
| 3.3. ADSGS Component..... | 58 |
| 3.3.1. User Interface..... | 58 |
| 3.3.2. Agents | 59 |

| | | |
|----------|---|-----|
| 3.3.2.1. | A Naive Agent | 59 |
| 3.3.2.2. | A FCFS Agent | 60 |
| 4. | THE PROPOSED ADSGS AGENT | 61 |
| 4.1. | The ADSGS Agent..... | 61 |
| 4.1.2. | Actors for ADSGS | 63 |
| 4.1.3. | Rule-based system implementation - Expert SINTA | 65 |
| 4.1.3.1. | The Shell | 66 |
| 4.1.3.2. | Questions | 69 |
| 4.1.3.3. | Rules for Expert SINTA | 71 |
| 4.2. | The ADSGS UML Modeling | 75 |
| 4.2.1. | Detailed Workflow of the ADSGS Agent..... | 76 |
| 4.2.2. | Databases of the Project | 81 |
| 4.2.3. | ADSGS Suggested Technologies | 81 |
| 5. | CASE STUDY AND RESULTS | 83 |
| 5.1. | Scenario #1. 1 Satellite to N Ground Stations..... | 83 |
| 5.2. | Scenario #2. N-Satellites and M-Ground Stations..... | 86 |
| 5.3. | A Brief Simulation Scenario using STK with MATLAB | 88 |
| 6. | CONCLUSIONS AND FUTURE WORK | 95 |
| 6.1. | Conclusions and final considerations..... | 95 |
| 6.2. | List of Publications Attained | 98 |
| 6.3. | Envisaged Future works | 99 |
| | REFERENCES..... | 101 |
| | APPENDIX A - THE ADSGS-ADAPTED GROUND STATION HARDWARE. | 113 |
| | APPENDIX B - SOFTWARE DEFINED RADIO | 123 |
| | APPENDIX C - HUNGARIAN ALGORITHM..... | 129 |
| | APPENDIX D - BRIEF CONCEPTS ON ARTIFICIAL INTELLIGENCE..... | 137 |
| | APPENDIX E - EXPERT SINTA..... | 151 |
| | APPENDIX F - THE ADSGS UML MODELING..... | 163 |
| | ATTACHMENT A - ABSTRACTS, POSTERS AND PAPERS..... | 177 |

1. INTRODUCTION

After being launched into the space, a variety of factors come into play for the success of the mission of a satellite. One of these factors for success will depend on the operation of the ground station which receives, validates, interprets, and sends data commands to the satellite.

A space system is divided into two segments (AGUIRRE, 2012): the space segment and the ground segment. The former, as it is well known, consists of space vehicles like a satellite or a spacecraft that contains equipment used for payloads. The latter is composed by an infrastructure that includes the ground station and the qualified personnel to monitor, operate and perform the technical maintenance of the satellite and the ground station.

Likewise, according Aguirre (2012), the ground segment consists of a ground system and the organization of the terrestrial operations. The first one is comprised by the human resources responsible for data management of the mission and the execution of the control operations of the satellite. Ground systems consist of ground elements (hardware, software, and installations) that offer support to control and monitor the satellite. This research focuses on studying the ground system dedicated to the operation of small satellites.

1.1. Motivation and Problem Definition

Thirty years ago, Muller and Vermillion (1987) stated that a ground station should not operate isolated since it can be connected to other ground stations and begin benefit from them. Henceforth many have created networks of ground stations to take advantage of the resources sharing. Furthermore, this setup can optimize the time satellites use to communicate the different stations. Satellite operators can use internet to send commands and receive telemetries without being affected by the location of the station or satellite. These actions can be feasible thanks to the existence of a ground station connected to other stations located in different parts of the world.

In order to reduce the operation costs these stations could be autonomous in the control and tracking of satellites. Besides that, these stations shall be dynamic in the sense they must be flexible enough to track different types of satellites. In particular, small satellites like the nano-satellites, pico-satellites and femto-satellites usually operate in UHF, HF, and S-band radio frequencies. This feature allows both radio amateurs and the universities to develop and share cost-effective satellites and ground stations solutions in order to fulfill their underline mission.

The operator of a ground station must be qualified to interpret data received, operate the equipment and identify abnormalities that might put the mission at risk. Nevertheless, maintaining an operator in a 24x7 scheme this is, 24 hours by 7 days per week, on a ground station represents high costs for the project. Therefore, any suitable procedure to automate processes in its operation and/or extent its coverage helps to reduce costs while maintaining reliability and maximizing the possibilities of mission success.

The constant miniaturization of satellites, their high operation costs and maintenance on ground stations as well as recently manifested INPE's commitment to the development of scientific satellites, they all offer a great opportunity for the challenges of this project.

1.2. Objectives and Expected Outcomes

The main objective of this study is to propose an autonomous and dynamic approach to small satellite space operations in a networked scenario of ground stations. Therefore, a middleware shall be proposed based on artificial intelligence concepts and using a rule-based system and knowledge processing. This approach will allow the realization of an autonomous and dynamic ground stations configuration which will react to events in this scenario.

This approach will also contribute to reduce operation costs since it is possible to reuse resources provided and flexibly changing the underlying infrastructure.

A way of achieving this goal will be by developing a methodology to allocate a particular satellite to the network of ground stations. This study will work with the SATNet network (SATNET, 2016) which connects a set of ground stations that are located in Spain, United States and Brazil, among others ones in formation.

1.3. Solution Outline

The main solution is the development of a middleware that provides autonomy and dynamism to ground stations. This middleware has been named ADSGS which stands for “**A**utonomous and **D**ynamic **S**ystem **G**round **S**tation”. It acts as an interface layer between the ordinary ground station hardware and the SATNet network.

The solution envisaged proposes the use of Commercial Off-the-Shelf (COTS) hardware in the ground stations associated to the SATNet as well as provides the technical configurations and the way in which the ground stations may be federated to this network.

Part of the proposed solution uses Artificial Intelligence (AI) in the ADSGS network agent who controls the allocation of ground station. The agent is based on a specialized rule-based system and employs a modified version of the Hungarian Algorithm (KUHN, 1955) that provides autonomy and dynamism to the ground station network.

This thesis proposes a model for the creation of a ground station that based on different technologies, allows the control and tracking of small satellites. This ground station differs from other existing ones, since the use of artificial intelligence, which will give autonomy to the same, the use of Software Defined Radio (SDR) as a software component will give dynamism to the station.

Additionally the hardware suggested with COTS elements will significantly reduce the value of the ground station, both in its acquisition, management and maintenance.

The addition of these components (AI, SDR, Hardware setup) make the proposed ground station different from existing ones and this thesis will contribute to the academic community in developing new ways to control and tracking small satellites. Finally the integration of this station to the SATNet network will allow a greater control of the satellites put into orbit, and the use of the ADSGS agent will give the autonomy and dynamism raised in this thesis.

1.4. Thesis Structure

This thesis report is comprised of 6 chapters structured as described hereafter.

A brief literature review and background information regarding ground station networks and major topics are explored in Chapter 2. It introduces the different architectures and use of these networks in the last years. Deep Space Network (DSN), Air Force Satellite Control Network (AFSCN), MERCURY, Ground Station Network (GSN), Mobile CubeSat Command and Control (MC3), Global Educational Network for Satellite Operations (GENSO), SATNOGS and SATNet are part of this study.

Chapter 3 explores the selection of ground station design equipment for the ADSGS architecture proposal. This chapter also introduces the analysis of the different hardware configurations of the selected ground station. Likewise, it describes the different software and hardware components proposed in the ADSGS such as the SATNet base, the Flight Plan Operation (FPO), the User Interface, Orbitron, the ground station scheduler, SDR, Telemetry, Tracking, and Command (TT&C) and ProEst.

On chapter 4 the ADSGS Agent is described with the components necessary for development this system. The actors, the rules-based system, the shell, the

questions and rules used in the SINTA expert are described in this chapter and the modeling on Unified Modeling Language (UML) is proposed.

Chapter 5 analyses the case study and results. Two scenarios are studied: scenario 1: 1 Satellite n Ground Station. Scenario 2. n Satellites m Ground Station and the brief simulation scenario using MATrix LABoratory (MATLAB) with System Tool Kit (STK).

Finally, chapter 6 presents the conclusions of this study by recalling the project objectives and outcomes. Some implications and areas for further research are discussed as well and a list of publications attained.

2. BACKGROUND AND LITERATURE REVIEW

This chapter contains the background and literature review concerning main thesis topics such as ground station networks (GSN), artificial intelligent (AI) and related works. In this section the existing GSN so far is described and a review is given on the literature found since the year 2004. In the description, emphasis is put on those networks that have some kind of automation or dynamism. At first, GSN was created for large satellites, most of them military. Later on, the GSN for small satellites was developed.

As from 1950, the first space programs began; each one had its own unique architecture and Control & Command Centers. Most of the satellites were on Low Earth Orbit (LEO) and had many limitations as regarding the Earth-satellite communications. This required that the ground stations were located in strategic points in order to optimize the coverage and increase the access duration to satellites. As a result of this idea, GSN distributed all over the world began to emerge and the necessity arose to standardize their frequencies of operation.

The first networks that created to track and control big military satellites are described below and they had importance in the development of others GSN.

2.1. Deep Space Network (DSN)

In 1958, the National Aeronautics and Space Administration (NASA) created the Deep Space Network with the objective of providing spacecraft with autonomous communication (NASA, 2014). This network spread worldwide allowed the operation of three initial ground stations located in Spain, the United States, and Australia.

These stations were strategically located at 120 degrees from one to another and allowed the constant observation of the deep space. Each station counts with a complex infrastructure with antennas of up to 70 meters. All this

information received by the antennas about the deep space is processed from the Jet Propulsion Laboratory (JPL) (JPL, 2016).

The DSN is only for the tracking of large spacecrafts and it is still in operation. In (DSN, 2016) it is possible to see the antennas as well as the spacecraft being monitored, their characteristics and, their current locations.

2.2. The Air Force Satellite Control Network (AFSCN)

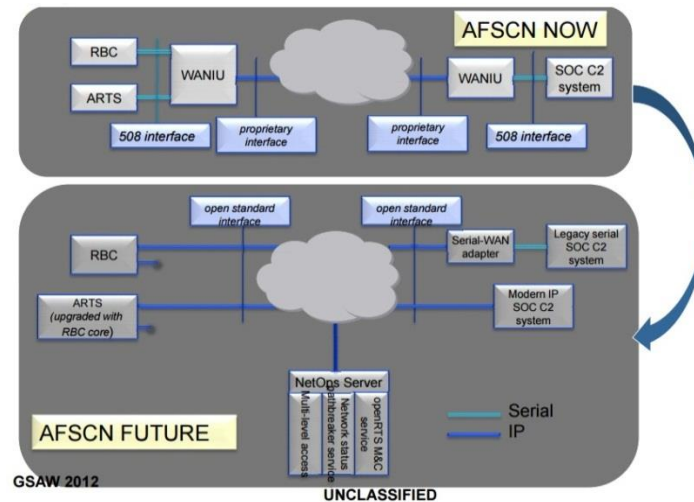
The Air Force Satellite Control Network was created in 1959 with the objective of giving support to the satellites of the US Department of Defense (DoD) (SPIRES and STURDEVANT, 1990). AFSCN was created to transmit telecommands to the satellites in orbit from ground station networks distributed around the world. AFSCN is a global system which provides command, control, and communications with spacecraft and controls around 80 communication, navigation, meteorological and missile prevention satellites as well as other DOD, NASA and UK missions (FAS, 2016).

According to Devi and Williams (2012), at present there is an improvement being made of the whole system as shown in Figure 2.1. The modernization includes changing the serial communication system and the use of the Internet Protocol (IP) for the communication between the different network nodes and, at the same time, the use of interfaces based on an open architecture so that its ground systems can overcome the limitations of the current network.

Nowadays, the DNS and AFSCN have joint operations for managing their satellites and having nodes located worldwide, although the operations are run from the Sunnyvale node, in California, where there is a backup of the Primary Operating Node located in Schriever Air Force Base (SAFB), near Colorado Springs, Colorado. The number of missions and satellites supported has increased considerably. Each satellite operates on a different frequency. For this reason, the Space Ground Link System (SGLS) has standardized the use

of frequencies. SGLS operates in communication bands S and L in 1755-1850 MHz frequencies for uplink and 2200-2300 MHz for downlink.

Figure 2.1 - Modernization Interfaces undergone on the AFSCN network



Source: Devi and Williams (2012)

The AFSCN operates under the command of Air Force Command and the 50th Space Wing, with a branch in SAFB. It is also a primary node and controls 8 remote tracking stations (RTS) located in Hawaii, California, Colorado, New Hampshire, Greenland, England, Diego García, and Guam. These remote places are interconnected and transmit telecommands, telemetries and, mission data to many satellites in many orbits (AIR FORCE, 2016).

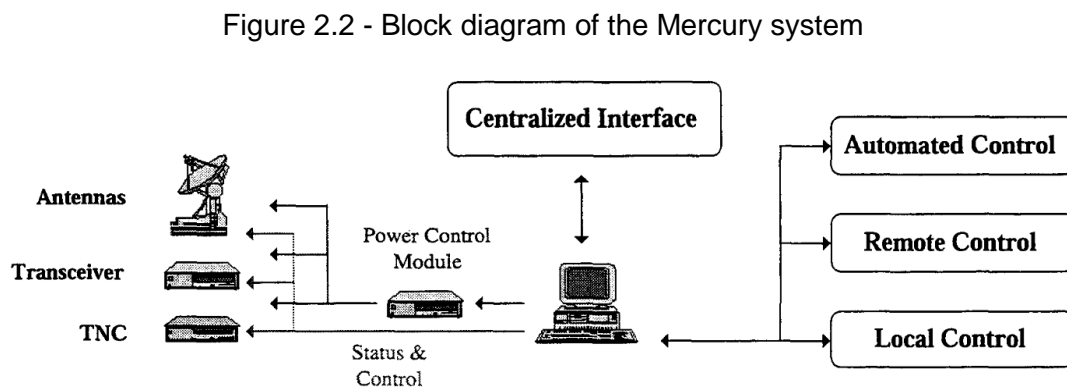
The next section has a description of the GSN whose main function is to trace and control small satellites. The GSN that will be presented in this section are: Mercury, GSN, MC3, GENSO, SatNOGS, and SATNet. These last two are currently in development. The six previously mentioned ground station networks operate in the amateur radio frequency bands Ultra High Frequency (UHF), Very High Frequency (VHF), and S-band.

2.3. Mercury System (MERCURY)

In the US at Stanford University's Space Systems Development Laboratory (SSDL), the Mercury network was developed with the idea of giving autonomy of control and the remote operation to ground stations of the type Orbital Satellite Carrying Amateur Radio (OSCAR) and was ready to be launched in 1999. The Mercury system implemented the SSDL OSCAR stations with operation architecture for multi-satellites and multi-stations (CLUTER, 2006).

The SSDL ground station is low cost and is equipped with 2 m and 70 cm antennas using the 140 MHz and 430 MHz amateur radio frequency bands. The station has control of the microsatellite from the moment of launch to when it enters in orbit. Here are some characteristics of the ground stations: full duplex transmission operating in bands with an RF amplifier of 160W power output, two Yagi type antennas with preamplifiers and their positioning control engines, a Terminal Node Controller (TNC) that acts as a modem using an AX.25 protocol at rates of 9600 bauds, and finally, a computer with the control positioning drivers of the antennas.

A block diagram of how the Mercury system provides a central software interface is shown in Figure 2.2 which controls all the station devices as well as the software for autonomous operation and the internet gateway to have access to the whole system of ground stations.



Source: Cluter *et al.* (2006)

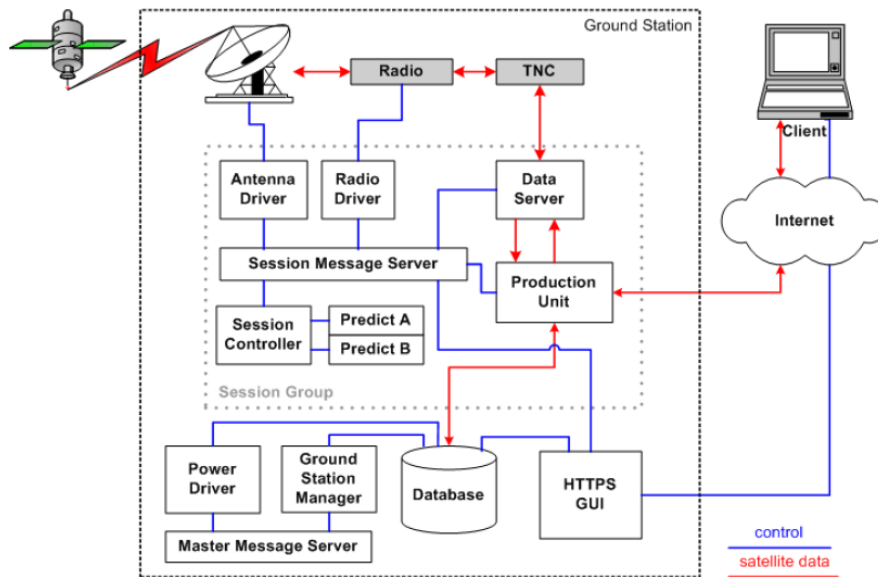
Some of the routines automatized by Mercury include: antennas positioning to track satellite passes, a closed loop that adjusts the transmission to compensate for the Doppler effect, archiving and formatting of the data collected from the TNC, configuration of the station for each satellite regarding the protocol to use, the communication frequency of the satellite, the tracking data for each session, diagnostics, and monitoring of the performance of the station equipment, update, and synchronization of the Keplerian elements.

As Cluter (2006) mentions, the TNC has an interference through a serial port of 9600 bauds, with 8 bits of data, 7 of parity, and 1 of stop. These parameters can be changed to optimize the communication with the TNC. Mercury manages the output of the TNC in binary or text format. The Mercury system, as mentioned before, is designed to work with OSCAR type ground stations. The TNC provides the modulation and the demodulation, and uses AX25 packets. Mercury is designed to work with two types of TNC, the data server provides a socket Transfer Control Protocol (TCP) that can be converted to an RS232 port.

Figure 2.3 shows the Mercury architecture and its components are (MDT, 2006):

- Antenna/Radio/Power Drivers
- Data Server
- Production Unit
- Session Controller
- Session Message Server
- Predict
- Ground Station Manager
- Master Message Server
- Database
- Hyper Text Transfer Protocol (HTTP) Graphics User Interface (GUI)

Figure 2.3 - Architecture of the system



Source: MDT (2006)

The same creators of Mercury designed a control and command language built as a framework which was called Ground Station Markup Language (GSML). It allows users to work with university ground stations around the world, which are called the Mercury Ground Station Network (MGSN) (CLUTER, 2004).

GSML is a framework based on the command language eXtensible Markup Language (XML) and controls the station on three hierarchical levels: the virtual hardware level, the session level, and the network level. The first, the Virtual Hardware Level (VHL), gives the station the capacity to generate the code and the commands for the hardware components. The Session Level (SL) captures the typical tasks and automates the services of a ground station. The user defines the session which reserves the hardware resources that will be used to maintain the communication channel connected to the destination. Finally, the Network Level (NL) captures the services of the federatively managed station network. This network coordinates the ground stations that are under its

domain, the users, the tasks, and the activities to be carried out. It enables and optimizes the connection with the satellite (CLUTER, 2004).

Professor Cluter (2006) also affirms that GSML consists of objects, methods, and a protocol to describe the interaction between these objects. The objects, described as components, among which the users, stations, and antennas are found, are captured according to the state and attributes that identify them. The methods describe the procedures for the interaction of the components to improve, for example, its efficiency.

In addition, part of the mission of the Mercury system was to give support to the operations of the QuakeSat-1 microsattellites operated by Stanford University in Fairbanks, Alaska (FGS) and Stanford, California (SGS). The stations were Commercial Off-the-Shelf (COTS) supporting satellites that operated in amateur radio frequency bands of 2 m and 70 cm, with 1200 and 9600 bps links (CLUTER and FOX, 2006).

2.4. The Ground Station Network (GSN)

The Ground Station Network was developed in Japan in 2004 (ERRICO, 2012). GSN was developed by a work group from the University of Tokyo which, suggested to a group of students from the University Space Engineering Consortium (UNISEC) to create a network system based on ground stations located in Japan and remotely controlled through the internet.

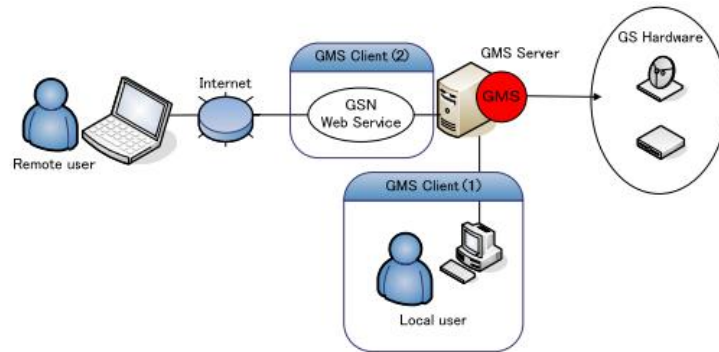
Errico (2012) also comments that the GSN network is based on Ground Station Management Server (GMS), software that remotely controls the station's hardware. Each ground station has different hardware architecture and specific drivers for the different devices and needs to implement the interface or software of the GMS client. The information exchange between the local client and the devices uses web services technology. Using GMS the operator can control radios or the TNC of the remote station, or any other hardware directly.

The system has two parts: GMS and Ground Station Operation Web Service (GROWS). GMS provides control of the ground station and the drivers of the different computers that form the ground station, and that are used by nanosatellites that, at the same time, use amateur radio frequency bands. GROWS provides secure access to the remote stations using GMS Web Services. In 2006, an international collaboration started between the United States, Japan, and Sweden in the operation of picosatellites using GSN technology (ODA et al., 2009).

What is important in this method is that the actions of each task may give rise to a total optimization. This means that it is possible to use a parallel computer (each task corresponds to a sole device node), thus an optimal result can be obtained in a short period of time. This is an advantageous characteristic of GSN, where reprogramming is frequently required (NAKAMURA; NAKASUKA, 2009).

In the GMS function manual (GSN, 2006) one can read that GMS is a “service window” software. It needs a software client to have access; in this case the local operator of the ground stations operates the computer, manipulating some formulas that will serve as the GMS client. In this case, the users can use the system remotely with a web browser that the client has installed to use that service. Each station can develop its own application. For this, it is not necessary to know the internal configuration of GMS, but it is necessary to understand the functions and interfaces of the system. In Figure 2.4 it is observed how a remote user can have remote access (through the internet) to the hardware of a ground station, using GMS for this purpose.

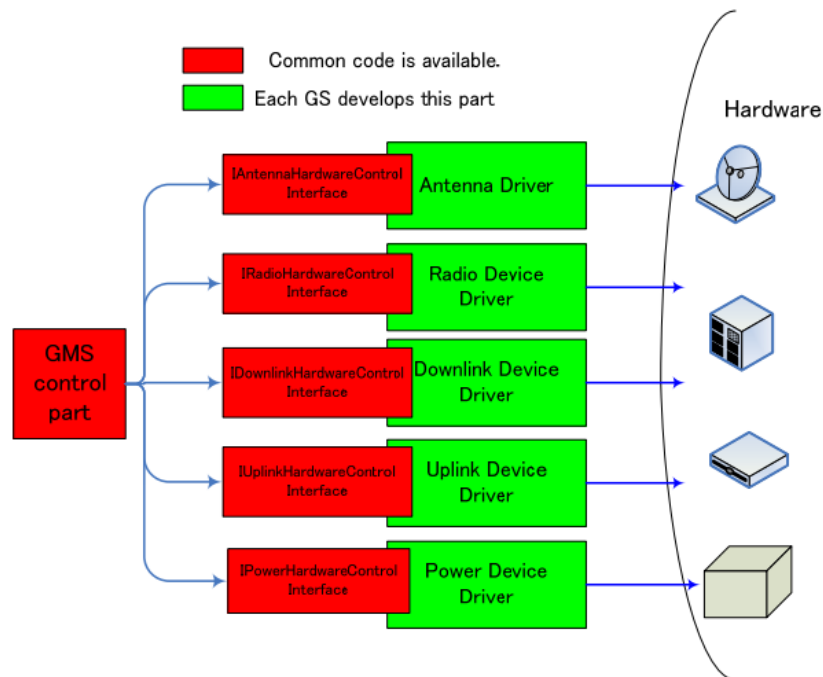
Figure 2.4 - GMS Client access



Source: GSN (2006)

On the other hand, GMS allows the creation of drivers for each device, generic or not. As can be seen in Figure 2.5, the following drivers can be accessed: power control devices, TNC for uplink, TNC for downlink, transceiver and antenna control drivers (GSN, 2006). This makes GSN system to have multiple advantages with regards to the hardware configuration of each ground station.

Figure 2.5 - Implementation of the GMS drivers



Source: GSN (2006)

2.5. The Mobile CubeSat Command and Control (MC3)

The Mobile CubeSat Command and Control network program was developed as part of the ground segment for the Colony satellite program (KALMAN, 2009; SCHULZ, 2010). It was developed to carry out experiments with CubeSat because of its low cost.

The MC3 network contains COTS hardware-type and the software was developed with the collaboration of the government and universities that participated in the Colony II program. MC3 was developed in the Naval Research Laboratory (NRL) along with the Naval Postgraduate School (NPS) in the United States, and university students with the help of the military who developed the ground stations network for CubeSat.

The MC3 architecture was designed to track the picosatellites Colony I and Colony II. The NPS and the NRL both collaborated in the design of the antennas that operated in the S (2.1 GHz.) band and UHF band (450 MHz).

The present MC3 rack, as presented in Figure 2.6, shows all the parts that make up a ground station: the radios, the S band as well as the UHF, the rotor controllers of the antennas, the rack type computer, and the power sources.

Figure 2.6 - The MC3 rack and its contents



Source: Griffith (2011)

Regarding the MC3 software, it uses Common Ground Architecture (CGA), which is an available software architecture that allows for the on-orbit testing of operations during the satellite's mission. Using CGA the whole station can automate the tracking and communication with its satellites. Users can program the passes and the software automatically assigns the resources for the tracking. For example, the necessary commands, the movement of the antennas, the satellite data, among others. CGA allows for remote access to the station through the network. Figure 2.7 shows CGA's functionality (GRIFFITH, 2011).

Figure 2.7 - CGA characteristics and capacities

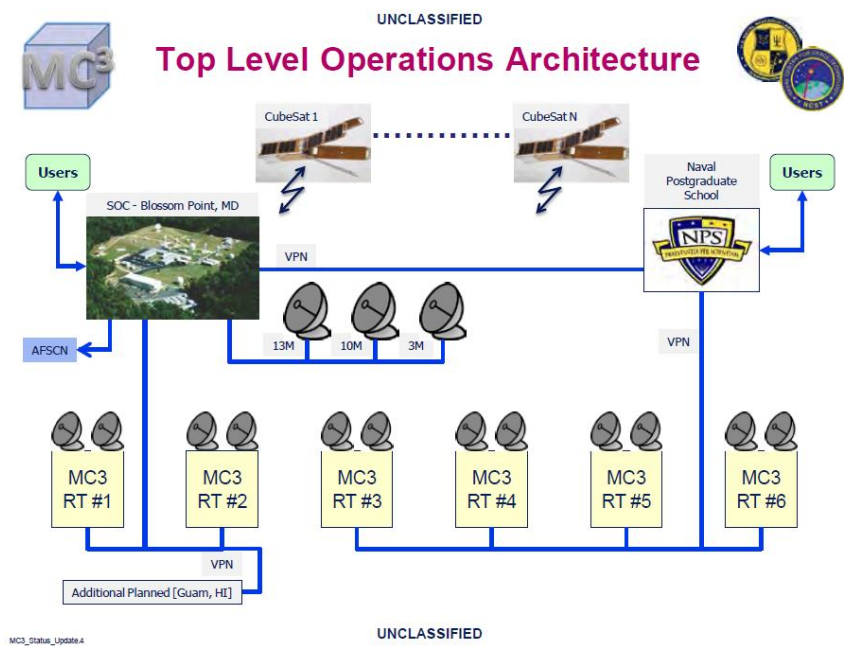
| CGA | | |
|--|--|---|
| Common | Ground | Architecture |
| Government-Owned COMMON Software Infrastructure Product | That Provides Command and Control GROUND Applications | Using a Distributed Computer and Software ARCHITECTURE |
| <ul style="list-style-type: none"> ▪ Process Communication And Data Distribution Methods ▪ Data Recording And Retrieval Toolset ▪ Graphical User Interfaces ▪ Logging Methods ▪ Database Formatting Tools ▪ Mission-Unique Software Development Toolset ▪ Task Control And Status | <ul style="list-style-type: none"> ▪ Concurrent Command And Telemetry Processing For Multiple Satellites ▪ Hardware Resource Allocation And Control ▪ Automated System And Contact Scheduling ▪ Data Recording And Analysis ▪ Operator Display And Control ▪ Orbital Determination Support | <ul style="list-style-type: none"> ▪ Multi-Node Sun Platforms/UNIX ▪ TCP/IP Network ▪ Distributed Processing And Centralized Control ▪ Windows XP Platform Connectivity ▪ Application Libraries And Interfaces For Use Of Common Software ▪ Extensible And Scalable For 1 To 255 Satellites |

Source: Griffith (2011)

The MC3 ground station network connects the different stations distributed around the world using a Virtual Private Network (VPN) through the internet. Some of the stations are located in USA: Logan, UT; Fairbanks, AL; Guam; College Station, TX; Dayton, OH; Albuquerque, NM; University of Hawaii; and Melbourne, FL.

The connection of MC3 network through a VPN over the internet is shown in Figure 2.8. MC3 receives and sends TT&C information to the bus as well as to the payload of satellite Colony II. CGA determines which MC3 station is available to communicate with the satellite, defining the time, location, priority, and availability of MC3. This gives support to many space missions.

Figure 2.8 - The MC3 architecture



Source: Arnold (2011)

Table 2.1 shows a list of radios and antennas used by MC3. They are duly certified to operate in the frequency bands previously specified.

Table 2.1 - MC3 Radios and Antennas

| Nomenclature | Purpose |
|----------------------|-----------------------|
| ICOM 9100 radio (2) | Transceiver |
| GDP radio | Receiver |
| Yagi Antenna | 450 MHz antenna |
| 917 Yagi Antenna | 915 MHz antenna |
| 1975-23 Yagi Antenna | 1925-2100 MHz antenna |
| 2227-21 Yagi Antenna | 2210-2245 MHz antenna |

Source: (GRIFFITH, 2011)

2.6. Global Educational Network for Satellite Operations (GENSO)

The international project Global Educational Network for Satellite Operations was officially created in October 2006 with its main objective being the provision of communications to university satellites and, moreover, to create an international network of university ground stations and amateur radio enthusiasts (LEVEQUE et al., 2007). With the creation of this network it is possible to download experimental data and a large number of students can participate, operating the satellites missions in real time.

GENSO began with the support and coordination of the International Space Education Board (ISEB), and was supported by the European Space Agency (ESA). This project was also integrated by several universities from the United States of America, Europe and Japan, which developed the software for the operation of the GSN network (ODA et al., 2009). As was mentioned, it has advantages regarding the access and remote control of the ground stations and its automatic programming in the stations. Likewise, GENSO is formed by the Education Departments of the Canadian Space Agency (CSA), the Japan Aerospace Exploration Agency (JAXA) and NASA. The project includes universities and institutes like California Polytechnic State University in the United States; the University of Tokyo, and UNISEC in Japan; Aalborg

University in Denmark; the Technical University of Vienna, in Austria; Lulea University, in Switzerland; AMSAT-UK in England; and SETI-Association in France (LEVEQUE et al., 2007).

The aim of GENSO is to develop a network of ground stations distributed around the world in order to offer operational support to university satellites. For this, new open source software is used, which permits capturing of mission data through the internet from different ground stations and perform mission control. The software is based on Java, developed by a team of students and amateur radio enthusiasts (LANGE; FOLEY, 2011).

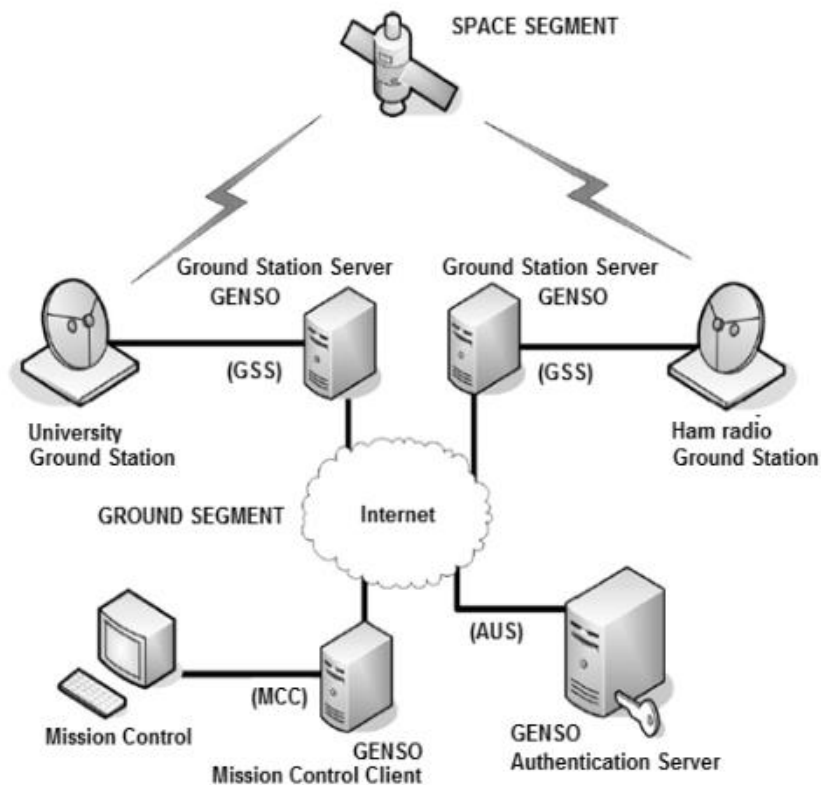
Among GENSO's objectives are the following (GENSO, 2016):

- Global access to mission operators of educational and amateur radio spacecraft.
- Remote access for operators to real-time mission data.
- Scheduling of uplinks through ground stations.
- Downlink error correction through the comparison of multiple data streams.
- Definition of standard solution for the educational ground-segment hardware, designed to optimize GENSO's performance at minimal cost.
- Recommendations for the GENSO hardware in order to improve the performance of the mission.
- The close collaboration with amateur radio enthusiasts to support a common interface in order to apply for frequency allocation and coordination.

Access to the GENSO network is possible using one or more software applications. The ground station's operator uses Ground Station Server (GSS)

and the controller of the mission runs Mission Control Client (MCC). The access security to the network is run by the Authentication Server (AUS), which is in charge of verifying that all the network users have licenses. In Figure 2.9, there is a basic diagram of its architecture (DASCAL et al., 2011).

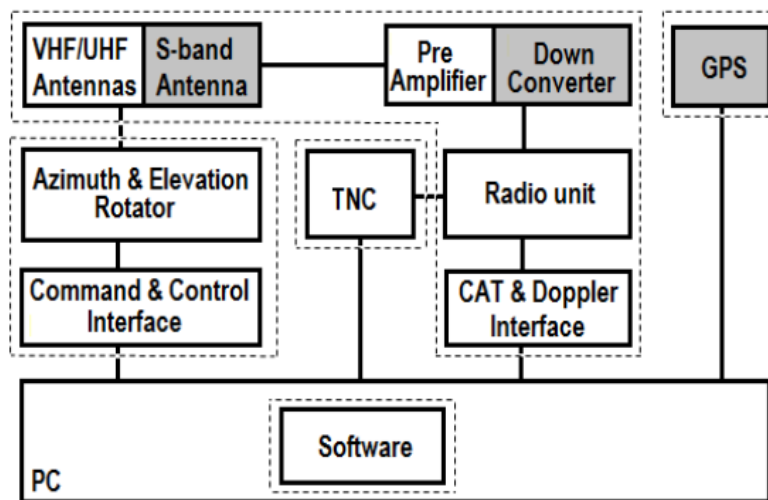
Figure 2.9 - Architecture of the GENSO network



Source: Dascal *et al.* (2011)

GENSO's hardware architecture is shown in Figure 2.10. There it can be observed the hardware as well as the software integrated within the ground station.

Figure 2.10 - Design of GENSO ground station type VHF/UHF



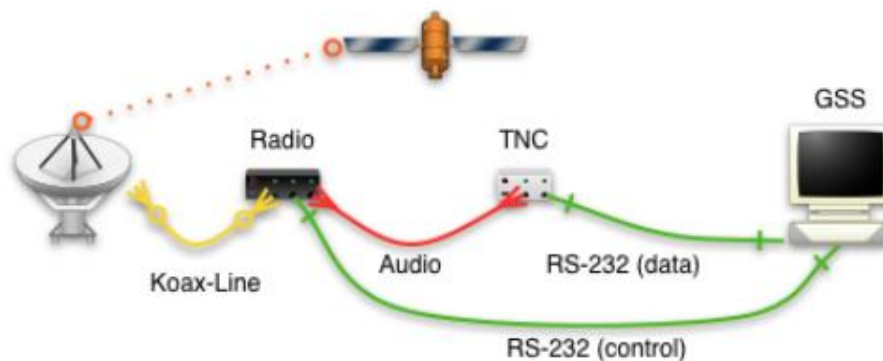
Source: Melville (2008)

The GENSO's architecture is modular and it gathers different requirements for optimal performance with the following features:

- Engine rotor module includes automatic interface of commands to rotate in azimuth or in elevation.
- The RF block depends on the use of the satellite; it has antennas for the VHF, UHF, and S bands, preamplifiers, transmitters, Doppler Effect correctors.
- The Terminal Node Controller (TNC) block is used for digital satellite communication techniques (AX25 and APRS packages).
- The Global Positioning System (GPS) block, is used for geographic coordinates.
- A PC that has the open code or proprietary software to track the satellites using Two-Line Element (TLE) and a database where the mission data is saved.

A GENSO type station has two Yagi type antennas used to receive and send amateur radio frequencies 434-437 Mhz in UHF and 2.4 Ghz in S band. The radio is used to receive the beacon signal. Figure 2.11 shows the wiring between the different components.

Figure 2.11 - Hardware design



Source: Preindl *et al.* (2009a)

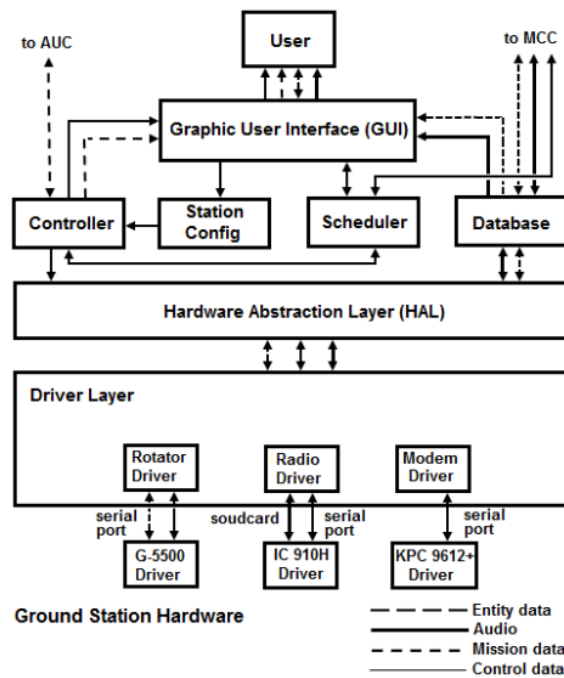
The radio is connected through an RS232 port to the TNC, and this is, at the same time, connected to the Ground Station Server (GSS) (Preindl *et al.*, 2009a).

GSS is part of the GENSO standard that describes the software and hardware used in the station. The integration of the ground station into the GENSO network involves the use of GSS software.

A block diagram of the software is shown in Figure 2.12. All the details of the hardware components (rotors, transceivers, etc.) are configured manually through the Graphic User Interface (GUI) of the GSS application. The scheduler is automatic and executes the downlink in the different passes of the satellite, and allows MCC to bi-directionally reserve the work sessions.

The control module communicates with the AUS and receives an updated list of the technical details of the GENSO satellites. All the data of the missions are archived in a database that is automatically sent to the control mission operator.

Figure 2.12 - Functional diagram of the GSS application



Source: Dascal (2011)

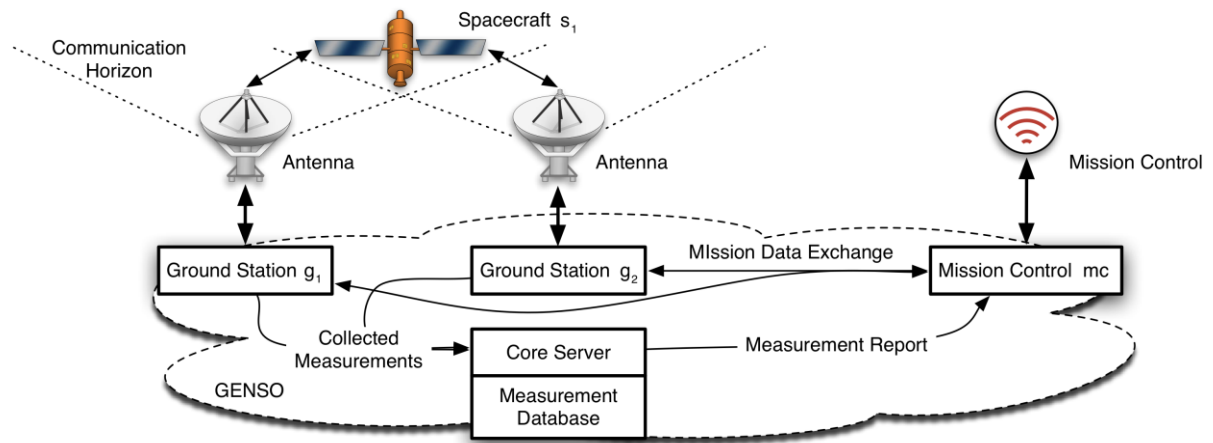
The Hardware Abstraction Layer (HAL) provides specifications of the necessary drivers for the local hardware and ensures the compatibility with GSS software. This software also has a database where the TLE is stored with the information of each satellite as well as data such as modulation, the bit rate, the frequency of operation, among others. This information is available for each satellite that passes over the ground station, and it is available for its access, such as the antenna configuration, location and, other technical characteristics.

After revising the hardware and software that form the GENSO type ground station, the development of an autonomous and dynamic approach to automate the ground station networks was proposed with the objective of reducing the operational costs (Dascal et al., 2011), but that development did not take place.

According to Preindl et al., (2009), the direct exchange of data between the mission control and the ground stations registers the quality measurements of the station's signal, and it is recorded in a central server. The server saves all

the fundamental measurements collected for scientific research as shown in Figure 2.13 as well as information for creating mission controller reports.

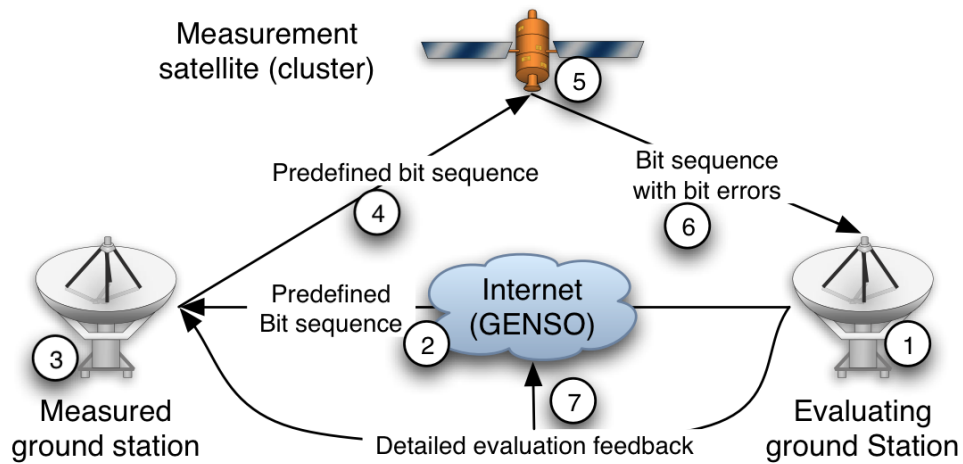
Figure 2.13 - Direct Exchange of data in a GENSO network



Source: Preindl *et al.* (2009a)

An example on how a satellite transfers the bit sequence to the ground station through the internet using GENSO is depicted in Figure 2.14. When the satellite passes over the station, it continues transmitting the bit sequence to the satellite. When the satellite passes over the main ground station, it transmits what it has saved in the buffer, including the sequence and the bit error to the station. The main station receives and sends a report to the GENSO network and the other stations (PREINDL *et al.*, 2009b).

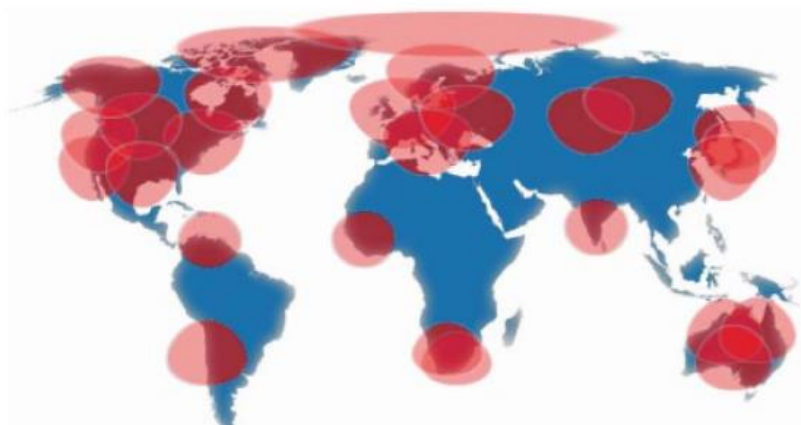
Figure 2.14 - Scheme of uplink sequence



Source: Preindl *et al.*, (2009b)

The GENSO world coverage in Low Earth Orbit in 2012 is shown Figure 2.15 which is present in all the continents. Nevertheless, in 2012 the GENSO network stopped being serviced by its creators and collaborators, leaving a big gap in the tracking of university satellites, basically because to keep the GENSO network operational, customers had to pay a lot of money, they stopped paying because access to the network was limited.

Figure 2.15 - LEO coverage projected for the GENSO network



Source: Melville (2007)

2.7. The Satellite Networked Open Ground Station (SatNOGS)

SatNOGS is a platform based on open source code standards, which is currently being developed in the University of Athens, Greece, which intends to create a modular and scalable stack for the creation of ground stations for the tracking of satellites in LEO orbit (CHARITOPOULOS, et al., 2015). SatNOGS provides the creation of a network of ground stations that are easy to build with available materials, affordable, and designed to integrate with existing and future technologies, with a variety of instruments for operating the station. The design is focused on an easy way of building, using off-the-shelf type components, and quick build materials including 3D printers.

According to Charitopoulos (2015), all designs are available under Open Hardware License (CERN OHLv1.2), maximizing the reproduction within the community that cooperates with the development of this project. The sub-systems of radio reception and transmission (RF RX-TX) use amateur radio bands (2 m and 70 cm) implemented through Yagi-Uda antennas, Low Noise amplifiers, SDR. The software of the Server as well as the Client is developed in Python (including Django web framework and PyEphem library). All the codes are under Open Software license (AGPLv3 y GPLv3).

The satellite operators have the information on the satellites in a database available through a web interface and applications for everything related with the network programmer.

The Global Management Network is a key part of the SatNOGS stack and it consists of the connection of various observers with multiple ground stations, which permits the tracking and surveillance of the LEO satellites from multiple locations worldwide. All the data gathered will be accessible to the public through their network website.

SatNOGS provides a basis for:

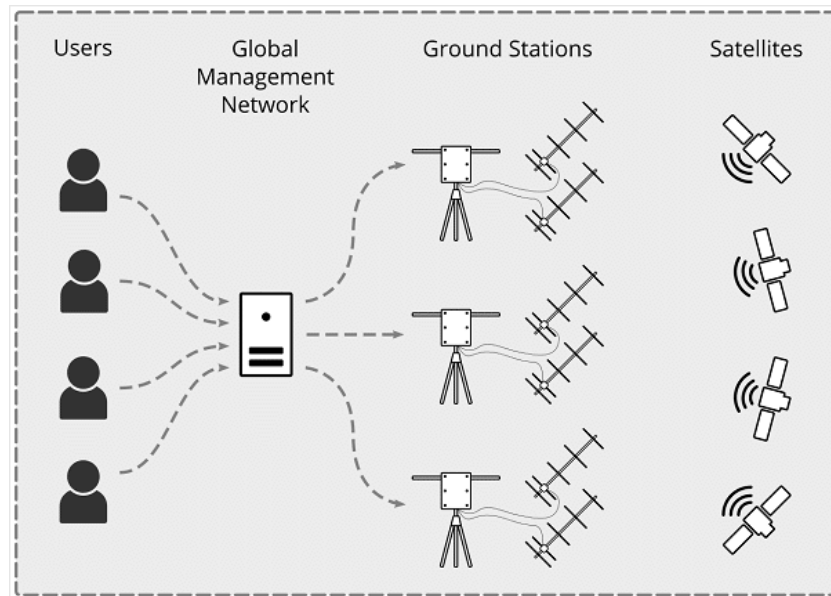
- The large scale construction and display of affordable ground satellite stations;
- Modular design for the integration of existing and future technologies;
- A platform for a variety of instruments for ground station operations;
- A suitable platform for Ground Station collaboration (one to one, one to many, many to many);
- The approach is based on the development community of ground station and
- The solution allow for massive automation of ground stations by the operator, based on open standards.

The basic components of SatNOGS are (CHARITOPOULOS, et al., 2015): (1) SatNOGS network, (2) Network connectivity, (3) SatNOGS Client (for connections with the SatNOGS network), (4) Use of gPredict and Gqrx as tracking software, (5) RaspberryPi, (6) UdroidU3, (7) Beaglebone Black, (8) A PC, (10) SatNOGS rotor, (11) Commercial rotor (YAESU G-5500), (12) UHF Antenna (yagi or helical), (13) VHF Antenna (yagi or helical).

The basic SatNogs architecture is shown in Figure 2.16 which is composed of the users, the global network, the ground stations, and the satellites.

The SatNOGS Project implements four sub-projects as part of the general stack of the network (WHITE et al., 2015):

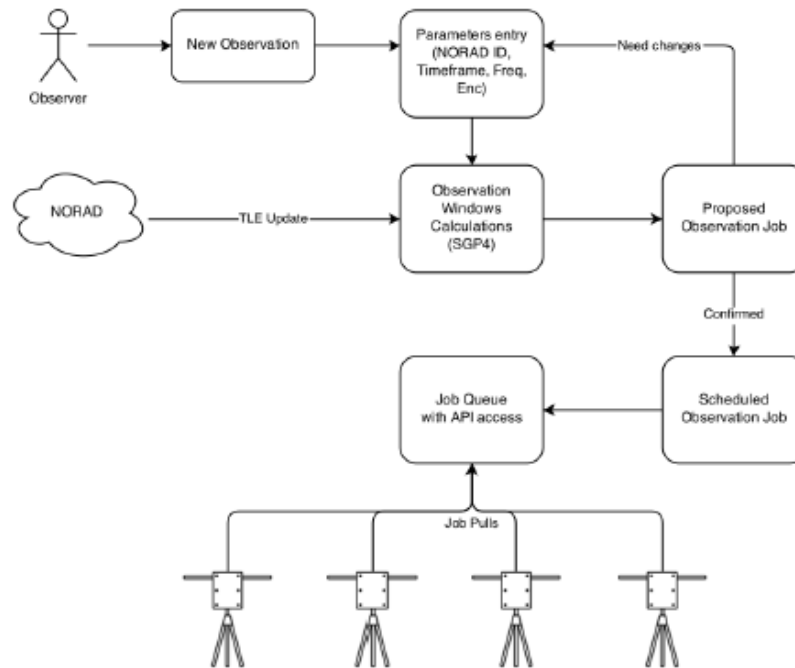
Figure 2.16 - Global vision of the SatNOGS concept



Source: WHITE *et al.*, (2015)

SatNOGS network: It provides the programming and detection of servers as well as a global management interface to facilitate the multiple operations of the ground station remotely and an observer is able to take advantage of the whole SatNOGS network of ground stations worldwide. It can be accessed by users through a web interface. They give details on the observation they want to conduct (which satellite, band, encoding signal, etc), see Figure 2.17. This information is used by the system to calculate a possible window of observation from the current available ground station connected to the network.

Figure 2.17- Scheduling diagram of the observation of the network



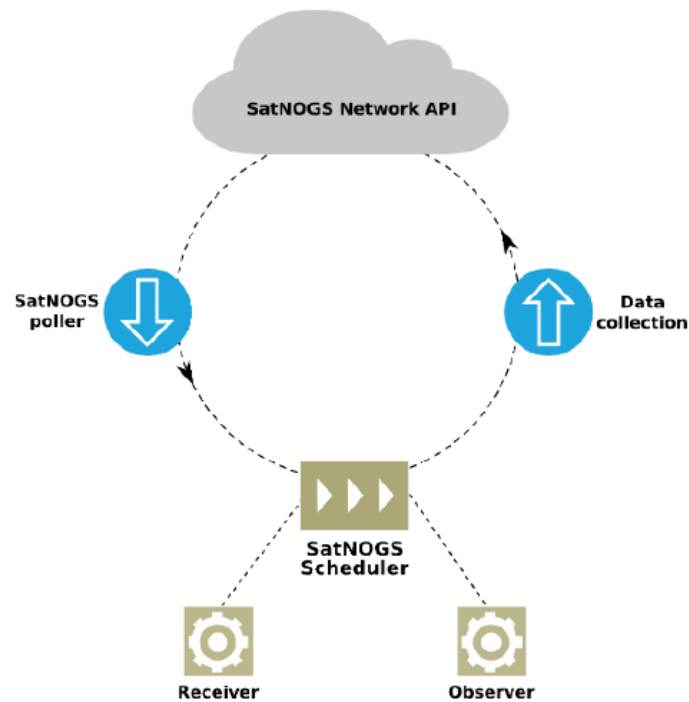
Source: White *et al.*, (2015)

SatNOGS DB: It saves information from the satellites' transponders such as the transmission frequencies and modulation; as well as the information issued by the North American Aerospace Defense Command (NORAD) regarding the catalogue number and the name of each satellite. Access to this database is open to the public. This information is used by the network for possible observations.

SatNOGS Client: Consists of software that is executed in the computer. It controls the software of the ground station, receives the operation scheduled from the network, records an observation, and sends it back. In Figure 2.18 it is possible to see the interaction of the client with the network.

SatNOGS Ground Station: There are designs available of the components of the ground station, the instruments such as a tracking system (rotor), antennas, LNA, and connection with the client. They can be made with a 3D printer.

Figure 2.18 - Components and interactions of the Client software



Source: White *et al.*, (2015)

This network is in development at the moment.

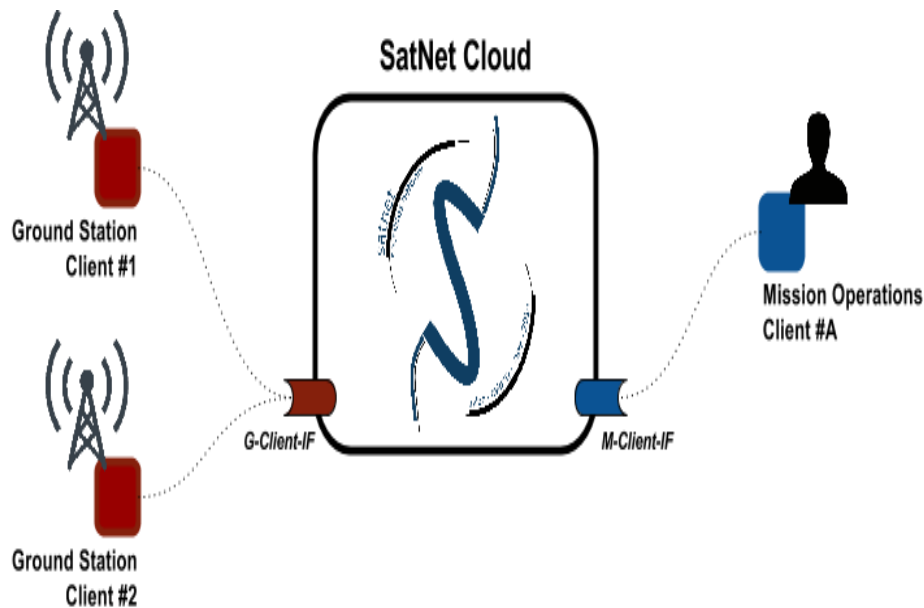
2.8. Satellite Network (SATNet)

The SATNet was created in 2013. Is a network of ground stations with a growing number of nodes around the world. Its objective is to keep more efficient control of the tracking of small satellites. With a many ground stations, it will be easy to increment the contact time with the satellite signal.

According to Tubio (2013), the basic architecture of SATNet is formed by the concepts represented in the blocks which can be observed in Figure 2.19.

Network architecture: servers distributed; N-System; main network node; G-Client; ground station client; M-Client; operator of the client mission, services provided by the central N-System, and Clients: remote access library.

Figure 2.19 - SATNet - Network basic concept



Source: Tubio (2013)

Each central node creates an isolated sub-network (similar approach to IP routers). The users decide to interconnect the nodes to adjust the scale of the sub-networks. The interconnection nodes have the capacity to create stability, redundancy, and privacy.

The SATNet network is designed to give a solution based on flexible computing in the cloud. This solution allows the use of its resources in the cloud so that the CubeSat type ground stations can have access. This repository contains the source code of a generic client of the SATNet network. This software code is open source. It allows the station to use the TNC to modulate/demodulate the frames that are being received/sent from or to the satellite.

The elements that form the network, presented in Figure 2.19 are defined as:

- A set of software clients for spacecraft operators to command remotely the satellites. From now on, they will be defined as Mission Operations Clients or M-Clients for short.

- A set of software clients for providing direct access to the services of the ground station facilities. From now on, Ground Station Clients or G-Clients for short.
- A cloud system for the coordination of the communications in between these two types of clients. From now on, Network Communications System or N-System. It is important to note that the N-System is not a single server but a cloud-computing-based system. This way and depending on further implementation decisions, this cloud system may evolve into a network of interconnected servers that will provide the required service.

The N-System implements the following interfaces to allow automatic communication between software entities without the need for direct human interaction:

- G-Client Interface (G-Client-IF) allows the clients of the ground station to connect to the network services.
- M-Client Interface (M-Client-IF) allows the clients who operate the mission to connect to the network services.
- Direct Client Interface (Direct-IF) allows the clients who operate the mission to connect to the clients of the ground station.

The current implementation of the SATNet network only gives the interfaces of the G-Client-If, M-Client-If and the direct connection IF is left for future versions. This decision was made given that it is possible to incorporate the community feedback once the initial display is made. The definition of this client to client remote interface can be adapted to the specific operational needs the Cubesat developers.

Although this approach allows the use of fully automated ground stations, it provides the possibility of human intervention during the functioning of the

CubeSats. This is a basic requirement of the software, given that the International Amateur Radio Union (IARU) imposes the presence of a human operator in the remote ground stations if an amateur radio frequency band is used (GARPESTAD, 2015). Thus, the SATNet network allows the combination of fully automated ground stations and human operators to allow different ground segments being used as part of the same network.

SATNet is part of this thesis, so it will be resumed in the next chapter as part of external components of the proposed ADSGS architecture. It is shown in Table 2.2 below a summary of the state of art made on GSN. Some common aspects are characterized, such as the paradigm used by each network, what type of information can be accessed remotely, if its administration is federative, if the scheduler is centralized or distributed, what type of security it uses, where the source codes are found, and if there is access or not, what type of software license they have and, finally, the country where it was or is being developed.

Table 2.2 - Comparison of the different ground stations

| | Mercury | GSN | MC3 | GENSO | SatNOGS | SATNet |
|-------------------|----------------------|---------------|------------|--------------|---------------------|---------------------|
| Paradigm | Client Server | Client Server | Legacy GCA | Hybrid P2P | Centralized | Distributed network |
| Remote GS | Data | Data | Data | Audio/Data | Data | Data |
| Federation | Yes | - | Yes | - | Yes | Yes |
| Scheduling | - | - | Central | Distributed | Central | Hybrid |
| Security | Central | - | Central | Central | Central | Distributed |
| Sources | Source forget (2003) | - | - | Unshared | Github ¹ | Github ² |
| License | GPLv2 | - | - | - | GNU Affero GPL | Apache v2 |
| Automated | Yes | - | - | - | - | Yes |
| Used IA | - | - | - | - | - | - |
| Country | USA | Japan | USA | ESA | Greece | Spain/USA /Brazil |

Source: Author

Of the previous table highlights that no of ground stations network have used Artificial Intelligence in its development, and only the Mercury and SATNet network use some kind of automation.

Similarly, in Table 2.3 the following is described: the mission of each one of these networks, the advantages, disadvantages, and the type of frequency bands supported by each one of these ground station networks.

¹ Available on: <<https://github.com/satnogs>>. Accessed on: 20 Apr. 2016

² Available on: <<https://github.com/satnet-project>>. Accessed on: 20 Apr. 2016

Table 2.3 - Mission, advantages, disadvantages, and frequency bands

| Network | Mission, use or applicability | Advantages, opportunities or strengths | Disadvantages, limitations or weaknesses | Frequency bands |
|--|---|--|---|-----------------|
| <i>Deep Space Network (DSN)</i> | Observation of deep space. | First networks to monitor large size satellites | Complex infrastructure | S and L |
| <i>Air Force Satellite Control Network (AFSCN)</i> | Gives support to the satellites of the DoD of the United States. | | Very high operation costs | |
| MERCURY | <ol style="list-style-type: none"> 1. Gives support to the operations of microsatellites. 2. Architecture of operation for multi-satellites and multi-stations. 3. Gives autonomy to the control and the remote operation of OSCAR type ground stations. | <ol style="list-style-type: none"> 1. Automated operation routines. 2. Full documentation of the architecture and its components.. 3. Low cost ground station.. 4. Full duplex transmission 5. Control and command language created as a <i>framework</i>, which allows work with university ground stations around the world | As the services are managed federatively, this can cause the failure of the mission if all the networks are not synchronized. | HF, VHF and UHF |
| <i>Ground Station Network (GSN)</i> | 1. Creates a network system based on the ground stations located in Japan and remotely controlled through the internet, used in the acquisition of nanosatellite and picosatellite images. | <ol style="list-style-type: none"> 1. Schedule of specific activities (centralized) according to the need of each ground station. 2. Remote configuration of the stations' hardware. 3. Network remotely controlled through the internet. | The users may have scheduling problems given that it is possible that two operators want to use the same ground station at the same time, or simply that one station is being serviced at that very moment. | UHF, VHF and S |
| <i>Mobile Cubesat Command and Control (MC3)</i> | 1. Trace the picosatellite program of the constellation Colony. | <ol style="list-style-type: none"> 1. The station can automate the tracking and the communications with the satellite. 2. Allows remote access to the station. | It was developed just for some picosatellites and its code is not open. | S and UHF |
| <i>Globla Educational</i> | 1. Provides communications to the | 1. Uses an open code, Java based software. | In 2012, it ceased to receive | VHF, UHF and |

| | | | | |
|--|---|---|---|----------------|
| <i>Network for Satellite Operation (GENSO)</i> | <p>university satellites.</p> <p>2. Creates an international network of university ground stations and amateur radio enthusiasts.</p> | <p>2. It is part of the amateur radio community.</p> <p>3. Remote access and control and automatic programming of ground stations.</p> | <p>maintenance from its creators and collaborators.</p> | S |
| SATNOGS | <p>Satellite tracking with the aim of sharing information with all the ground station networks located all around the world, in order to publish it on the networks website afterwards.</p> | <p>Contemplates four modules for each ground station:</p> <ol style="list-style-type: none"> 1. Programming and detection of servers. 2. Database with the information of the satellites' characteristics. 3. Management and automatic programming software. 4. Available designs of the components to implement ground stations. | <p>It is oriented towards the creation of software.</p> <p>It is in development, so there is no evidence on how it works.</p> | VHF, UHF and S |
| SATNet | <p>Cubesat type picosatellite tracking. For communication, navigation, and other services, such as the collection of meteorological data.</p> | <ol style="list-style-type: none"> 1. The ground stations that want to join this network do not need to change their hardware. 2. Open and available Code on the network. 3. Flexibility in the administration of the ground station. 4. Automated operation routines | <p>This network is being tested at the moment. In development.</p> | VHF, UHF and S |

Source: Author

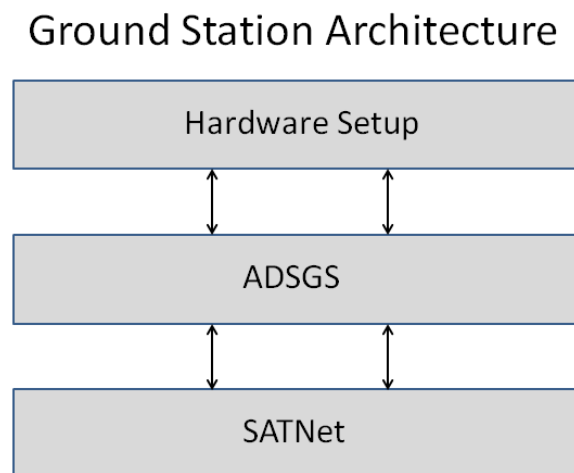
3. THE ADSGS ARCHITECTURAL AND MODELING SOLUTION

As mentioned in Chapter 2, the SATNet ground station network relies on a distributed architecture and it currently has three nodes or ground stations.

Some SATNet stations still lack architecture for ground stations; hence this chapter proposes and details an open architecture for them that will be part of this network. But it should be emphasized here that this proposal serves for any types of ground station used for tracking and control of small satellites.

A ground station generally has two basic elements: hardware and software; this work proposes the inclusion of a third one; a middleware defined under the SATNet and named ADSGS. Figure 3.1 shows the three elements proposed for a ground station used for tracking and controlling of satellites. Each of these customized components is defined hereafter in the ADSGS proposal.

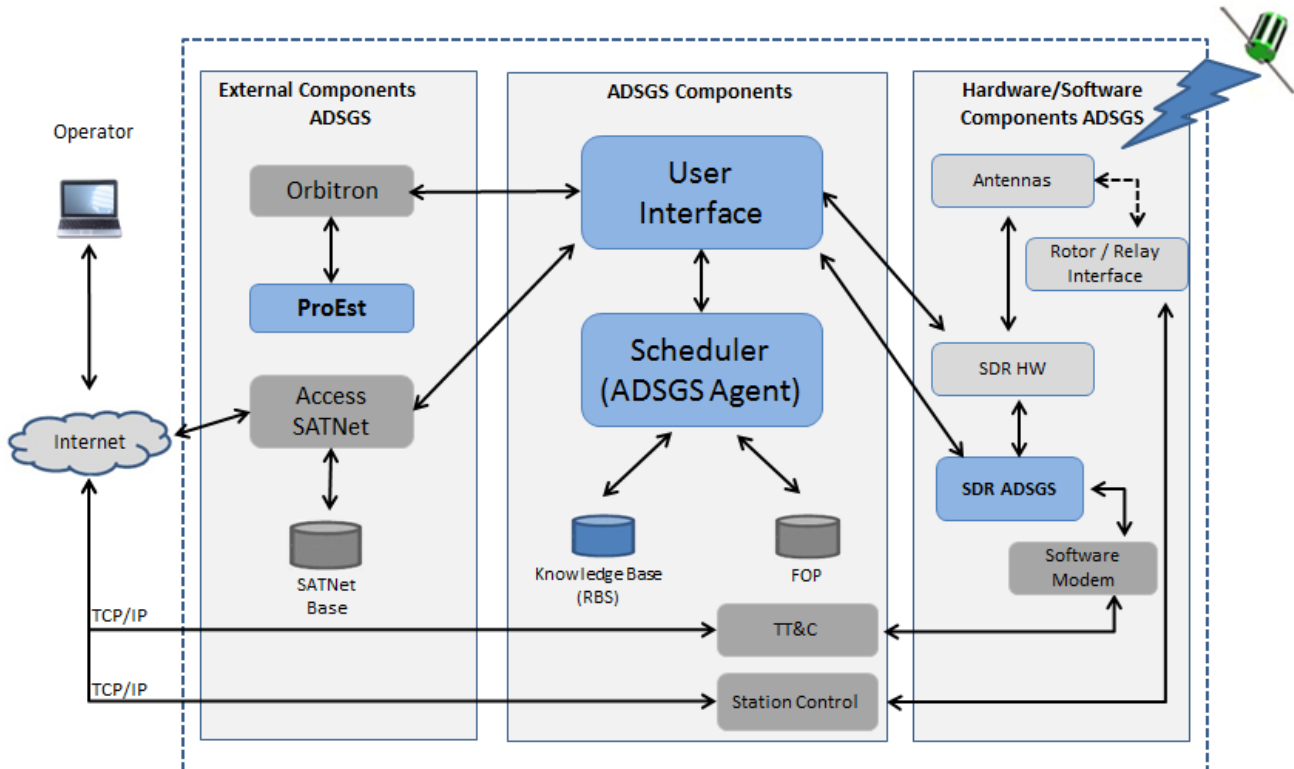
Figure 3.1 - The ADSGS middleware in typical ground station architecture



Source: Author

The general architecture of ADSGS is shown in Figure 3.2 where all the above-mentioned components are integrated: hardware, software external and internal components (proper of ADSGS) and SATNet. Each of these components will be briefly described below.

Figure 3.2 - The ADSGS Ground Station Architecture



Source: Author

3.1. Hardware and Software Components ADSGS

All the hardware in the ground station is properly connected and configured so that allows to control and track the satellites previously registered.

The operation frequencies in the ground station are the same used by the amateur radio community, i.e., 400 to 470 MHz range and 70 cm wavelength to UHF, 130 to 170 MHz range and 2 m wavelength to VHF and 2.4 GHz to S-band.

In the software case, it is going to be used the modem software to modulate/demodulate the sent/received signal and SDR software defined as SDR#. This software is responsible for receiving the signal from the satellite,

sending it to the modem to decode and storing it in a database, so that this information could be consulted by authorized users in SATNet.

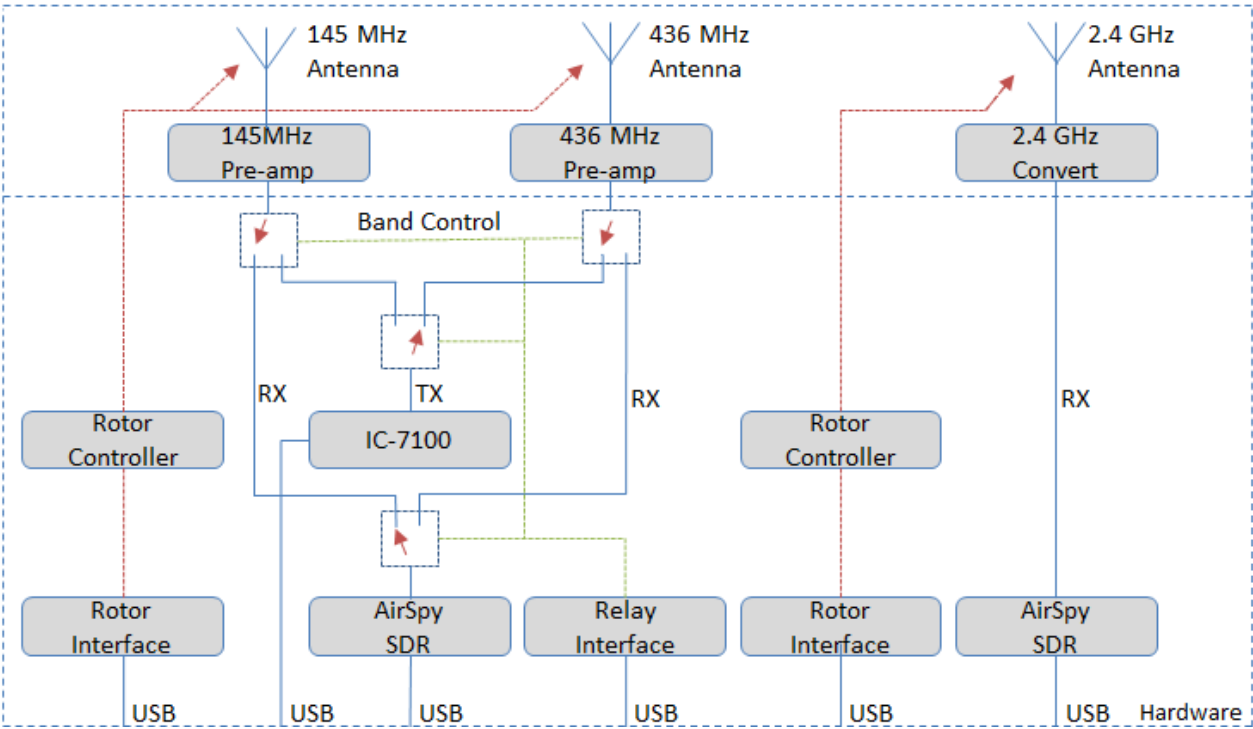
3.1.1. The ADSGS-Adapted Ground Station Hardware

This section shows the different hardware elements in the suggested ground station such as uplink and downlink, antennas defined in frequencies UHF, VHF and S-Band which allow transmitting telecommands to the satellite and receiving telemetry sent to the ground station. The movement in azimuth and elevation of these antennas is given by powered servomotors connected to their corresponding controllers, connected, by its turn to a computer prepared for this purpose.

Both UHF and VHF antennas should be connected to a radio transmitter able to send telecommands in the frequencies defined for each satellite. In the case of S-Band antenna, it should be connected to SDR the same as UHF/VHF antennas in order to receive telemetry in the frequencies defined for each satellite. In any case there must be a connection between the antennas, the radios, the selected SDR and the rotors.

The proposed ADSGS adapted hardware consists of several COTS devices, such as: antennas, preamplifiers, converters, rotors, controllers, relay interface and rotor interface, radios and connectors. This hardware architecture was proposed in collaboration with Edson Wander Pereira - PY2SDR, an amateur radio operator, member of AMSAT Brazil. Figure 3.3 shows all the different hardware components of this architecture.

Figure 3.3 - ADSGS-adapted Hardware components



Sent signal —————
Antenna connection - - - - -
SDR switching - - - - -

Source: Author

Appendix A shows the technical characteristics of each of the hardware components suggested for the creation of an ADSGS ground station, describes the UHF, VHF and S-band antennas, as well as the radios, the motors used to move the antennas, controllers for these motors, SDR, Low Noise Amplifiers (LNA), etc.

Table 3.1 shows some technical specifications of ground station hardware and summarizes appendix A.

Table 3.1 - Summary of ADSGS-Adapted Hardware specifications

| Device | Description | Interface | Brand |
|-------------------|---|----------------------------|-------------------------------|
| Antennas | - Crossboom 145 MHz - Crossboom 436 MHz - S-band 1.8m dish 2.4 GHz | SO-239 SO-239 SO-239 | M2 Antenna Systems Shaanxi |
| Pre-amp | - MSP144VDG-160 145 MHz - MSP432VDG-160 436 MHz | SO-239 SO-239 | AR2 AR2 |
| Converter | - UEK-3000 2.4 GHz | SO-239 | SSB Electronic |
| Radios | - VHF/UHF Transceiver: ICOM IC-7100 - AirSpy SDR Receivers | USB USB | ICOM AIRSpy |
| Relay Interface | - Coaxial Relays Tohtsu CX600N | USB | Tohtsu |
| Radio Transceiver | IC-7100 Transceiver | USB | ICOM |
| Rotor Controller | - VHF/UHF antenna rotor Yaesu G-5500 - Dish antenna rotor SPID, type BIG-RAS/HR Azimuth & Elevation Rotor MD01 | RS232 USB | Yaesu Rf Hamdesign |
| Rotor Interface | - GS-232A for VHF/UHF rotor - SPID BIG-RAS/HR controller for S-Band rotor | USB USB | Yaesu Rf Hamdesign |

Source: Author

There are some companies in the market, offering ground station elements and there are others offering ready-to-use ground stations. Table 3.2 shows the companies with wider impact on pico and nanosatellite market. According to NASA (2014a), the price of these ground stations oscillates between USD \$10.000 and USD \$100.000. It should be noted that the price of the proposed ground station is less than USD \$ 8,000.

According Rodriguez (2015), such companies offer ready-to-go solutions with different services. One of the best known companies is Small Satellite Ground

Station (ISIS) that features the highest TRL³ (ISIS TRL = 9), offering modular ground stations with services on VHF, UHF and S-Band frequencies. ISIS supplied two stations in Brazil, the first one for NANOSATC-BR1 project and the second one, for AESP-14.

Table 3.2 - Commercial companies that offer solutions GS

| Technology | Description | Country | TRL | USD\$ |
|---|---|---|---|---------|
| Small Satellite Ground Station - ISIS | Configuration of ground station for microsatellite and cubesat (VHF, UHF and S-Band) | Innovate Solutions in Space Netherlands | 9. It has been successfully used at least one mission: Delfi-C3, NanoSat- CBR1, Serpens, BGUSat, ITASat | 63,000 |
| Open System of Agile Ground System - OSAGS | Network of low cost, three band-S equatorial stations | Space Inc. USA | 8. Used successfully in August 2002 to operate MIT HETE-2 mission | 190,000 |
| Satellite Tracking and Command Station - STAC | Configuration of ground station for microsatellite and cubesat (VHF, UHF, L and S-Band) | Clyde Space Scotland | 8. Installed at the University of Strathclyde in Scotland. Operational for two years | 79,000 |
| Mobile Cubesat Command & Control Ground Station – MC3 | Network ground stations, totally autonomous to support the Colony program | Naval Postgraduate School USA | 6. TRL evaluation supported by Griffith (2011) | 120,000 |

Source: Adapted from NASA (2014)

The creation of this hardware setup is a clear contribution to the amateur radio community in the world as they benefit from the reduction of acquisition costs of the ground station; this also fulfills part of the objectives proposed in this thesis.

³ TRL: Technology Readiness Level, numerical scale used to express the maturity (readiness) of a technology to be used in a space mission (WERTZ et al, 2011)

3.1.2. The ADSGS-Adapted Ground Station Software

The proposed ADSGS-adapted ground station software must comply with some technical specifications to provide autonomy and dynamism to the ground station control.

This software architecture should comprise: (1) a rule-based expert system responsible for the entire station autonomy; (2) SDR to provide dynamism and, (3) a scheduler that based on an ADSGS agent is responsible for integrating the knowledge base, the SATNet database with compatible satellites with the station.

All the above components are integrated in the User Interface (UI) that uses different software such as Orbitron (TLE based monitoring), SDR to receive telemetry, ADSGS agent that gives autonomy to the station, SATNet database to manage the ground station network, the FOP (Flight Operational Plan) of each of the satellites, TT&C (Telemetry, Tracking and Control), ProEst to manage the autonomous movement of antennas and Modem Software to encode/decode the information sent/received by satellite.

3.1.2.1. Software Defined Radio

The term software radio has been first used by the employees of E-Systems Inc. in a company newsletter in 1984. The first military program that had the physical layer components of a radio implemented in software was called SPEAKeasy, designed by Defense Advanced Research Projects Agency (DARPA) in the United States. The main objective was to build a single radio that is compatible with ten different military radio protocols, can operate anywhere between 2 MHz and 2 GHz, and also have the possibility of including new modulations and protocols later. (NUTAQ, 2014).

According to Reed (2002), the term SDR was introduced by Joseph Mitola in 1991 to refer to a kind of programmable and reconfigurable radio. The term refers to those radios with a flexible software working upon a static hardware.

Radio components such as modulators, demodulators and amplifiers are traditionally hardware components. But thanks to SDR, these components are now implemented by software.

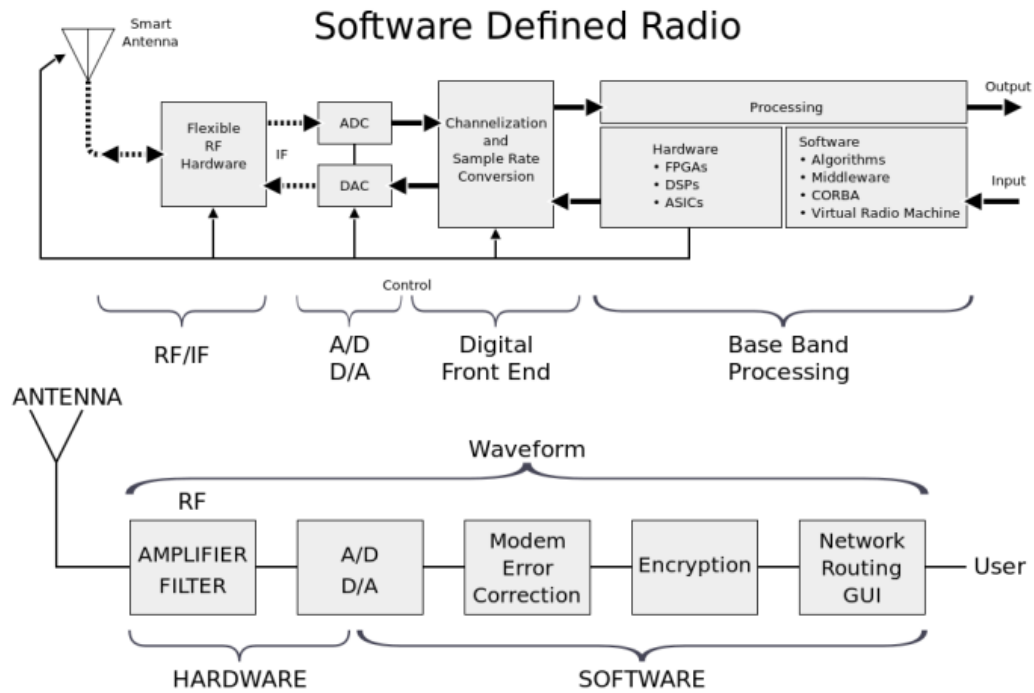
The last means that in the same hardware different function in different times can be modified and the software is able to configure and control it from an interface that allows the total control of the hardware.

When talking about SDR it also should be mentioned the Digital Signal Processing (DSP) that according to Youngblood (2002), is a high-speed digital processor for numeric operations, that in this case performs radio functions that could not be done with analog radios. DSP allowed to amateur radio community to have better performance devices, noise reduction and the ability to do a digital filter of the signal. In Figure 3.4 it is shown the block diagram of the concept of SDR.

The systems that the term Software Defined Radio covers, implement some or all physical layer functions of a radio in software instead of hardware, which also implies that the software does DSP tasks. (WIF, 2016).

Appendix B contains the most common software classification based on SDR, as well as a table showing SDR hardware.

Figure 3.4 - The SDR Concept



Source: Mcsush (2016)

3.2. External Components

SATNet, which was mentioned in section 2.1.8, is a network that being developed by a research group led by Professors Jordi Puig co-creator of Cubesat standard, Walter Abrahão dos Santos, Mauricio Gonçalves Vieira Ferreira, Ricardo Tubio Pardavila, Samuel Góngora García, Adair Rohling and Jorge Enrique Espíndola Diaz author of this thesis. The main goal of this network is to provide control and tracking services to small orbiting satellites using 3 ground stations: Calpoly, Uvigo e INPE-ITA (in the future more stations could be added).

ADSGS uses Orbitron as the software responsible for visualizing the tracking of each satellite, as mentioned Orbitron allows to track graphically and to visualize the pass of satellite over the ground station using TLE to predict it. Additionally

this software allows making a simulation of the pass of any satellite. Besides it is responsible to send azimuth and elevation data to ProEst software, which in turn is responsible for moving the antennas autonomously using data sent by Orbitron. ProEst includes a correction to Doppler Effect.

Follows a brief description of each of the external components, in this case SATNet, Orbitron and ProEst.

3.2.1. SATNet

This project still in development allows having a global ground station network, with global coverage of the satellites willing to be part of the network. This is an open source project, developed in Python and hosted at⁴. On the website⁵ it's found the user interface that gives access and control to the network. It is shown below a description of its main modules.

- SATNet Main screen and login window of the system. It is required to be registered in order to use the system.
- SATNet Ground Stations option allows to add new ground stations, set its location (longitude, latitude and elevation), set its availability schedule and the antennas parameters, i.e., accepted frequencies range in UHF, VHF and S-Band (if available), low noise amplifier type and available bandwidth.
- SATNet Spacecraft added satellites, this option allows adding and configuring, among others: the Flight Operational Plan, the configuration related to frequencies, modulation, bitrate, required bandwidth for transmission, bitrate type, mission and payload of each satellite.
- SATNet Availability. The availability of each registered ground station on the network, where it displays the time at which the station can be used.

⁴ Available on: <<https://github.com/satnet-project>>. Accessed on: 14 May. 2016

⁵ Available on: <<https://satnet.aero.calpoly.edu/>>. Accessed on: 14 May. 2016

- SATNet Compatibility. It is required to configure the compatibility between each ground station and each satellite, considering the communication channel used in each case. SATNet displays a detailed list of each case to guarantee the entire network compatibility.
- SATNet Operational. Displays the operational slots of each ground station, so that ADSGS scheduler is able to program a pass over this station if available at that time.

Each station to be registered in SATNet should provide the featured technical specifications such as antennas band (UHF, VHF, S-Band), available frequencies, low noise amplifier type, bandwidth capacity, etc.

The integration of ADSGS with SATNet will be possible through the database provided by SATNet which gives to ADSGS GUI (Graphical User Interface), the required data to know which users are registered, which satellites should be tracked, which are the parameters of each satellite (protocol, modulation type, bitrate, radio format, required bandwidth for transmission, etc.), the location of the stations in SATNet, information regarding the ground station - satellite compatibility, ground station availability, upcoming passes over ground station of registered satellites.

This database is provided by the development group of SATNet ground stations network. The database was developed in MySQL upon Django, an open source database manager which is totally compatible with Python programming.

This relational database contains tables related to all references used in SATNet.

The information stored in this database corresponds to:

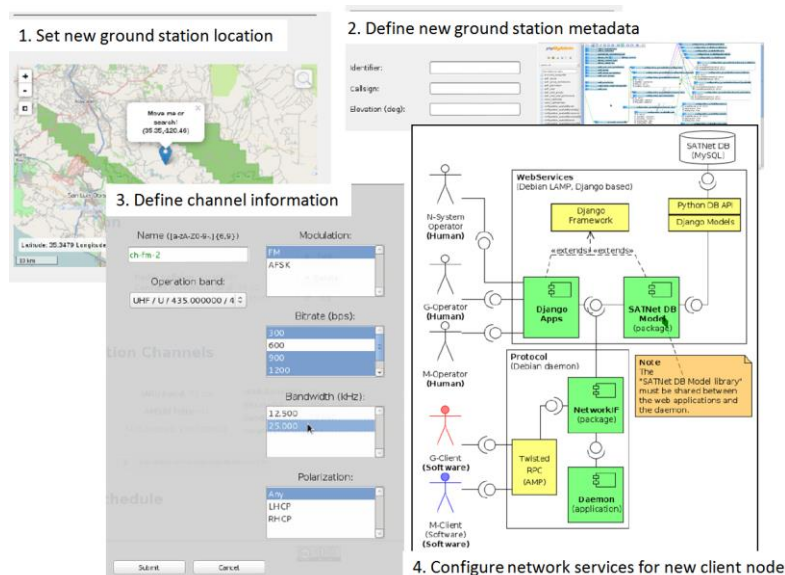
- List of compatible ground stations
- Compatibility between the different GS
- Availability

- Operability
- Satellite Communication Requirements:
 - Frequency (downlink, uplink)
 - Modulation type (AFSK, PSK, etc)
 - Bit Rate (1200 bps, 9600 bps, etc).

The integration process follows the steps shown in Figure 3.5 and, basically, some of the key parameters are defined for the new node of ground station integrating into the main network of SATNet (TUBÍO et al., 2016):

1. Establish the location of the new ground station.
2. Define new ground station metadata
3. Define the information channel for the new ground station, and
4. Configuration of network services for the additional client node.

Figure 3.5 - Simple steps to integrate of nodes into the SATNet network.



Source: Tubio (2013)

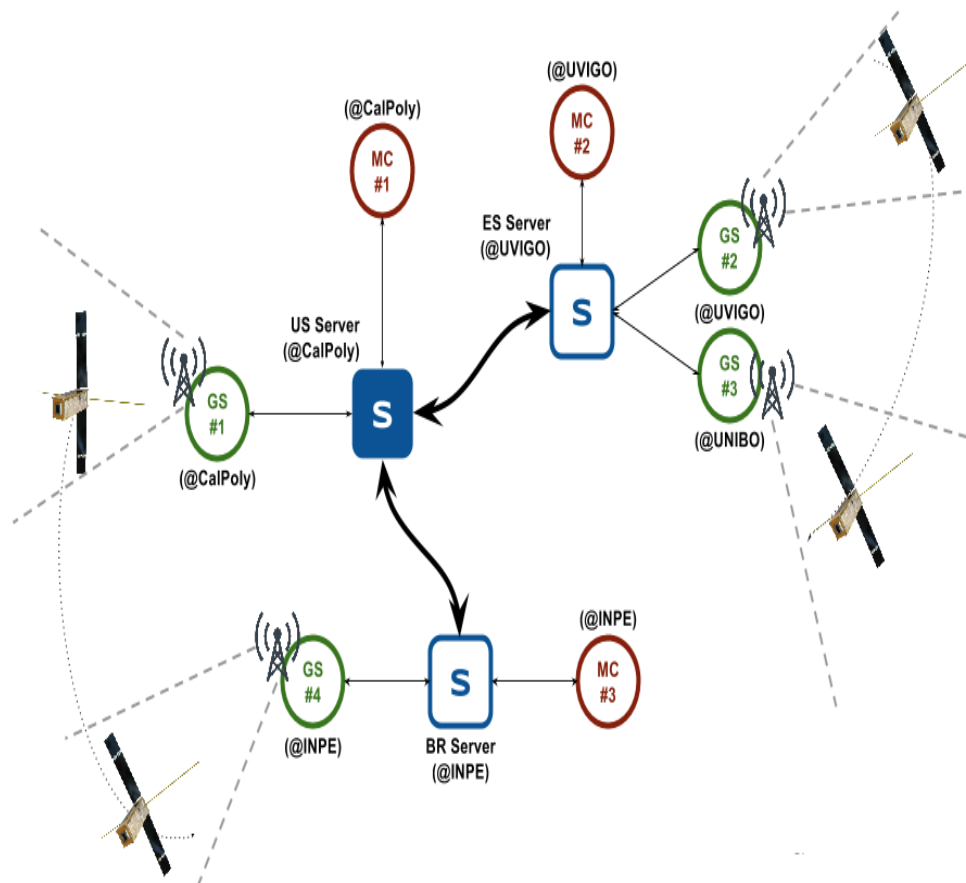
At present, some trial activities are taking place for the integration of the ground station INPE in the SATNet network. This would increase the capabilities of the

network, giving additional coverage for CubeSat remote operations in the southern hemisphere. The planned scenario is shown in Figure 3.6 where the server of the SATNet network is currently deployed in CalPoly where it is already providing operational services.

In the near future, there are two more servers to be deployed: one in the University of Vigo, and the other in the INPE, which will provide coverage for the southern hemisphere.

These three servers interconnected through the Internet and in turn to the ground stations (GS) located in the already mentioned institutions allow the control and tracking of satellites (MC).

Figure 3.6 - Current display of the INPE node in the SATNet network



Source: Author

3.2.2. Orbitron

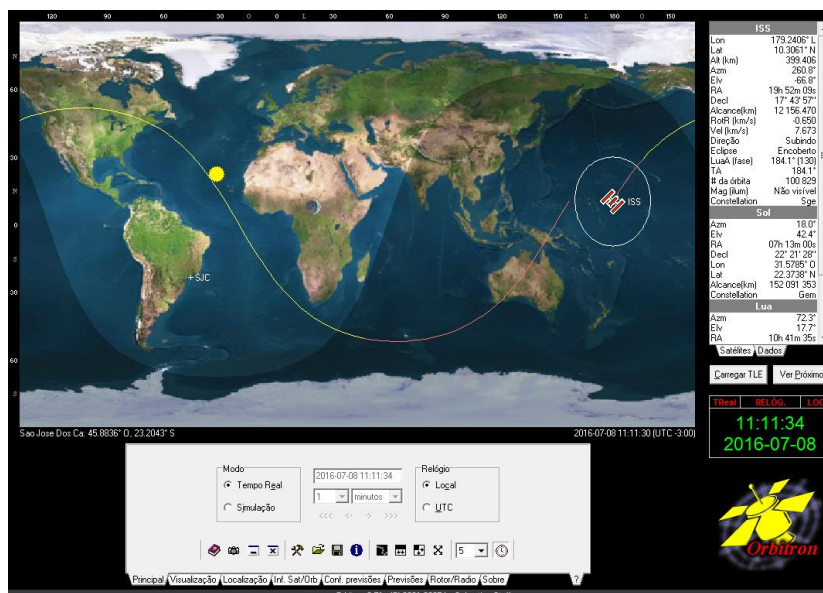
When using Orbitron as graphical user interface, the user is able to connect this software to the rotor controllers of UHF/VHF/S-Band antennas.

Orbitron allows ADSSGS to easily have control of a very large number of satellites with the advantage that it can operate in stand-alone mode, where TLE update can be automatically programmed but it does not require to be online to operate.

Orbitron is a satellite tracking system for radio amateur and observing purposes. It's also used by weather professionals, satellite communication users, astronomers, among others (STOFF, 2016).

This application shows the positions of satellites at any given moment, in real or simulator time, and it can be seen in Figure 3.7. It's free and it's probably one of the easiest and most powerful satellite trackers, according to opinions of thousands of users from all over the world⁶.

Figure 3.7 - Orbitron - Interface



Source: Taken from stoff.pl (2016)

⁶ Available on: <<http://www.stoff.pl/>>. Accessed on: 11 Mar. 2016

The program allows loading Two-line Elements (TLE) from Internet. This files contain information about the orbit of a given satellite or group of satellites, updated constantly.

The position of each satellite can be followed, visualizing them in a World Map or with a Radar View.

This interface can be customized, including the view or not of the sun, the moon and its phases, tails, compass, etc.

Information about each satellite is available, like its name, NORAD number, COSPAR designator, Epoch (UTC and Orbit Number), inclination and more. Orbitron can be updated directly from Celestrak, either manually or automatically⁷.

The position of each satellite at any given time can be predicted. For more than ten years people have been able to track satellites on a computer using satellite tracking software like Orbitron.

Due to predictable conditions of satellite movement in space (lack of atmosphere) computer software can calculate a satellite's position at any given moment. Calculations are done based on known orbit parameters determined at epoch. Having known orbital parameters like inclination, eccentricity, argument of perigee, mean motion (revolutions per day), let us track satellite for a reasonable period of time after epoch. Orbital data for each object is grouped, and distributed as a TLE file.

To keep tracking software working precisely, an elements update should be done periodically. For low orbiting objects (altitude less than 500 km) TLE data should be updated every few days. For higher orbits, TLE can be updated every few weeks. Other important feature consist of making predictions as close to

⁷ Available: < <https://www.celestrak.com/index.asp>>.Accessed on: 11 Mar. 2016

real time as possible by using time synchronization, and the precise coordinates of current location.

It is important to recall that TLE data for an object that has maneuvered since the last time is no longer any good. Updating is most important for satellites like Progress, Soyuz, and the ISS which maneuvers often (docking, deorbit, changing of orbit).

3.2.3. ProEst

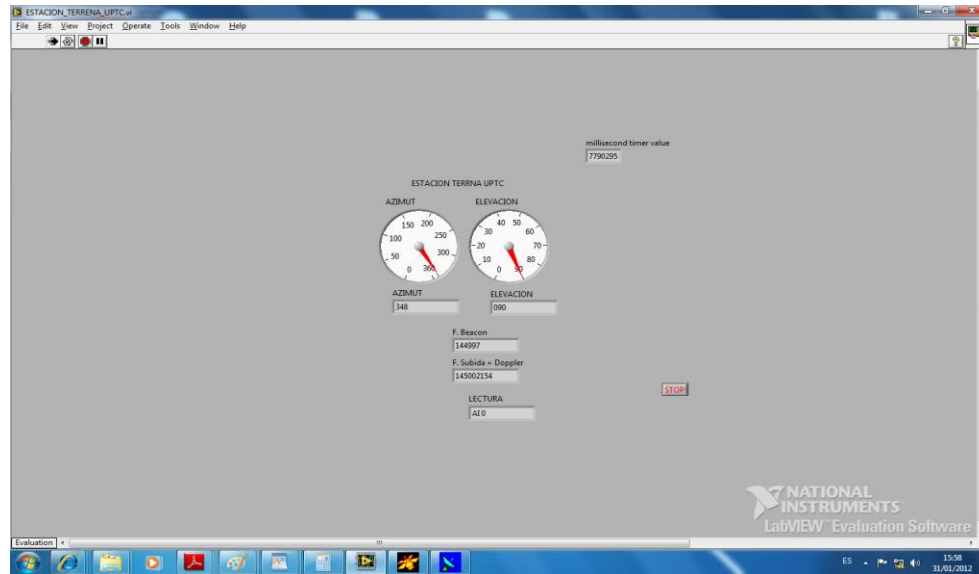
ProEst is a software developed by students Carlos Nonsoque and Julian Molano, as part of their degree work to get the systems engineering degree in the UPTC, Colombia. This work was done under the direction of this thesis' author and aims to move autonomously antennas using for this the TLE supplied by Norad and the graphical interface Orbitron that using drivers can send the angles of azimuth and elevation needed to trace satellite. (NONSOQUE, 2012).

ProEst is an application developed in LabVIEW⁸ to perform the communication process between the satellite and the station automatically. This application is in charge of the control of the radio and the servomotor mechanism of the station through the use of commands, allowing visualizing its state.

For the control of the radio ProEst, is responsible for establishing the serial communication between the radio and the computer, and imports the data of the downlink frequencies with doppler effect shift from the prediction software. Then the serial port is configured to communicate with the radio and the computer; by default setting a serial port, communication speed, flow control, word length and parity. The graphic interface of ProEst allows the visualization of azimuth and elevation values imported to the software of prediction Orbitron (see figure 3.8).

⁸ Available on: <<http://www.ni.com/labview/pt/>>. Accessed on: 1 May. 2016

Figure 3.8 - ProEst Software



Source: Author

The antenna control is done through commercial rotor Yaesu® brand, G-5500 model. This rotor is controlled by a controller model of the same manufacturer Gs 5500, whose function is to control the supply of current to the motor rotor. The rotor control equipment is connected to the computer through the GS 232B interface, allowing the antenna control system to work automatically (ESPINDOLA et al, 2013).

3.2.4. Modem Software

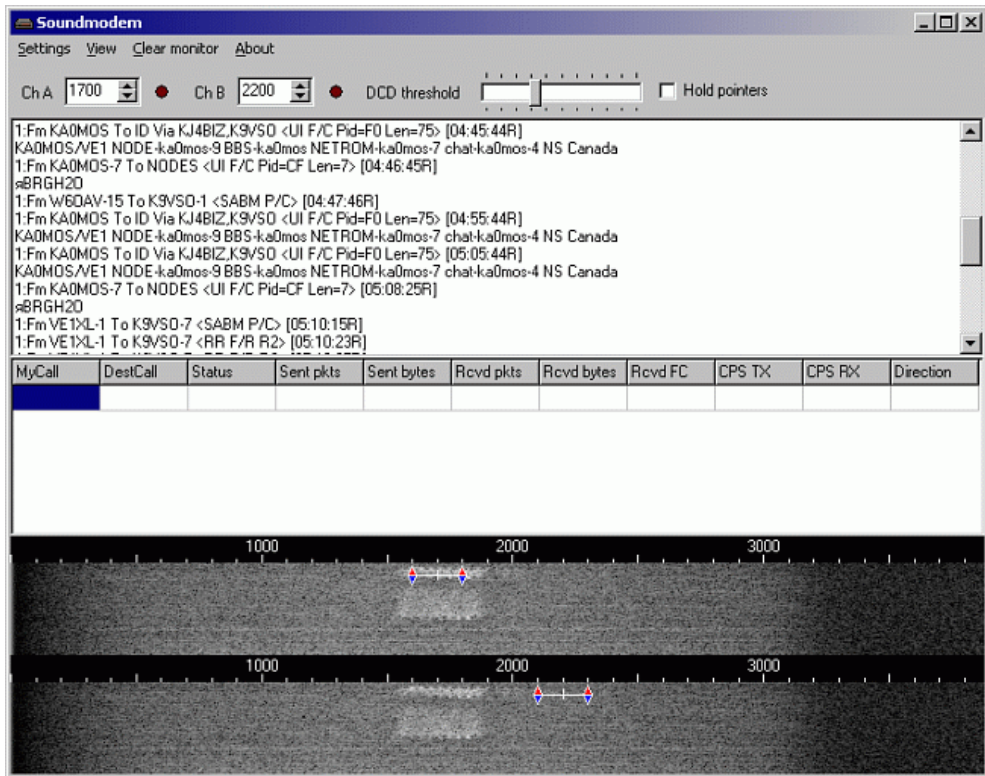
It is known as modem software the module in charge of encoding or decoding received telemetry or commands sent to the satellite. There is a problem here when you want to track a satellite and not have all the software needed to do this task. This is why it is necessary to have the software that each satellite uses to encode or decode its information.

When in the ground station the SDR receives telemetries, they are sent to the modem software, in the case of ADSGS it will be sent to SoundModem⁹, who

⁹ Available on: <<http://uz7.ho.ua/packetradio.htm>>. Accessed on: 28 Feb. 2017

converts the sound into information with an American Standard Code for Information Interchange (ASCII) format.

Figure 3.9 - SoundModem



Source: Take from uz7ho (2017)

SoundModem v.95 is software that converts satellite signals into ASCII encoded information (see figure 3.9). Basically, listen to the signal received by the radio through the computer microphone, in this case an SDR.

When SounModem receives the information encoding in ASCII it sends it to a local or remote machine, if it is remote it does it through the IP address. This process requires the SoundModem to be connected to a telemetry decoder.

Each satellite has its own software to decode engineering value telemetries that are understandable to any user.

In addition, other software is used to convert the SoundModem information into raw values; the telemetry values sent by any satellite are obtained and displayed in gross value. This software is called AGW Online Kiss¹⁰ v. 2.4.4 and is used for this purpose.

3.2.5. Flight Operational Plan - FOP

An automated ground station shall store Flight Operational Plans (FOP) of its satellites in a database. Control antennas allow access to Telemetry, Tracking and Commanding (TT&C) in the VHF/UHF/S-Band frequencies. This is performed during the window time called “pass” when the satellite passes over the ground station. Some ground operations can execute real time procedures, such as sending commands to the satellite and receiving telemetry on the ground station. These procedures are executed in a sequential way, according to a specific FOP.

The main objective for operation planning is the generation of FOP and it is necessary to know precisely the position and the trajectory of the satellite to plan them. Having the previous information, it is possible to calculate the antennas target angles in ground station in both azimuth and elevation.

According to Tominaga (2010), routine planning is obtained from flight dynamics and from the Pass Visibility Prevision (PVP) files. These files contain a sequential list of different passes that satellites will perform and correspond to the TT&C of each satellite for each ground station, within a determined future time interval.

The conflicts generated among different passes are managed with temporal and space overlays. A single ground station cannot perform two tracking simultaneously with the same antenna. On the other hand, a satellite may not establish an uplink simultaneously with other ground stations.

¹⁰ Available in: <<http://www.pe0sat.vgnet.nl/decoding/tlm-decoding-software/dk3wn/>> Access in: 28 Feb. 2017

After managing the potential conflicts generated, the programmed passes are divided into time units in order to perform a planned control. This stage helps in the FOP generation (Tominaga, 2010).

ADSGS uses the FOP to keep information on the location and other variables that each satellite generates updated. The scheduler queries the FOP database each time a specific satellite pass is to occur.

3.3. ADSGS Component

The main components of ADSGS are the User Interface and the Scheduler composed by an ADSGS Agent, which, based on artificial intelligence and using the SATNet database and with the help of the Adapted Hungarian Algorithm allows the ground station to solve the problem of allocation of resources, in this case ground stations and satellites in passing.

The User Interface and the ADSGS Agent are described below.

3.3.1. User Interface

ADSGS has an end user interface responsible for integrating each component needed to provide autonomy and dynamism to the ground station. This interface includes the following software components: Orbitron, Proest, ADSGS Agent, SDR (both hardware and software) and TT&C, which in turn are integrated into the SATNet database, the rules-based expert system and to the FOP.

This interface allows the ADSGS system to have control over the ground station, once it is run on a computer located in the station which in turn is connected to the rotor drivers for the movement of the antennas, to the SDR, as well as SATNet via the internet.

For these connections TCP/IP is used as communication protocol, the user interface can be executed on a computer with minimal configuration characteristics, namely: x86 processor or similar with minimum 1 Gb of RAM

and storage of 500 Mb, must have internet connection and socket available to install SDR.

The User Interface can be developed in Python programming language, this to save compatibility with the SATNet database and the expert system implemented in the expert SINTA as an example.

Next, is described the ADSGS Agent, that is the most significant contribution of this thesis, this algorithm is in charge of operating the ground station minimizing the costs of operation and maximizing the use of hardware and software resources.

3.3.2. Agents

Three different agent categories are presented for the sake of comparison: (1) a Naïve agent, (2) a First Come-First Serve (FCFS) agent using a strategy and finally (3) the proposed ADSGS agent. This last one runs an Adapted Hungarian Algorithm in order to perform its decisions which jointly with the expert system tool provide dynamical and autonomy features.

3.3.2.1. A Naive Agent

According to Salehi-Abari and White (2010), a naïve agent is unable to decide correctly how, when and who interact with. As such, it can't detect and stop the interaction with non-reliable agents due to its inability to adequately evaluate other agents. Naïve agents are optimistic; they consider the other agents as completely reliable and always cooperate with every member of the society. Naïve agents, usually, do not have malicious intents.

A naïve agent uses a simplistic FCFS strategy which allocates resource as they are needed, i.e., when a satellite passes it assigns to the nearest station without taking any further analysis. This strategy simply increases ground stations handovers since the ground station assignment may not actually be compatible to

the satellite under consideration. One can easily see the inefficiency of this strategy.

3.3.2.2. A FCFS Agent

According to Vemon and Manber (1988), the key idea of the First-Come-First-Served (FCFS) algorithm is that allocations are done in the temporal order they take place. A FCFS network agent in this case, differently from its naïve counterpart, performs some previous analysis on the satellite nature before assigning a ground station to it. One can infer that this scheme has a lower number of ground station handovers than a naïve strategy. Nevertheless, this analysis is still shallow and limited as can be seen in the next chapter.

In the chapter 4 the structure of the ADSGS agent is described.

4. THE PROPOSED ADSGS AGENT

This chapter shows the proposed ADSGS agent, its operation and the proposed rules for the system based on Expert SINTA, the suggested modeling in UML for the construction of this agent is presented.

This element plays a key role in the ADSGS proposal since it is responsible for providing the dynamical (react to changes) and autonomy (perform actions unsupervised) features. The author believes that this is one of the major contributions of this thesis work, namely, the assignment problem which is not done in the SATNet autonomously and dynamically.

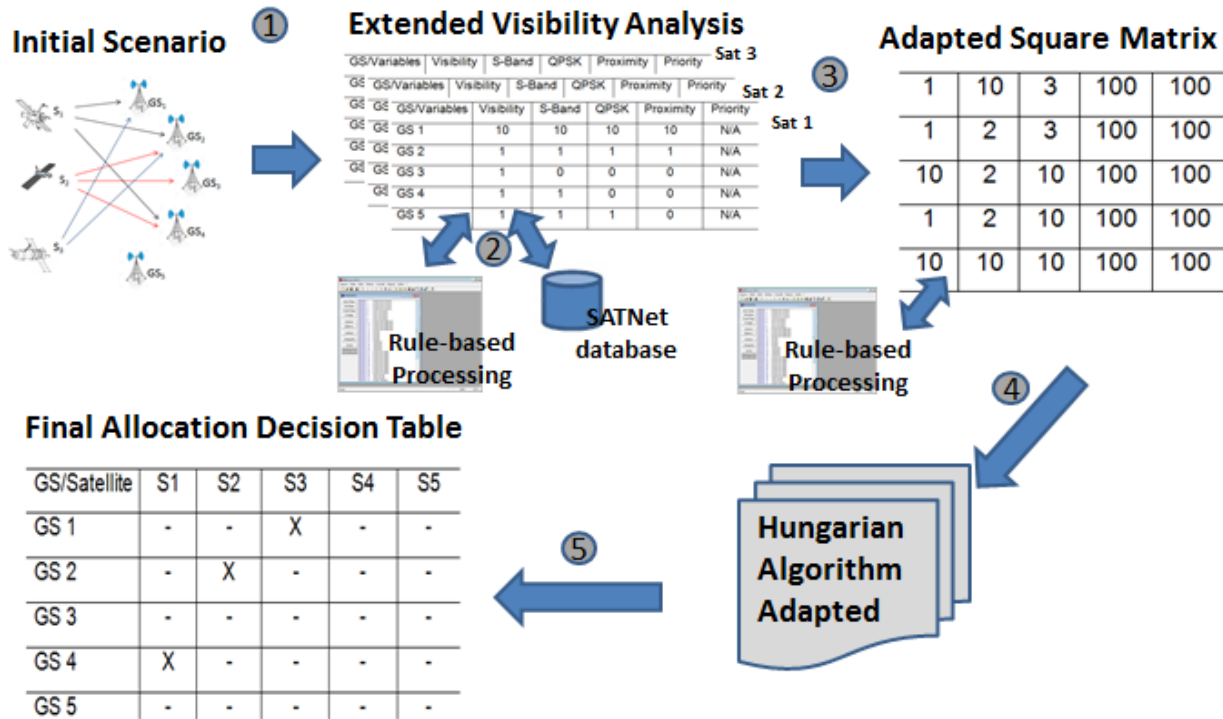
4.1. The ADSGS Agent

According to Russell and Norvig (2004), an agent is something that reasons (agent comes from Latin *agere*, “to do”). But of computer agents are expected to have other attributes that distinguish them from conventional 'programs', such as that they are endowed with autonomous, environmentally aware, persisting, adaptive, and time-consuming controls; are capable of achieving different goals. An agent is one who acts with the intention of achieving the best result or, when there is uncertainty, the best expected result.

The ADSGS agent is responsible for providing to the ground station the autonomy and dynamism required for tracking and control of small satellites, using the hardware and software components already described in this document: the knowledge base managed by Expert SINTA, the database supplied by SATNet, ProEst, SDR-ADSGS, Orbitron and hardware setup.

Figure 4.1 describes the behavior of the ADSGS agent in a scenario where there are 3 satellites and 5 stations, the agent must determine the optimal way of allocating resources, these are matching a satellite to a station.

Figure 4.1 - ADSGS Agent



Source: Author

In order to achieve the optimal allocation, ADSGS agent will follow the next five steps:

1. Initial scenario features three satellites in LEO orbit and stations located around the Earth. A more complete description of this scenario is given in section 5.2 of Chapter 5.
2. Each satellite has its own characteristics defined in the Flight Operational Plan supplied by the SATNet database which in turn, is evaluated by the rule-based expert system created for this purpose. Some of these variables are: visibility from each station, whether it requires S-band for telemetry, type of communication scheme (e.g. QPSK), proximity from the satellite to the ground station and the priority it has.
3. As a result of this analysis, a matrix. This case provides a 3x5 matrix, but running the Adapted Hungarian Algorithm requires a square matrix, so in

order to convert it into a square one an adaptation is made by adding high values (100 for example) in the values that will never be reached or that have no values, such as the absence of satellite 4 and 5. To create this new matrix the agent is based on previous decisions made using the rule-based expert system.

4. The Hungarian Algorithm created in 1955 (Kuhn, 1955), in this thesis it is proposed to make a modification, the adaptation of this algorithm will allow the development of the ADSGS agent part of this proposal. The Adapted Hungarian Algorithm is executed, in Appendix C its execution is found step by step, this algorithm optimizes the resources allocation and allows the ADSGS agent to make the best decision.
5. Finally, a table is obtained in which a satellite is assigned to each ground station, complying with the considerations and requirements of the proposed scenario.

4.1.2. Actors for ADSGS

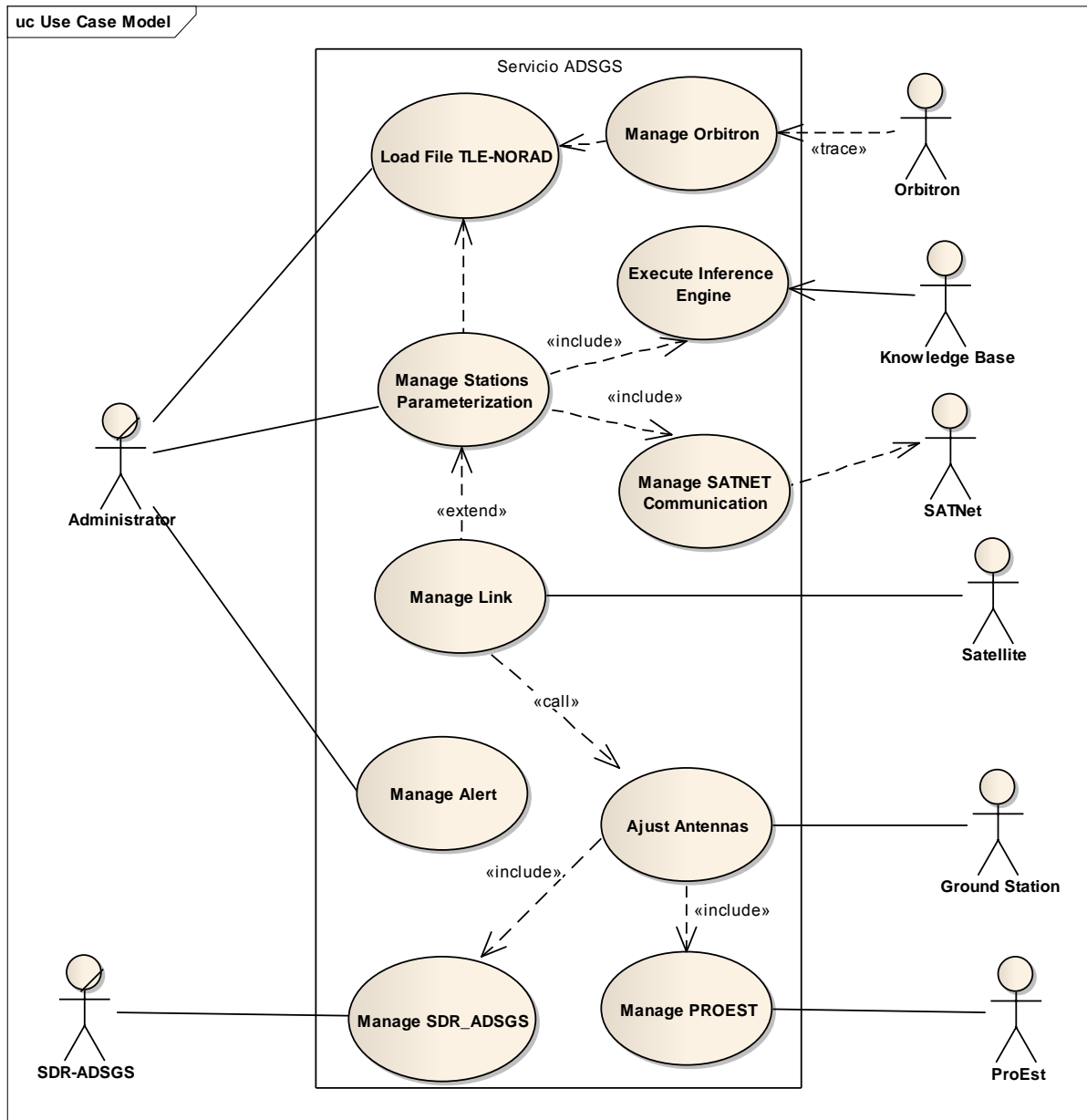
ADSGS features some actors: Administrator, Ground Station, Satellite, Knowledge Base, SATNet, SDR-ADSGS, Orbitron and ProEst. A description of each one is made below:

- Administrator: Is the user with service subscription, it is in charge to manage all the system and the users (create, edit, configure, delete and assign system use privileges). It is also responsible to load the TLE published by NORAD, manage antennas and antennas' parameterization.
- Ground Station: is the element union (hardware, software) that to system behaves as one. It represents the behavior of the ground station regarding to its tracking movement; it sends information to SDR-ADSGS and receives information from ProEst.

- Satellite: is represented by the aircraft in space sending telemetry and receiving telecommands. It manages the link with the ground station.
- Knowledge Base: it is integrated by the knowledge base and the rule-based expert system developed to give autonomy to the ground station. This user runs the inference engine.
- SATNet: is the ground station network which allows that a satellite is connected to the ground station the most time available. It manages the communication with SATNet.
- SDR-ADSGS: allows the connection between the ground station hardware and ADSGS system. It is based on SDR and receives the signal generated by the satellite, to decode it then.
- ProEst: is in charge to manage the automatic movement of antennas in both azimuth and elevation to perform satellite tracking.
- Orbitron: Is in charge to display a visualization of the satellite pass over the ground station. It also gives to ProEst azimuth and elevation values.

Figure 4.2 shows the diagram describing the interaction between the actors and the defined functional requirements.

Figure 4.2 - ADSGS - Actors Diagram



Source: Author

4.1.3. Rule-based system implementation - Expert SINTA

It is part of this proposal the use of artificial intelligence; this allows ADSGS to have autonomy in the control of the ground station. A small study of the most common artificial intelligence techniques is given in Appendix D.

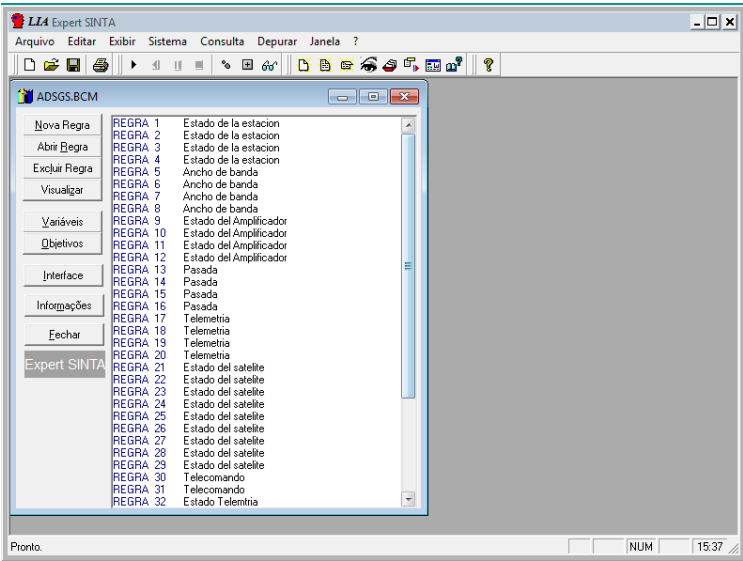
Based on this study it was determined to use a rule-based expert system due to the low cost of its implementation, the processing speed to take a decision at the moment of a pass (it is known that a pass in the best case scenario, is no longer than 10 minutes), the reliability offered by its fast decision-taking and the ease to replicate this system in any other ground station.

In this case will be used Expert SINTA due to its complexity but also due to its user friendly interface.

4.1.3.1. The Shell

Expert SINTA is a software that uses artificial intelligence techniques in order to achieve automated generation of rule-based expert systems. Its main goal is to simplify the work to implement expert systems (SINTA, 1996). Expert SINTA shell was developed at the Computer Science Research Lab of *Universidade Federal do Ceará*; it was implemented upon Borland Delphi 1.0, an object-oriented programming language. SINTA allows developing modular databases through an easy to use interface (SINTA, 1996). Figure 4.3 shows the Expert SINTA main screen.

Figure 4.3 - Expert SINTA - Shell window

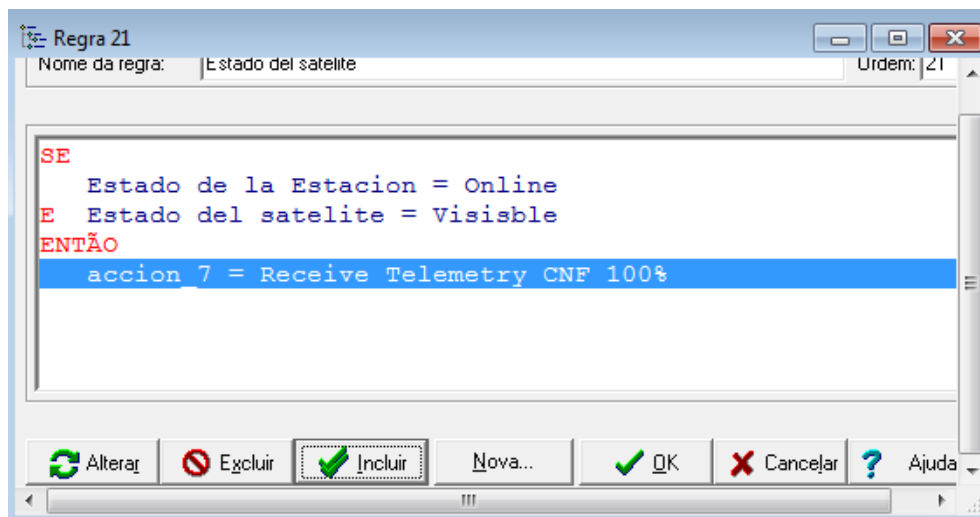


Source: Generated by SINTA

The Shell features a tool bar with the options to edit the knowledge base, a window called “Knowledge-in-a-box” (KIB) with a list of all the present rules at that moment (SINTA, 1996).

Expert SINTA keeps track of knowledge bases generated, in *.BCM files and it is based on production and probabilities rules. These rules are a condition set with the “IF...THEN” structure, allowing the inclusion of logical connectors AND, OR and NOT, to relate the conditions of a rule; it also allows to include the correspondent degree of trust of a certain conclusion. Expert SINTA offers a basic inference engine mainly relying on back chaining. (SINTA, 1996).

Figure 4.4 - Expert SINTA - Production Rule scheme

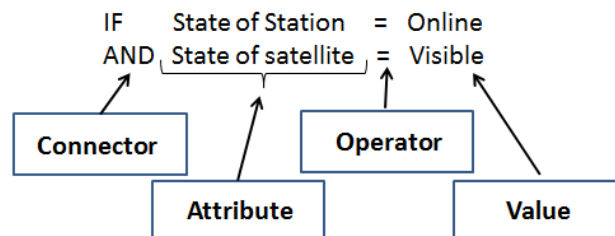


Source: Generated by SINTA

Figure 4.4 shows an example of a production rule. Each rule is composed by premise(s) and conclusion(s). The premises are known as “tail” and the conclusions as “heads”. The set of all tails is known as rule antecedents and the set of all heads as rule consequences (SINTA, 1996). Tails have the following structure:

Connector -> Attribute -> Operator -> Value

Ex:

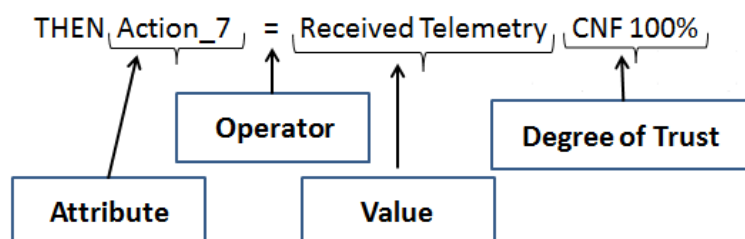


- Connector. Its function is to join the sentences to the premise set. It is formed by logical values AND, OR and NOT.
- Attribute. It is a variable. For instance, state of satellite with predefined values (visible, not visible)
- Operator. It is a link between the attribute and the premise value defining the comparative type. Some operators are: =, >, <=, <>.
- Value. It is an item from a list previously created and related to an attribute.

Heads have the following structure:

Attribute -> = -> Value -> Degree of Trust

Ex:



- Attribute: It is a variable that can assume one or many instances through the query to the knowledge base.
- “=”: It's the operator taking attribute's value.
- Value: It is an item from a list previously created and related to an attribute, it may vary according to the attribute and its type.
- Degree of trust: Is the percentage indicating the certainty level of the specific conclusion of a rule. Varies between 0 and 100%.

To generate the rule-based expert system using Expert SINTA, should take the following steps:

- Set the variables
- Set the objectives (output variables)
- Generate rules
- Set the interfaces (questions that will be asked to the user)

Appendix E shows how Expert SINTA performs each one of these tasks.

4.1.3.2. Questions

As a part of this work a rule-based expert system was implemented using Expert SINTA, the system gives autonomy to a ground station through ADSGS. In Table 4.1 are shown some of the questions/answers that the expert system will use to make decisions, these have no specific order.

Table 4.1 - Questions/Answer

| Questions | Answer |
|---|--|
| The user is registered in SATNet Database? | On / Off |
| Which is the state of the Ground Station now? | Online / Offline |
| Which is the state of the Satellite now? | Visible / No visible |
| Which bandwidth type does the Ground Station accept? | High bandwidth / Low bandwidth |
| What is the state of the Telecommand? | Send Ok / Send Error |
| What is the state of the Telemetry? | Receive Ok / Receive Error |
| The Ground Station is registered in SATNet database? | Satnet On / Satnet Off |
| The next passed is? | Pass \leq 10 minutes / |
| What Satellite is passing now? | Satellite A in passed / Satellite B in passed |
| Which bandwidth type is required for the Flight Operation Plan? | High bandwidth / Low bandwidth |
| What's the state of the Power Amplifier? | On_Line / Off_Line |
| What's the priority of the Satellite? | Priority Satellite A \geq Priority Satellite B / Priority Satellite A $<$ Priority Satellite B |
| Which was the telemetry accepted? | Tm15 $>$ 10v / Tm15 $<$ 10v |
| Which was the telemetry accepted? | Tm5 $>$ 10a / Tm 5 $<$ 10a |

Source: Author

Expert SINTA allows exporting the knowledge base; this system contains the required libraries to this type of exportation, allowing AD SGS to use the created knowledge base and thus give autonomy to the ground station.

As a summary of this section, it can be said that AD SGS uses automation in two moments: the first moment when through ProEst sends the azimuth and elevation angles to the antenna system for the servomotor to autonomously move said antennas. A second moment occurs when with the use of artificial intelligence the system makes the necessary decisions to receive the telemetry

sent by the satellite that is being tracked at that moment, regardless of which ground station will receive it, the above coordinated with SATNet.

The decisions that ADSGS takes are based on the knowledge of an expert who feeds the knowledge base, this allows decisions made in time to optimize the control of the ground station.

ADSGS through the expert SINTA allows adding more rules; this must be done with the help of an expert.

4.1.3.3. Rules for Expert SINTA

Below are all the rules created for the Ruled-based System along with the tables that allows it to work.

Rule1 = If visible station is offline => Migrate FOP for next online station

Station A offline => Migrate all FOP to station B

Table 4.2 - Rule 1

| Ground Station | Satellite | Action |
|----------------|-------------|--------------------|
| On-line | Visible | Accept Rx |
| On-line | Not visible | Wait for next pass |
| Off-line | Visible | Migrate to GS B |
| Off-line | Not visible | Wait for next pass |

Source: Author

Rule2 = If FOP demands larger band=> Migrate PV for next station with larger capacity.

{FOP says to download images}, {Station A and station B are visible}, {Station B with more antenna gain} => Configure pipeline RX to Station B.

Table 4.3 - Rule 2

| Ground Station | Flight Operational Plan | Action |
|--------------------------|-------------------------|-----------------|
| GS A Accepts larger band | Larger Band | Accept Rx |
| GS A Accept larger band | Lower Band | Accept Rx |
| GS A Accepts lower band | Larger Band | Migrate to GS B |
| GS A Accept lower band | Lower Band | Accept Rx |

Source: Author

Rule3 = If power amplifier from Station A fails => Migrate transmission to Station B

Station A - PA fails => Migrate TX to Station B

Table 4.4 - Rule 3

| Ground Station | Power amplifier | Action |
|----------------|-----------------|--------------------|
| On-Line | On-Line | Accept Rx |
| On-Line | Off-Line | Migrate to GS B |
| Off-Line | On-Line | Wait for next pass |
| Off-Line | Off-Line | Wait for next pass |

Source: Author

Rule4 = If there is a 10 min window to next satellite pass => Generate an audible alarm.

Table 4.5 - Rule 4

| Ground Station | Pass | Action |
|----------------|---------------------------|---|
| On-Line | Time to Pass \leq 10m. | Generate an audible alarm or text message |
| On-Line | Time to Pass $>$ 10 m. | Accept Rx |
| Off-Line | Time to Pass \leq 10 m. | Standby |
| Off-Line | Time to Pass $>$ 10 m. | Standby |

Source: Author

Rule5 = If telemetry 5 > 10a => Send telecommand 120

Table 4.6 - Rule 5

| Rx | Telemetry | Action |
|-------------|-----------|-------------|
| If Rx = Tm5 | Tm5 > 10a | Send Tc120 |
| If Rx = Tm5 | Tm5 < 10a | Send Tc 125 |

Source: Author

Rule6 = If telemetry 15 > 10v => Send telecommand 130

Table 4.7 - Rule 6

| Rx | Telemetry | Action |
|--------------|------------|------------|
| If Rx = Tm15 | Tm15 > 10v | Send Tc130 |
| If Rx = Tm15 | Tm15 < 10v | Send Tc135 |

Source: Author

Rule7 = If connection with satellite is achieved => Receive telemetry

Table 4.8 - Rule 7

| Ground Station | Satellite | Action |
|----------------|-----------|--------------------|
| On-Line | On-Line | Receive Telemetry |
| On-Line | Off-Line | Wait for next pass |
| Off-Line | On-Line | Wait for next pass |
| Off-Line | Off-Line | Wait for next pass |

Source: Author

Rule8 = If connection with satellite is achieved => Send telecommand

Table 4.9 - Rule 8

| Ground Station | Satellite | Action |
|----------------|-----------|--------------------|
| On-Line | On-Line | Send Telecommand |
| On-Line | Off-Line | Wait for next pass |
| Off-Line | On-Line | Wait for next pass |
| Off-Line | Off-Line | Wait for next pass |

Source: Author

Rule9 = If telecommand was correctly sent = > Register in database

Table 4.10 - Rule 9

| Ground Station | Satellite | Action |
|----------------|-----------|----------------------|
| Tc send | Tc =ok | Register DB = ok |
| Tc send | Tc =no ok | Send message to user |

Source: Author

Rule10 = If telemetry was correctly received = > Register in database

Table 4.11 - Rule 10

| Ground Station | Satellite | Action |
|----------------|------------|----------------------|
| Tm send | Tm = ok | Register DB = ok |
| Tm send | Tm = no ok | Send message to user |

Source: Author

Rule11 = If the satellite is out of range => Send a message to user interface

Table 4.12 - Rule 11

| Ground Station | Satellite | Action |
|----------------|-----------|----------------------|
| On-Line | Out range | Send message to user |
| On-Line | In range | Accept Rx |

Source: Author

Rule12 = if two satellites pass simultaneously => Track satellite with higher priority

Table 4.13 - Rule 12

| Ground Station | Satellite | Action |
|---------------------|------------------------------|-----------------------|
| Satellite A in pass | Priority A \geq Priority B | Accept Rx satellite A |
| Satellite A in pass | Priority A < Priority B | Accept Rx satellite B |
| Satellite B in pass | Priority B \geq Priority A | Accept Rx satellite B |
| Satellite B in pass | Priority B < Priority A | Accept Rx satellite A |

Source: Author

Rule13 = If registered user in Satnet database => Allow access to network

Table 4.14 - Rule 13

| Ground Station | Satnet Database | Action |
|----------------|-------------------|------------------|
| Request access | User Satnet = on | Access = allowed |
| Request access | User Satnet = off | Access = denied |

Source: Author

Rule14 = If station is compatible with Satnet => Allow access to network

Table 4.15 - Rule 14

| Ground Station | Satnet Database | Action |
|----------------|--------------------------|------------------|
| Compatible | GS register Satnet = on | Access = allowed |
| Compatible | GS register Satnet = off | Access = denied |

Source: Author

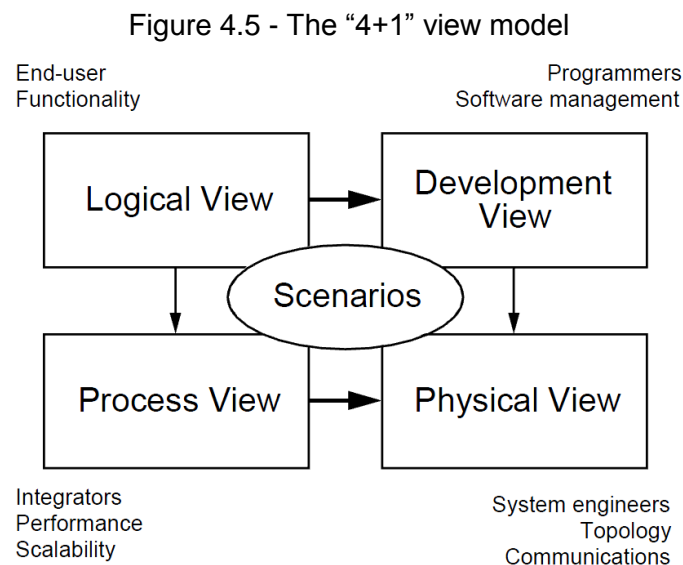
4.2. The ADSGS UML Modeling

Since the implementation of the ADSGS is out of the scope of this work, its conceptualization is expressed via its “4+1” view UML modeling. This view model of software architecture was proposed by Kruchten (1995), and it

consists of "describing the architecture of software-intensive systems, based on the use of multiple, concurrent views". The "4+1" is a view model which fits with the IEEE 1471-2000 standard (IEEE, 2016).

The views are used to describe the system from different point of view of parts involved like final users, developers and project managers. The four views of the model are: logical view, process view, development view and physical view. The use cases or scenarios are used to illustrate the architecture serving as the 'plus one' view. That's why the model contains 4+1 views (KRUCHTEN, 1995).

The "4+1" view model has a high degree of importance since it is closely related with all the parts involved according to its role inside the project development. Figure 4.5 shows the different views of this model.

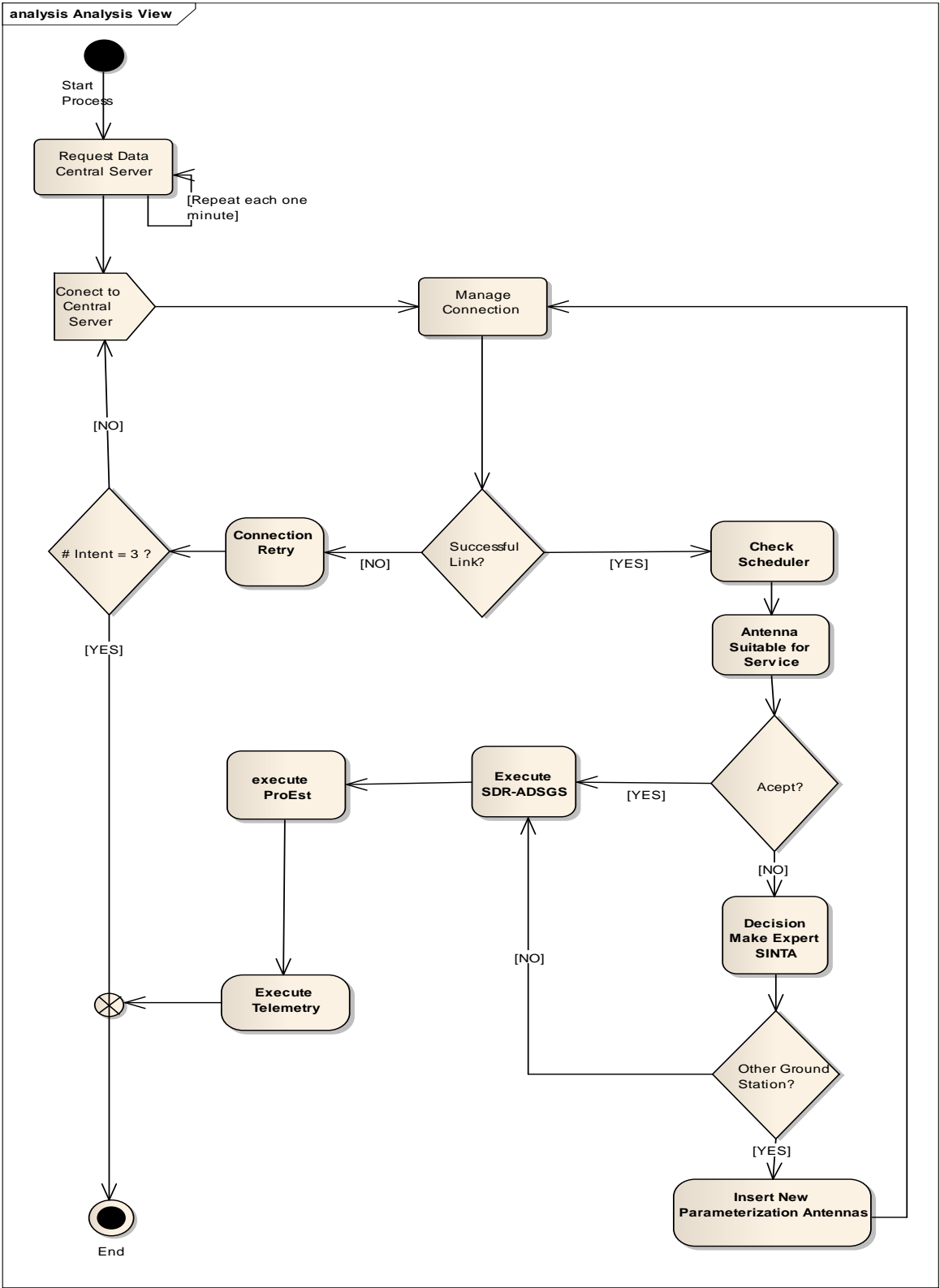


Source: Kruchten (1995)

4.2.1. Detailed Workflow of the ADSGS Agent

In the following section it is described the main sequence of ADSGS system. In the figure 4.6 show how ADSGS work.

Figure 4.6 - ADSGS - Flow Diagram



Source: Author

As a prerequisite the administrator must have stored and parameterized the following items:

- Parameterization of actions to perform at the antenna
- Parameterization of the expert system (Expert SINTA)
- Load of TLE file - NORAD:

The main flow of the system, Communication with Central System starts at the actor “Ground Station” and performs the following steps

1. There is a scheduled task that every X time performs a Communication with Central system.
 - This function is in charge to check and collect the following information:
 - Current location of the satellite scheduled to be tracked.
 - Current state variables of the antennas.
 - If ProEst is working
 - If SDR-ADSGS is working
 - If the last transaction was successful
 - If there are errors that the admin should know
 - If there is an internet connection.
 - Some other required variables to the expert system
 - If the system could collect the information, then get to the step 2, else it gets to 3.
 - The system keeps a record of the error that did not allow to review and collect the information. It also ends the process.
2. The system with the information collected in step 1 proceeds to communicate with Central System.

- If transaction is successful, it continues to step 3. Else it keeps a record of the error.
3. Central System receives the communication from station Y, based on its identifier and start to perform:
 - Check the parameterization configuration of the antenna.
 - Check which satellite must communicate with according to parameterization.
 - Get basic data from the satellite with assistance of Orbitron.
 4. The Central System after collecting the required information establishes a communication with the expert system.
 5. The expert system starts to iterate upon each one of the identified variables, sending one to one to the expert system where:
 - a) The variable is checked
 - b) It is sent to the knowledge database making a question previously associated with that variable
 - c) The knowledge database provides an answer
 - d) The expert system verifies if it requires to know other variable to take the decision. If so, it requests the Central System to provide that variable. Else, continues with e)
 - e) The expert system sends the answer to the Central System.
 - f) The Central System can evaluate three possible alternatives:
 - Send the information to perform at the antenna, to activate the tracking process. Then continues with step 6.

- Send an alert to SATNet, to collect information in other point.
 - Add a new parameterization task, to another or to the same antennas requesting it.
6. The system returns a communication to the antenna with the process to be performed.
 7. The ground station antenna receives notification and parameterization from Central System.
 8. The antennas sends new parameterization to the inner database in Derby, where it considers if it was already executed, if it should be updated or simply if it is already created and there is not necessary to add more information.
 9. The ground station antenna launches a process where verifies which is the next parameterization to be executed, considering priority, time and other variables the expert system had already evaluated.
 10. The ground station antenna activates the external software in the following order:
 - SDR-ADSGS: it is responsible for tuning if the radio frequency to downlink and for receiving telemetry.
 - ProEst: it is responsible for performing the movement of the antennas rotors in both azimuth and elevation.
 11. The Central System wait for the satellite pass and collects information of telemetry.
 12. It sends the information to the programmed place (SATNet) and informs that transaction was done.

4.2.2. Databases of the Project

Databases used in ADSGS development:

- Knowledge base (developed upon Expert SINTA). It contains libraries making it compatible to Java programming.
- SATNet database. It was developed in MySQL upon Django, an open source database manager which is totally compatible with Python programming.
- Central Server Database. It is included in the Flight Operational Plan of the satellites added to SATNet.
- Ground Station Database. This database is made in MySQL and it contains the information about the all Ground Station, Scheduler Station, Tracking and Satellite.

4.2.3. ADSGS Suggested Technologies

Here is a list of suggested software elements for a possible development of ADSGS, it should be clarified that this system has not been developed:

- Java SE 8. Open Source object-oriented programming language.
- Python 2 Release - 2.7.12 Open Source object-oriented programming language.
- MySQL upon Django.
- Derby Database. It is based upon Java and embed in Java 8 specification.
- Java EE7. Java specification oriented to Enterprise applications, featuring important benefits in security management, scalability, transactions among other robust systems requirements.

- Maven 3.0.4. Technology able to automate the process of packaging and creation of Java executable.
- IDE Eclipse JEE Kepler. Java Development Environment.
- Wildfly 10. Web application server, it is specialized on the execution of JEE7 applications.
- GIT, version control software.

Appendix F described of the different diagrams suggested by Kontio (2008), as well as the functional and non-functional requirements, and the suggested technologies for ADSGS implementation.

5. CASE STUDY AND RESULTS

In this chapter a provisional illustrative application for the allocation scheme of the ADSGS agent is exemplified and a comparison is done to a naive agent and a FCFS agent. A simple STK with MatLab simulation is presented as a motivation for the ADSGS agent needs. The proposed approach uses the Adapted Hungarian Algorithm Adapted in two study cases and the obtained results infers the efficiency of the strategy suggested.

In order to illustrate the ADSGS agent operation, two allocation scenarios are considered:

- **1-to-N Allocation:** one satellite is assigned to a set of ground stations
- **N-to-N Allocation:** a set of satellites is assigned to a set of ground stations

5.1. Scenario #1. 1 Satellite to N Ground Stations

The scenario adopted here has one satellite and prospective five ground stations. As the cost of ground stations hardware is dropping this scenario may be valid especially for the category of smallsats. The ADSGS agent will take the decisions required to allocate appropriate stations and take actions in case of unexpected events, such as a primary ground station that become unavailable.

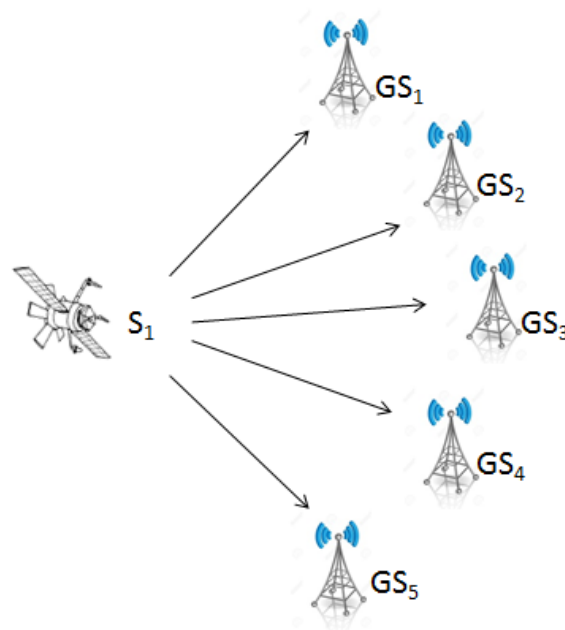
This scenario will consider a satellite context **C** under the following conditions:

- Satellite Name: X
- Mission: Optical Remote Sensing
- Frequency band: S-Band
- Modulation: QPSK at 9600 bps
- Battery Depth of Discharge (DOD): 40%

- Onboard Image Recorder Occupancy: 80%
- Sunlight State: Eclipse
- Proximity to a Ground Station: High

The distribution for this scenario #1 is shown in Figure 5.1. It considers one sole satellite in orbit which may pass over possible 5 ground stations distributed around the world. The connection between them is done through SATNet and ADSSGS agent is used to optimize the resource usage and to obtain the required telemetry.

Figure 5.1 - Scenario #1 - 1 Satellite to n Ground Stations



Source: Author

For running a rule-based reasoning scheme for decision's taking, it is necessary a list of "facts" **F** concerning the Ground Stations' status towards satellite X:

- Station 1 is unavailable at this moment.

- Station 2 supports S-Band, is able to process QPSK, and is the closest station.
- Stations 2, 3, 4 and 5 are visible.
- Station 3 is visible but with SDR malfunctioning.
- One station is assigned to communicate primarily and another station as standby for backup.
- Station 4 supports S-band.
- Station 5 supports S-Band and able to process QPSK.
- Station 5 is reserved for backup.

Taking into account this scenario composed of a satellite context **C** and the list of facts **F** just mentioned, one can assemble an “*extended-visibility*” matrix **M** as seen in Table 5.1 which shows the used variables according to this convention: (a) **1** = station meets requirements; (b) **0** = station does not meet requirements; (c) **10** = station does not apply at all.

Table 5.1 - Extended-visibility matrix for: *1-satellite to n-ground stations*

| GS\Variables | Visibility | S-Band | QPSK | Proximity | Priority |
|--------------|------------|--------|------|-----------|----------|
| GS1 | 10 | 10 | 10 | 10 | N/A |
| GS2 | 1 | 1 | 1 | 1 | N/A |
| GS3 | 1 | 0 | 0 | 0 | N/A |
| GS4 | 1 | 1 | 0 | 0 | N/A |
| GS5 | 1 | 1 | 1 | 0 | N/A |

Source: Author

In this case ADSSGS agent will evaluate the list of facts against the satellite context using an expert system developed for this purpose and database

information from the SATNet. The knowledge base is considered a set of rules **R** required to take decisions.

An FCFS agent in this case with a proper set of rules will arrive to the same results from a ADSSGS agent does. A naïve agent conversely would assign simply assign satellites following proximity criteria solely.

5.2. Scenario #2. N-Satellites and M-Ground Stations

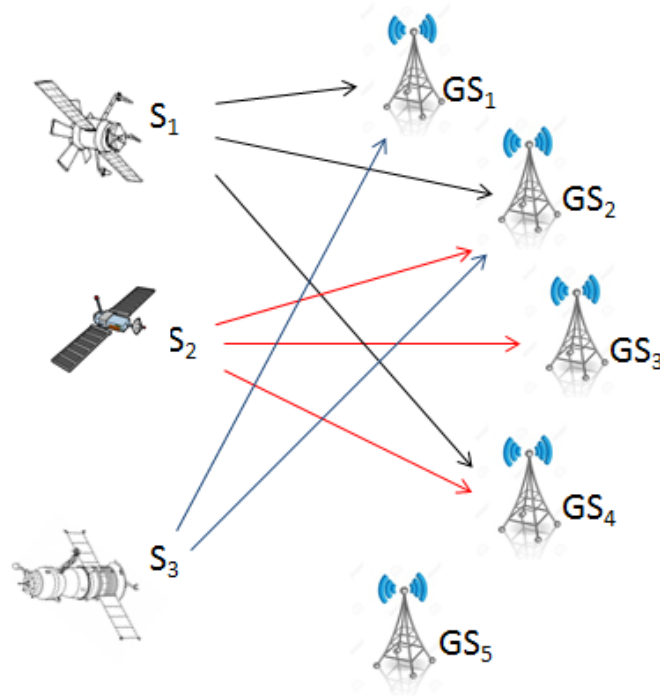
This scenario under consideration uses the same scenario presented when using the STK simulation of 3 satellites and 5 ground stations. Therefore three satellites, supposedly satellite 1 (NOAA_19) is used for weather prediction, satellite 2 (CBERS_4) for scientific use and satellite 3 (IRIDIUM_6) used for communications and five ground stations: SJC_GS, CALPOLY_GS, UPTC_GS, UVIGO_GS, OKINAWA_GS.

Here are presented the list of Facts, **F**:

- Satellites are used to take pictures, meteorology and communications respectively.
- Stations 1, 2 and 4 receive images and have a greater bandwidth.
- Stations 2, 3 and 4 have a greater bandwidth.
- Satellites that allows downloading images have a high priority
- Satellites that are able to store big amounts for weather prediction have a medium priority
- Satellites that can provide communications links with minor delay have a low priority.
- Station 5 set as backup in case of any failure.

The visibility distributions of $n=3$ satellites with $m=5$ ground stations are show in Figure 5.2

Figure 5.2 - Scenario #2 - n Satellite to m Ground Stations



Source: Author

Based on the visibility from Figure 5.2 and the list of facts F determined for this scenario one arrives to Table 5.2. The table is than transformed to a 5x5 matrix since the allocation uses the Adapted Hungarian Algorithm for plausible results.

Table 5.2 - Preferences to a n-satellite m-ground stations

| GS/Satellite | S1 | S2 | S3 | S4 | S5 |
|--------------|----|----|----|----|----|
| GS 1 | 1 | - | 3 | - | - |
| GS 2 | 1 | 2 | 3 | - | - |
| GS 3 | - | 2 | - | - | - |
| GS 4 | 1 | 2 | - | - | - |
| GS 5 | - | - | - | - | - |

Source: Author

Using an adapted version of the 5x5 matrix, the Adapted Hungarian Algorithm is run and the final allocation obtained in the Table 5.3. In Appendix C this algorithm is shown step by step.

Table 5.3 - Allocation result from the Adapted Hungarian Algorithm

| GS/Satellite | S1 | S2 | S3 | S4 | S5 |
|--------------|----|----|----|----|----|
| GS 1 | - | - | X | - | - |
| GS 2 | - | X | - | - | - |
| GS 3 | - | - | - | - | - |
| GS 4 | X | - | - | - | - |
| GS 5 | - | - | - | - | - |

Source: Author

The optimal value is satellite 1 to be assigned to station 4, satellite 2 to station 2 and satellite 3 assigned to station 1, the ground station 5 may be used as a backup resource just in case of service disruption. In the advent of any service disruptions the ADSGS agent will dynamically react and reassign new mappings of satellites to ground stations autonomously.

5.3. A Brief Simulation Scenario using STK with MATLAB

A simulation using STK¹¹ with MATLAB¹² is presented here. The STK is a physics-based software package which has been developed since 1989 as a commercial off the shelf software tool. STK is a 2D and 3D modeling environment used by engineers, mission analysts, operators, and decision-makers to model complex systems such as aircraft, satellites, ground vehicles, and their sensors to evaluate their performance in real or simulated time. MatLab is a language program, is in automobile active safety systems, interplanetary spacecraft, health monitoring devices, smart power grids, and LTE cellular networks. It is used for machine learning, signal processing, image

¹¹ Available on: < <https://www.agi.com/products/stk/>> Accessed on: 10 Nov. 2016.

¹² Available on: <https://www.mathworks.com/products/matlab.html> Accessed on: 12 Jun. 2017.

processing, computer vision, communications, computational finance, control design, robotics, and much more.

This scenario considers again the case of 3 satellites and 6 ground stations. Each satellite has its own mission: satellite 1 (NOAA_19) is used for weather prediction, satellite 2 (CBERS_4) for scientific use and satellite 3 (IRIDIUM_6) used for communications. In the other hand, six ground stations are distributed around the world. Table 5.4 shows the selected stations to develop this scenario which scenario is reused later on.

Table 5.4 - Ground Stations

| Ground Station | City | Localization |
|----------------|-----------------------------------|--|
| SJC_GS | São Jose dos Campos SP, Brazil | Latitude: -23.2106° Longitude: -45.8756° Altitude: 604m. |
| CALPOLY_GS | San Luis Obispo CA, USA | Latitude: 35.3042° Longitude: -120.6631° Altitude: 96m. |
| UPTC_GS | Tunja BOY, Colombia | Latitude: 5.5519° Longitude: -73.3566° Altitude: 2697m. |
| UVIGO_GS | Vigo Spain | Latitude: 42.1702° Longitude: -8.6837° Altitude: 463m. |
| OKINAWA_GS | Okinawa Japan | Latitude: 26.4998° Longitude: 127.8963° Altitude: 65m. |
| BGU_GS | Beer-Sheva Israel | Latitude: 32.1647° Longitude: 34.8263° Altitude: 509m. |

Source: Author

Table 5.5 shows the selected satellites for this scenario, with LEO orbit and different missions.

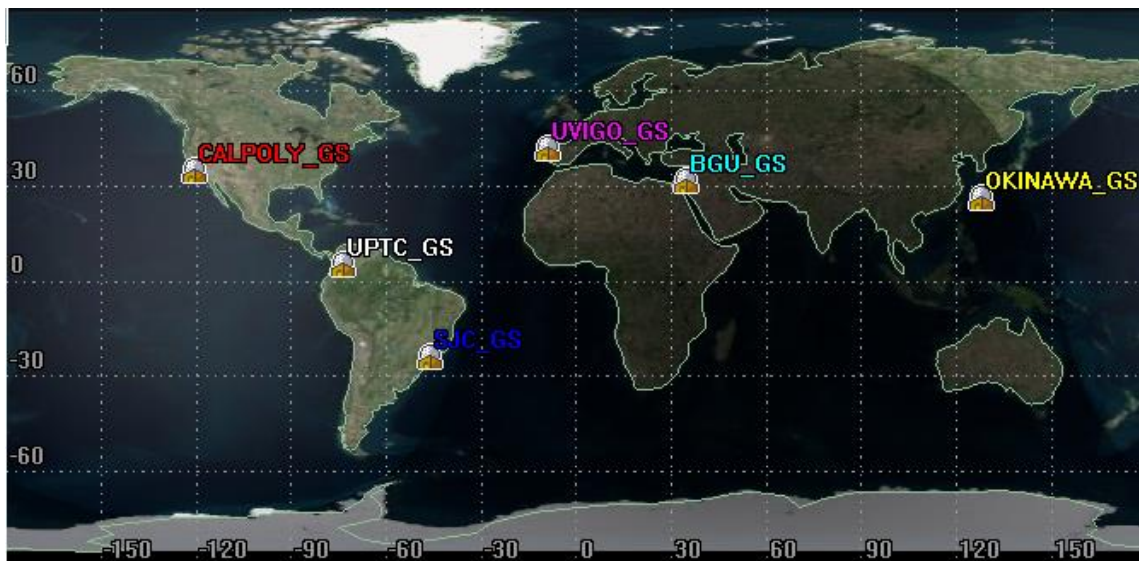
Table 5.5 - Satellites

| Satellite | Mission | Data of launch | Own |
|-------------------|----------------|----------------|---------------|
| NOAA_19 - 33591 | Weather | 06/02/2009 | United States |
| CBERS_4 - 40336 | Scientific | 2014/12/07 | China/Brazil |
| IRIDIUM_6 - 24794 | Communications | 05/05/1997 | United States |

Source: Author

The intention was to simulate 1 day starting May 1th 2017 at 00:00 hours to May 2th 2017 at 00:00 hours. In this time lapse, four conflicts occurred, showing the real problem existing in satellite tracking from a station. Figure 5.3 shows mentioned ground stations and satellites and they real location and orbital tracks.

Figure 5.3 - Illustrative simulation scenario using STK



Source: Author

The simulation using STK in 2D and 3D is equivalent to 24 hours starting May 1th 2017 at 12:00 and ending May 2th 2017 at 12:00; this simulation lasts 2 minutes (120 seconds) which means every 5 seconds under simulation equals one hour in real life.

Considering the limited number of stations, potential access conflicts may occur hence they are identified in Table 5.6. This happens when a single ground station sees two or more satellites simultaneously and has to decide which satellite it will track. Here the simulation performed in MatLab runs the ADSGS Agent Algorithm. This is responsible for allocating the satellite autonomously that will be attended to and the next step to follow, in the same way it will be ready for the next pass, giving dynamism to the system.

For the above, the ADSGS Agent uses the Adapted Hungarian Algorithm, the Knowledge Base and the SATNet Base as shown in figure 4.1 of section 4.1

This is the moment where an agent may come into play and arbitrates autonomously which object to track according to its allocation scheme. In this scenario for example, allocation is based onto an access priority scheme where a meteorological satellite has a higher priority than a communications satellite but a lower one compared to a remote sensing satellite presumably. A rule-based agent can also assign a next ground station available to redeem those satellites which were not designated in a first moment as shown the last column of Table 5.6. In this table to show some conflict occurred in this simulation.

Table 5.6 - Satellite access conflicts identified during the STK simulation

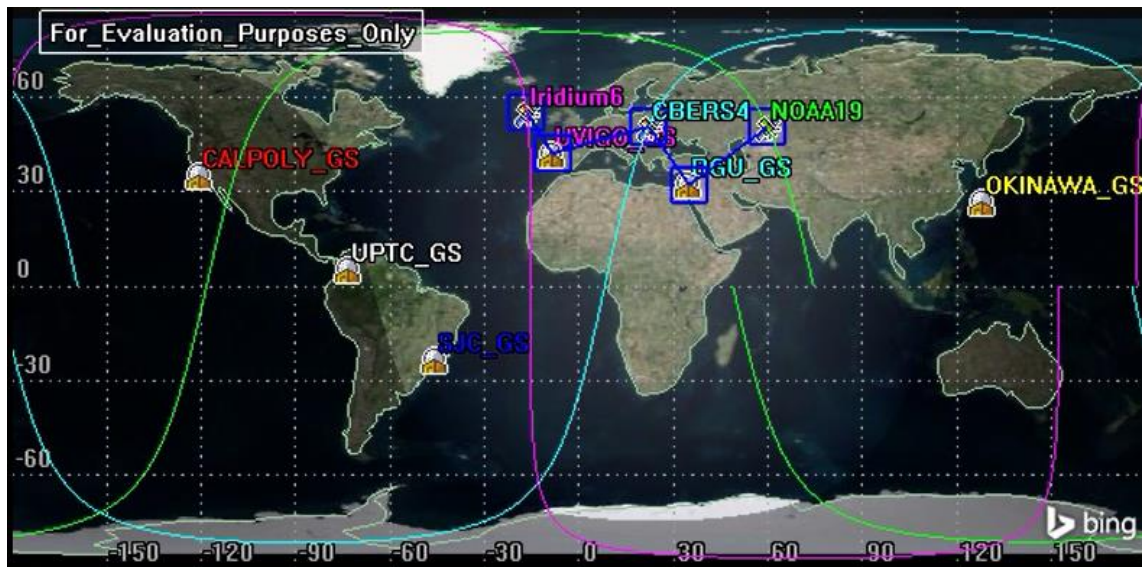
| Conflict # | Time (2 m) | Ground Station | Satellite 1 | Satellite 2 | ADSGS Agent | Later Decision |
|------------|------------|----------------|-------------|-------------|-------------|----------------------|
| 1 | 0.04 | SJC_GS | IRIDIUM6 | CBERS4 | IRIDIUM6 | CBERS4 at OKINAWA_GS |
| 2 | 0.20 | UPTC_GS | IRIDIUM6 | CBERS4 | IRIDIUM6 | CBERS4 at CALPOLY_GS |
| 3 | 0.33 | BGU_GS | CBERS4 | IRIDIUM6 | CBERS4 | IRIDIUM6 at BGU_GS* |
| 4 | 1.51 | UVIGO_GS | IRIDIUM6 | CBERS4 | IRIDIUM6 | CBERS4 at BGU_GS |
| 5 | 1.51 | BGU_GS | CBERS4 | NOAA19 | NOAA19 | CBERS4 at UVIGO_GS |

*One cycle after

Source: Author

Finally, a fourth and five conflict (#4 and #5) is shown in Figure 5.5 at time=1.51s between satellites IRIDIUM6 and CBERS4 over the ground station located in Vigo - Spain at the Vigo University. Therefore, the ADSGS agent assigns the IRIDIUM6 satellite to the UVIGO_GS station and it updates the CBERS4 FOP so that the BGU_GS station connects to it later for operations procedures. At the same time the agent decides to attend the satellite CBERS4 in the station BGU_GS and sends the FOP of the satellite CBERS4 to the station OKINAWA_GS.

Figure 5.5 - Satellite access conflict #4 and #5



Source: Author

With this simulation, it illustrates how the proposed autonomous ADSGS agent acts and it allows further usage optimization of available resources when performing control and tracking of satellites. This contributes to reducing the operational cost of a ground station.

6. CONCLUSIONS AND FUTURE WORK

Here are described the conclusions and final considerations of the developments performed so far along with suggested future works. Sections 6.1 content the conclusions and final considerations, sections 6.2 corresponding to list of publications attained (conferences, abstracts and papers) and the section 6.3 content de envisaged future works, as follows.

6.1. Conclusions and final considerations

The purpose of this thesis was to propose an autonomous and dynamic approach to small satellite operation. In order to achieve this goal, It was proposed a middleware using COTS elements hardware for ground station and interface to the SATNet ground station network.

The proposed approach was named ADSGS, a system relying on a rule-based expert system, providing autonomy and dynamism to the ground station and in integration to the SATNet. The use of this solution can reduce operation costs of a station allowing the optimization of hardware and software resources and small satellites' control and track processes.

After the revision of different artificial intelligence techniques, it was chosen a rule-based expert system with a knowledge base, due to its low cost application, the reliability offered by its high speed decision making during a pass over the station and the ease to replicate in any other ground station.

The use of a rule-based expert system provides autonomy to ADSGS at handling decisions the system needs to perform in a specific situation, thus facilitating ground station control.

All ADSGS software components integrated to SATNet network allows to a ground station to respond immediately to different assigned missions, providing dynamism to the station.

This thesis proposes the use of SDR hardware as a solution to replace conventional radios favors widely the costs of a ground station and in turn, allows using a wider frequency range in VHF, UHF and S-Band.

SDR software allows changing the frequencies of the antennas dynamically, this allows ADSGS to offer dynamism to ground station, being understood as the ability to change frequencies according to the satellite that is to be traced at that time. SDR also expands possibilities of use since it is an open code project allowing to program new applications and uses.

The use of Expert SINTA to develop the rule-based expert system provided agility to this proposal and in turn, allowed to create an agile system with an easy user interface but with a robust inference engine responding to the needs of ADSGS.

By creating a hardware configuration to the ground station based on the experience of the radio amateur community, it provides solidity and agility in the satellite-station communication and in turn, with all different available stations deployed around the world.

ADSGS could be a standard to other ground stations dedicated to control and tracking of small satellites as it covers all communication modules required to meet the needs of a satellite link (telemetry, telecommand, tracking, coding / decoding).

Among the contributions of this work are:

- The amateur radio community and the universities are interested in reducing the costs of acquiring and implementing a ground station, the hardware setup proposed here is based on COTS elements, becoming an inexpensive solution accessible to every kind of user.
- The contribution to the network of SATNet ground stations in their hardware and software components of the station and integration with

ADSGS should be highlighted, increasing its potential to become a ground station standard.

- The ADSGS Agent allows ground stations to be efficient in the use of their resources and to integrate artificial intelligence in this Agent, makes it different from the existing stations. In addition, using this algorithm in a ground station networks will increase the control time over satellites in orbit.
- The use of SDR in both the hardware and the software in this proposal suppose a contribution for the tracking of satellites using the amateur radio frequencies, thus increasing the range of these frequencies in the different bands.
- The adaptation of the Hungarian Algorithm makes a contribution in the solution to problems of allocation since it allows to convert non-quadratic matrices into quadratic matrices adapting thus to the solution of more problems of this type.
- A comparative study of the different networks ground stations in the last 15 years was made, emphasizing that they have not used artificial intelligence in their development.
- The use of artificial intelligence to provide autonomy to a ground station increases the reliability on ADSGS to manage this kind of projects.
- ADSGS is a contribution to the telecommunications that are made daily with the satellites in orbit from the ground stations, since it optimizes the time, the cost, the control and the maintenance of the ground system.

6.2. List of Publications Attained

Below is a list of the works done by the authors of this thesis over the last four years. The attachment A contents the abstracts, posters and papers submitted to conferences and/or *magazines specialized*.

- Conference: *V Congreso Internacional de Ciencia y Tecnología Aeroespacial - CICTA 2014*. Automation of Ground Station State of Art. November 18-20, 2014. Bogota, Colombia.
- Conference: 1st Latin American IAA CubeSat Workshop. *A survey on State of Art in Ground Station for Miniaturized Satellites*. December 8-11, 2014. Brasilia, Brazil.
- Conference: 6^o. Workshop em Engenharia e Tecnologia Espaciais 2015, *Uma Abordagem Dinâmica e Autônoma para Operações Espaciais de Pequenos Satélites*. August 18-20, 2015. São Jose dos Campos, Brazil.
- Poster: Spring 2015 CubeSat Developer's Workshop. *Autonomous and Dynamic System for Ground Station*. April 22-24, 2015. San Luis Obispo, CA, USA.
- Conference: II Latin American IAA CubeSat Workshop. *Integration of the INPE Ground Station into the SATNet Network for Supporting Small Satellites Programs in Brazil*. February 28 to March 2, 2016. Ingleses Beach, Florianopolis, Brazil.
- Abstract: 13th Annual CubeSat Developer's Workshop. *Deployment of the SATNet Network at the INPE (Brazil) and University of Vigo (Spain) Facilities with a Central Server at the California Polytechnic State University (USA)*. April 20-22, 2016. San Luis Obispo, CA, USA.
- Abstract: 14th International Conference on Space Operations - SpaceOps 2016. *An Autonomous and Dynamic Approach to Ground Stations*

Control for Small Satellites. May 16-20, 2016. Daejeon, Republic of Korea.

- Conference: *7o. Workshop em Engenharia e Tecnologia Espaciais 2016*, ADSGS - Autonomous Dynamic System Ground Station. September 23-24, 2016. São Jose dos Campos, Brazil.
- Conference: *VI Congreso Internacional de Ciencia y Tecnología Aeroespacial - CICTA 2016*. ADSGS and SATNet. October 27-28, 2016. Cali, Colombia.
- Paper: Ground station for miniaturized satellites: a state of the art survey. Magazine: *Redes de Ingenieria*. ISSN: 2248-762X

6.3. Envisaged Future works

In order to continue with this research work, it is suggested to perform, among others, the following activities:

- Develop and implement the ADSGS system based on the suggested UML modeling. This can be done, for example, by using Python for implementation because it reduces the interface problems between the different ADSGS components.
- It is suggested to develop a specific rule-based expert system in order to facilitate the interoperability with heterogeneous tools like Java, using Python, Delphi or tools for expert systems.
- Implement ADSGS in a 100% compatible station. For this purpose, at *Universidad Pedagógica y Tecnológica de Colombia* – UPTC located in Tunja city, Colombia, was already bought the suggested ground station hardware proposed in this work. This particular station will be part of SATNet, allowing using ADSGS to control it.

REFERENCES

- AGUIRRE, M. **Introduction to space systems: design and synthesis**. New York: Springer, 2012. ISBN: 978-1-4614-3757-4 (Print) 978-1-4614-3758-1 (Online). DOI:10.1007/978-1-4614-3758-1.
- AIR FO. **Air force satellite control network**. Available on: <<http://www.afscn.com>>. Accessed on: 11 Mar. 2016.
- AIRSPY. **SDR software package**. Available on: <<http://airspy.com/>>. Accessed on: 15 Apr. 2016.
- ALONSO, G.; BECERRIL, J. L. **Introducción a la inteligencia artificial**. Ed. Multimedia Ediciones S.A. Barcelona. 1993.
- AMOOZEGAR, F.; RUGGIER, C. **Neural network based satellite tracking for deep space applications**. Automatic Target Recognition XIII, p. 119-135. SPIE 0277-786X/03. 2003.
- ANDRE, D; ASTRO T. Evolving team Darwin United. In: ASADA, M.; KITANO, H. (eds). **RoboCup-98: robot soccer world cup II**. Springer-Verlag, 1999. V. 1604, p. 346-352. Lecture Notes in Computer Science.
- AR2. **High Performance 160 watt mast mounted amateur preamplifiers**. Available on: <<http://www.advancedreceiver.com/page10.html>>. Accessed on: 01 Apr. 2016.
- BABUSCIA, A.; MCCORMACK M.; MUNOZ, M.; PARRA S.; MILLER D. Mit castor satellite: design, implementation, and testing of the communication system. **Acta Astronáutica**, v. 81, p. 111-121, 2012. doi:10.1016.2012.07.005.
- BANZHAF, W.; REEVES, C. **Foundations of genetic algorithms 5**. San Francisco, California, USA: Morgan Kaufmann Publishers, 1999.
- BERNARD, J. A. Use of a rule-based system for process control. **IEEE Control Systems Magazine**, v. 8, n. 5, p. 3-13, 1988.
- BROWN, D.; ROTHERY, P. **Models in biology: mathematics, statistics and computing**. New York: John Wiley & Sons, 1993.
- CANDINI, G. P.; PIERGENTILI, F.; SANTONI, F. Miniaturized attitude control system for nanosatellites. **Acta Astronáutica**, v. 81, p. 325-334, 2012. doi:10.1016/j.actaastro.2012.07.027.
- CCSDR. **Simple schedule format specification**. Draft Recommended Standard, Issue 1. CCSDR 902.1-R-1. Washington D.C. September, 2014.

CHARITOPOULOS, I.; GIANNELOS, I.; KOSMAS, E.; PAPADEAS, D.; PAPADEAS, P.; PAPAMATTHAIIOU, M.; ROUSSOS, N.; TSILIGIANNIS, V.; ZISIMATOS, A. SatNOGS: an open satellite ground station network. In: EUROPEAN CUBESAT SYMPOSIUM, 7., 2015, Athens, Greece. **Proceedings...** Athens: Libre Space Foundation, 2015. Available on: <https://www.cubesatsymposium.eu/download/7ECS_agenda2.pdf>. Accessed on: 18 Mar. 2016.

CHMYREV, V.; SMITH A., KATARIA D.; NESTEROV B.; OWEN C.; SAMMONDS P.; SOROKIN V.; VALLIANATOS F. Detection and monitoring of earthquake precursors: **TwinSat, a Russia–UK satellite project**, **Advances in Space Research**, v. 52, p. 1135-1145, 2013. doi:10.1016/j.asr.2013.06.017.

CHOI, J. S.; SANDERS, A. L. Cost-Effective telemetry and command ground systems automation strategy for the soil moisture active passive (SMAP) mission. In: SPACEOPS 2012 CONFERENCE, 2012, Stockholm, Sweden. **Proceedings...** Reston, VA: AIAA, 2012. doi:10.2514/6.2012-1275978.

CLARK, C. **Software defined radio**: with gnu radio USRP. McGraw-Hill Professional, 2008.

CUTLER, J. Ground station markup language. In: IEEE AEROESPACE CONFERENCE, 2004, Stanford. **Proceedings...** Space System Development Laboratory, Stanford University, Stanford, California, 2004.

CUTLER, J. **Ground station virtualization**. Stanford, California: NASA, Stanford University, 2006.

CLUTER, J. FOX, A. A framework for robust and flexible ground station networks. **Journal of Aerospace Computing, Information, and Communication**, v. 3, March 2006. Stanford University, Stanford, California, 2006.

CUTLER, J. W., Ground station virtualization. In: INTERNATIONAL SYMPOSIUM ON REDUCING THE COST OF SPACECRAFT GROUND SYSTEMS AND OPERATIONS (RCSGSO), 5., 2004, Stanford, CA. **Proceedings...** Stanford University, Stanford, California, 2004.

CUTLER, J. W.; KITTS, C. Mercury: a satellite ground station control system. In: AEROSPACE CONFERENCE, 1999, Snowmass at Aspen, CO. **Proceedings...** Piscataway, NJ.: IEEE, 1999. v. 2. p. 51-58. doi:10.1109/AERO.1999.793142.

DAMIANI, S.; DREIHAHN H.; NOLL, J.; NIEZETTE, M.; CALZOLARI, G. **Automated allocation of ESA ground station network services**. Darmstadt, Germany: American association for artificial intelligence, 2006. D-64293.

DASCAL, V.; DOLEA, P.; PALADE, T.; CRISTEA, O. Aspects of a low-cost ground station development for GENSO network. **Acta Technica Napocensis - Electronics and Telecommunications**, v. 52, n. 4, p. 36-40, 2011.

DE CARVALHO, M.; JOTHA, L.; LIMA, J.; BIONDI, R.; AQUINO, P.; LIMA, D. **Estudo de estratégias de mitigação de detritos espaciais para uma constelação de nano satélites de coleta de dados ambientais**. Natal: Instituto Nacional de Pesquisas Espaciais. Available on <[http://www.crn2.inpe.br/conasat1/Documentos/artigos/Artigo%20aceito%20no%2016th%20ISU%20Anual%20International%20Symposium%20em%202012%20\(Equipe%20CONASAT\).pdf](http://www.crn2.inpe.br/conasat1/Documentos/artigos/Artigo%20aceito%20no%2016th%20ISU%20Anual%20International%20Symposium%20em%202012%20(Equipe%20CONASAT).pdf)> 2012. Accessed on: 25 Apr. 2016.

DEAN, T.; ALLEN, J.; ALOIMONOS, Y. **Artificial Intelligence** - theory and practice. Addison-Wesley, 1995. ISBN 0-8053-2547-6.

DECHAO, R.; TAO S.; LU, C.; XIAOQIAN, C.; YONK, Z. Attitude control system design and on-orbit performance analysis of nanosatellite - Tian Tuo 1. **Chinese journal of aeronautics**, v. 27, p. 593-601, 2014. doi:10.1016/j.cja.2013.11.001.

D'ERRICO, M. **Distributed space missions for earth system monitoring**. Springer Science & Business Media. 2012 - 678 p. ISBN 978-1-4614-4540-1 Doi: 10.1007/978-1-4614-4541-8 IAA – International Academy of Astronautics, Microcosm Press. El Segundo, California, 2012.

FAS. National security space road map. **Air Force Satellite Control Network (AFSCN) (U)**. Available on: <<http://fas.org/spp/military/program/nssrm/categories/sfssoafs.htm>>. Accessed on: 19 Mar. 2016.

DEVI, B.; WILLIAMS, L. **An architecture for AFSCN ground interface modernization**. GSAW 2012. Published by The Aerospace Corporation. Available on: <<http://gsaw.org/wp-content/uploads/2013/07/2012s01devi.pdf>>. Accessed on: 11 Mar. 2016.

JET PROPULSION LABORATORY. DSN. **Deep space network**. California Institute of Technology. Available on: <<http://eyes.nasa.gov/dsn/dsn.html>>. Accessed on: 11 Mar. 2016.

DUSSAUZE, J.; FELTRIN, G.; TROILLARD, J. AstroTerra control ground segment: operations concept and implementation. In: SPACEOPS 2012

CONFERENCE, 2012, Stockholm, Sweden. **Proceedings...** Reston, VA: AIAA, 2012. doi:10.2514/6.2012-1289133.

EDLUND, K.; GREEN, M.; KRAGELUND, M.; MICHELSEN, A.; STOUGAARD, R. Generic distributed mission control center for student satellites 2004, IEEE AEROSPACE CONFERENCE, 2004, Big Sky, MT, USA. **Proceedins...** IEEE, 2004. v. 22.

ESPINDOLA J.; FERRO R.; MESA J. Direccionamiento automático de antenas en estaciones terrenas de seguimiento a pico satélites. **Revista Tecnura. Universidad Distrital.** v. 17, n. 35, p 26-37, 2013. ISSN: 0123-921X Bogotá.

Expert SINTA. **Manual do usuário laboratório de inteligência artificial/LIA-UFC.** Projeto financiado pela CAPES e CNPq. 1996.

FEDOR, J.; NISHINAGA R.; SUNSHINE C.; JAMES P. Evolution of the air force satellite control network. **Crosslink**, v. 7, edition 1 (Spring 2006) ISSN 1527-5264. Available on: <<http://aerospace.wpengine.netdna-cdn.com/wp-content/uploads/crosslink/V7N1.pdf>>. Accessed on: 11 Mar. 2016.

FULLER, R. **Neural fuzzy systems.** Abo Akademi University, 1995. ISBN 951-650-624-0.

GARPESTAD, O. **The IARU satellite frequency coordination 2015.** The International Amateur Radio Union. Available on: <<http://www.itu.int/en/ITU-R/space/workshops/2015-prague-small-sat/Presentations/IARU%20Satellite%20coordination.pdf>>. Accessed on: 17 Mar. 2016.

GENSO. **Project background.** Available on: <http://www.esa.int/Education/Project_background>. Accessed on: 15 Mar. 2016.

GINSBERG, M. **Essentials of Artificial Intelligence.** Morgan Kaufmann Publishers, San Francisco, California, 1993.

GOLDBERG, D. **Genetics algorithms in search, optimization and machine learning.** Boston, MA: Addison Wesley. Longman Publishing Co., Inc. USA, 1989. ISBN:0201157675.

GRIFFITH, R. C. **Mobile CubeSat Command and Control (mc3).** 2011. 76 p. Dissertation (Master in Space Systems Operations) - Naval Postgraduate School, Monterey, California, 2011.

GSN. **GS management service manual (English version).** GMS manual for GSN projects 2006/07/10 (Release). Available on:

<http://www.astro.mech.tohoku.ac.jp/~gsn/en/documents/gms7e_manual_060710.pdf>. Accessed on: 14 Mar. 2016.

HARMON, P.; KING, D. **Expert systems** - applications of artificial intelligence in business. Madrid: Wiley. 1988.

HENRYRADIO. **Tohtsu coaxial relays, CX-600N**. Available on: <<http://www.henryradio.com/tohtsu.html#cx600n>>. Accessed on: 15 Apr. 2016.

HERTZ, J.; KROGH, A.; PALMER, R. G. **Introduction to the Theory of neural Computation**. Addison – Wesley, Redwood City, CA 1991.

HUNGARIANALGORITHM.COM, **The hungarian algorithm**. Available on: <<http://www.hungarianalgorithm.com/index.php>>. Accessed on: 7 Sep. 2016.

ICOM. **IC-7100 HF/VHF/UHF transceiver**. Available on: <<http://www.icomamerica.com/en/products/amateur/hf/7100/default.aspx>>. Accessed on: 14 Apr. 2016.

IEEE STANDARDS ASSOCIATION. **IEEE 1471-2000 - Practice for Architectural Description for Software-Intensive Systems**. Available on: <<http://standards.ieee.org/findstds/standard/1471-2000.html>>. Accessed on: 7 Agu. 2016.

INGA, D.; VELASQUEZ I.; CACHI D. Implementation of an small-satellites automatic tracking system to reception and interpretation of beacon signals from ctic-uni's ground station. **Revista Eciperu**, v. 8, p. 87-91, 2011.

JAFFER, G.; NADER, R.; KOUDELKA, O. Internet-to-orbit gateway and virtual ground station: a tool for space research and scientific outreach. **Acta Astronáutica**, v. 69, p. 658-666, 2011. doi:10.1016/j.actaastro.201.05.021.

JET PROPULSION LABORATORY (JPL). **Deep space network**. California Institute of Technology. Available on: <<http://deepspace.jpl.nasa.gov/about/DSNComplexes/>>. Accessed on: 11 Mar. 2016.

KALMAN, A. E. Pumpkin's Colony I Cubesat bus: past, present and future. In: GAINSTAM Workshop, November 4, 2009.

KONTIO, M. **Architectural manifesto: designing software architectures**, Part 5. IBM Corporation, 2008. Available on: <<https://www.ibm.com/developerworks/architecture/library/ar-archman5/ar-archman5-pdf.pdf>>. Accessed on: 15 May. 2016.

KRUCHTEN, P. Architectural Blueprints - The “4+1” View Model of Software Architecture. **IEEE Software**, v.12, n. 6, p. 42-50. 1995.

KUHN, H. W. The Hungarian method for the assignment problem. **Naval Research Logistics Quartely**. v. 2, n. 1 e 2, p. 83-97, 1955.

LANGE, C.; FOLEY, J. Ground station automation with APO and “GENSO” In: AIAA PLUG-N-PLAY MISSION OPERATIONS WORKSHOP, 2011, San Jose, US. **Proceedings...** San Jose: IEEE, 2011.

LANNES, C.; PISSIAS, P.; RICCIO F. A new generation of monitoring and control system for ESTRACK. In: AEROSPACE 2012 CONFERENCE, 2012, Stockholm, Sweden. **Proceedings...** Reston, VA: AIAA, 2012. v. 3, p. 1890-1897 doi: 10.2514/6.2012-1286386.

LE CUN, Y. **Learning processes in an asymmetric threshold network**. In: E. BIENESTOCK, F. FOGELMEN SOULIÉ; WEISBUCH, G. (Eds.), **Disordered Systems and Biological Organization**: disordered systems and biological organization proceedings of the NATO advanced research workshop. Berlin: Springer, 1985;

LEFFINGWELL, D.; MANAGING, D. **Software requirements**: a use case approach. Publisher: Addison Wesley, 502 pp., 2003.

LEVEQUE, K.; PUIG-SUARI, J.; TURNER, C. Global educational network for satellite operations (GENSO). ssc07-xi-6. In: ANNUAL AIAA/USU CONFERENCE ON SMALL SATELLITE , 21., 2007, San Luis Obispo, CA . **Proceedings...** San Luis Obispo, CA: California Polytechnic State University, 2007.

LIAO, S.-H., Expert system methodologies and applications - a decade review from 1995 to 2004. **Expert Systems with Applications**, v. 28, p. 93-103, 2005.

M2. **Antenna systems**, SATPACK #1 EB144/EB432/CROSSBOOM. Available on: <<http://www.m2inc.com/amateur/satpack-1-eb144-eb432-crossboom/>>. Accessed on: 14 Apr. 2016.

MACHADO, F. N. **Analise e gestão de requisitos de software**: onde nascem os sistemas. Editora Erica, 2011. 286p. ISBN 978-85-365-0362-2 .

MACHADO, R.J.; ROCHA, A.F. A hybrid architecture for fuzzy connectionist expert systems. In: KANDEL, A; LANGHOLZ, G. (eds). **Hybrid architectures for intelligent systems**. Boca Raton, FL: CRC Press Book: 1992.

MCSUSH, J. **SDR. et WF**. Available on: Copyrighted free use
<<https://commons.wikimedia.org/w/index.php?curid=8831874>> Accessed on: 2 Jul. 2016.

MDT. Mercury design team. **Mercury architecture description - version 1.1**, 2006. Stanford University, Stanford, California, 2006. Available on:
<<http://mgsn.sourceforge.net/docs/arch1.1.html>>. Accessed on: 14 Mar. 2016.

MELVILLE, N. **Global educational network for satellite operations**: concept, progress and demonstration. Guildford: Surrey Space Centre, 2008.

_____. **Global educational network for satellite operations**. Hyderabad: International Space Education Board, 2007.

MISHKOFF, H. C. **In depth**: artificial intelligence. Madrid: Anaya, 1988.

MITCHELL M. **An introduction to genetic algorithms**. Cambridge, MA: MIT Press Cambridge, 1998. SBN: 0-262-13316-4.

MITCHELL, T. **Machine learning**. McGraw-Hill, 1997. ISBN 0070428077.

MU, Y.; HAO, W.; CHANGJU, W.; CHUNCHUI, W.; LICONG, D.; YANGMING, Z.; ZHONGHE, J. Space flight validation of design and engineering of the ZDPS-1A pico-satellite. **Chinese Journal of Aeronautics**, v.. 25, p. 725-738, 2011. doi:10.1016/S1000-936(11)60439-1.

MÜLLER, P.; DOAT, Y. Configuration management in a ground station network. In: SPACEOPS 2012 CONFERENCE, 2012, Stockholm, Sweden.
Proceedings... Reston, VA: AIAA, 2012. doi/abs/10.2514/6.2012-1263858. 2012.

MULLER, R; VERMILLION C. A Ground Station for use with small satellites. In: USU CONFERENCE ON SMALL SATELLITES, 1., 1987, North Logan, Utah.
Proceedings... October 7-9, 1987.

NAKAMURA, Y.; NAKASUKA, S. Ground station network to improve operation efficiency of small satellites and its operation scheduling method. In: INTERNATIONAL ASTRONAUTICAL CONGRESS 2006. ASTRODYNAMICS SYMPOSIUM (C1.) MISSION OPERATIONS (6.), 2006, Tokyo. **Proceedings...** Tokyo: University of Tokyo, Japan, 2006.

NASA. **Strategic Plan 2014**. Available on:
<https://www.nasa.gov/sites/default/files/files/FY2014_NASA_SP_508c.pdf>. Accessed on: 10 Sep. 2016.

NASA. **Small spacecraft technology state of the art**. Moffett Field, California, U.S.A: NASA Ames Research Center Moffett Field, 2014a.211p.

NEGNEVITSKY, M. **Artificial Intelligence**: a guide to intelligent systems. 3. ed. Addison Wesley, 2011. ISBN: 1408225743.

NIKOLOPOULOS, C. **Expert systems**: introduction to first and second generation and hybrid knowledge based systems. 1. ed. New York: Taylor & Francis, 1997. 331 p. ISBN (9780824799274).

NILSSON, N. **Inteligencia artificial**: una nueva síntesis Aravaca, Madrid: McGraw-Hill, 2001. ISBN 84-481-2824-9.

NONSOQUE, C. A. Q. **Montaje e implementación de una estación terrena para el seguimiento de satélites de órbita baja**. 2012. 99p. (TCC Ingeniero de Sistemas y Computación). Escuela de Sistemas y Computación, Universidad Pedagógica y Tecnológica de Colombia, Tunja (Boyacá).

NUTAQ. **A short history of software-defined radio (SDR) technology**. 2014 Available on: <<http://nutaq.com/en/blog/short-history-software-defined-radio-sdr-technology>>. Accessed on: 25 Apr. 2016.

ODA, Y.; KOMATSU, M.; KURAHARA, N.; NAKAMURA, Y.; SAKAMOTO, Y.; NAKASUKA, S.; WICKRAMANAYAKE, A.; FERNANDO, P. Improvement in university satellite operation using ground station network. **Trans, JSASS Space Tech**. Japan, v. 7, n. 26, p. Tu_31-Tu_34, 2009.

PALAZÓN A. Los sistemas expertos como ayuda a las empresas. **Dirección y progreso**, n. 119, p. 22-24, 1991.

PARKER, D. **Learning Logic**. Cambridge, MA: MIT, Center for Computational Research in Economics and management Science, 1985. Technical report TR-87.

PELAEZ J.; VELASCO J. **Sistemas inteligentes**. 2012. Available on: <<http://www.aic.uniovi.es/ssii>>. Accessed on: 11 Jul. 2016.

PERSONAL PAGE OF UZ7HO. The software Packet-Radio TNC. Available on: <<http://uz7.ho.ua/packetradio.htm>> Access on: 16 march. 2017.

POOLE, D.; MACKWORTH, A.; GOEBEL, R. **Computational intelligence**: a logical approach. New York: Oxford University Press, 1998.

PREINDL, B.; MEHNEN, L.; RATTAY, F.; NIELSEN, J. D.; KRINNINGER, S.; SØRENSEN, K. A global satellite link sensor network. In: IEEE CONFERENCE

ON SENSORS, CHRISTCHURCH, 8., 2009, New Zealand. **Proceedings...** Vienna: Institute of Analysis and Scientific Computing, Vienna University of Technology, Austria, 2009a. ISBN. 978-1-4244-5335-1.

PREINDL, B. L.; RATTAY M.; NIELSEN, J. D. Design of a small satellite for performing measurements in a ground station network. In: INTERNATIONAL WORKSHOP ON SATELLITE AND SPACE COMMUNICATIONS (IWSSC), 2009, Siena-Tuscany, Italy. **Proceedings...** Vienna Technical University, Austria and Aalborg University, Denmark, 2009b.

PROBECOM. **Meter earth station antenna**. Available on: <http://www.probecom.cn/up_files/1.8m%20earth%20antenna%20datasheet.pdf>. Accessed on: 10 Apr. 2016.

RADIXON GROUP. **Software defined radio**. Available on: <<http://www.winradio.com/home/facts.htm>> Accessed on: 12 Jul. 2016.

REED, J. **Software radio**: a modern approach to radio engineering. New Jersey: Prentice Hall PTR, 2002. 561p. ISBN 0-13-081158-0.

RF HAMDESIGN. **SPID, type BIG-RAS/HR azimuth & elevation rotor**. Available on: <<http://www.rfhamdesign.com/products/spid-hr-antenna-rotators/bigrashr/index.php>>. Accessed on: 10 Apr. 2016.

RODRIGUEZ, J. E. O. **Processo de referência para o desenvolvimento da arquitetura de uma estação terrena para pico e nanosatélites**. 2016. 235 p. Dissertation (Master in aerospace engineering) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2016. Available on on: <<http://urlib.net/8JMKD3MGP3W34P/3LDAGLL>>. Accessed on: 08 Nov. 2016.

ROSE, C. W.; WIRTHLIN, J. R. Using m&s to maximize space satellite data collection with multiple ground stations. **Procedia Computer Science**, v. 8, p. 124-128, 2012. doi:10.1016/j.procs.2012.01.024.

RTL-SDR.COM, **About RTL-SDR**. Available on: <<http://www.rtl-sdr.com/about-rtl-sdr/>> Accessed on: 2 Jul. 2016.

RUSSELL, S.; NORVIG, P. **Inteligencia artificial**. Un enfoque moderno. 2. ed. Prentice Hall Int., 2004. ISBN 978-84-205-4003-0.

SALEHI-ABARI, A.; WHITE, T. The impact of naive agents in heterogeneous trust-aware societies. In: INTERNATIONAL WORKSHOP ON MULTI-AGENT SYSTEMS AND AGENT-BASED SIMULATION, 2010, Budapest, Hungary. **Revised and selected papers**. Springer, March 2010. p. 110–122, LNAI 5683.

SANDAU, R. Status and trends of small satellite missions for earth observation, **Acta Astronáutica**, v. 66, p. 1-12, 2009. doi:10.1016/j.actaastro.209.06.008.

SATNet. **The SATNET network**. Available on:
<https://satnet.aero.calpoly.edu/accounts/login/?next=/accounts/login_ok/>.
Accessed on: 9 Agu. 2016.

SCHULZ, D. A. Colony: a new business model for R&D. In: ANNUAL AIAA/USU CONFERENCE ONSMALL SATELLITES, 31., 2010, Logan, UT, USA. **Proceedings...** Utah State University, 2010..

SINTA. Expert SINTA – **Uma ferramenta visual para criação de sistemas especialistas** – Versão 1.1 – Manual do Usuário. – Laboratório de Inteligência Artificial/LIA - UFC - 1996.

SORENSEN, T.; PILGUER, E.; WOOD, M.; GREGORY, E.; NUNES, M. Development of the mission operations support tool (MOST) In: SPACEOPS 2010 CONFERENCE, 2010, Hunstville, Alabama. **Proceedings...** 2010. doi:10.2514/6.2010-2230.

SPIRES, D.; STURDEVANT, R. From advent to Milstar: The U.S. air force and the challenges of military satellite communications. In: NASA (ed.). **SP-4217 beyond de ionosphere**. 1990. Chapter 7. Available on:
<<http://history.nasa.gov/SP-4217/ch7.htm>>. Accessed on: 22 Sep. 2016.

SPRINGMANN, J. C.; CUTLER, J. W.; BAHCIIVAN, H. Initial flight results of the radio aurora explorer. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, , 62., 2011, Cape Town. **Proceedings...** Cape Town, South Africa, 2011.

SSB. **UEK-3000 2.4GHz. mode "S" down converter**. Available on:
<<http://www.ssbusa.com/modes.html>>. Accessed on: 10 Apr. 2016.

STOFF. **Orbitron software**. Available on: <<http://www.stoff.pl/>>. Accessed on: 12 Jul. 2016.

SUPERKUH. **RTL-SDR and GNU radio with realtek rtl2832u [elonics e4000/raphael micro r820t] software defined radio receivers**. 2016. Available on: <<http://superkuh.com/rtlsdr.html>>. Accessed on: 15 Apr. 2016.

TOMINAGA, J. **Simulador de satélites para verificação de planos de operações em vôo**. 2010. 174 p. IBI: <8JMKD3MGP7W/37HL3J8>. (sid.inpe.br/mtc-m19@80/2010/05.24.18.55-TDI). Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2010. Disponível em:
<<http://urlib.net/8JMKD3MGP7W/37HL3J8>>..

TUBIO, R. **User specification**. SATNet Project 2013. Available on: <<https://github.com/satnet-project/documentation/blob/master/documentation/satnet-1-UserSpecification-DRAFT-1-2013.10.11.pdf>>. Accessed on: 18 Mar. 2016.

TUBÍO, R.; VAZQUEZ, A.; PUIG, J.; KURAHARA, N.; BELLARDO, J. The SATNet project: towards an open-source ground stations network for CubeSats. SPRING CUBESAT DEVELOPERS' WORKSHOP 2014. **Powerpoint...** Available on: <http://mstl.atl.calpoly.edu/~bklofas/Presentations/DevelopersWorkshop2014/Tubio_SATNet.pdf>. Accessed on: 17 Mar. 2016.

VERON, M.; MANBER, U. **Distributed round-robin and first-come first-serve protocols and their application to multiprocessor bus arbitration**. In: ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE, 15., 1988, Honolulu, Hawaii. **Proceedings...** Honolulu, 1988. Computer Sciences Technical Report #745.

WASSERMANN, P. D. **Neural computing: theory practice**. New York: Coriolis Group (Sd) , 1985. v.. 1, 230p.

WERTZ, J. R.; EVERETT, D. F.; PUSCHELL, J. J. **Space mission engineering: the new smad** (space technology library, vol. 28). Hawthorne, CA: Microcosm Press, 2011.

WHITE, D. J.; GIANNELLOS, I.; ZISSIMATOS, A.; KOSMAS, E.; PAPADEAS, D.; PAPADEAS, P.; PAPAMATHAIIOU, M.; ROUSSOS, N.; TSILIGIANNIS, V.; CHARITOPOULOS, I. **SATnogs: satellite networked open ground station**. Engineering faculty publications. Paper 40. 2015. Available on: <http://scholar.valpo.edu/engineering_fac_pub/40>. Accessed on: 19 Mar. 2016.

WIF. Wireless Innovation Forum. **What is software defined radio**. Available on: <http://www.wirelessinnovation.org/Introduction_to_SDR> Accessed on: 25 Apr. 2016.

WOELLERT, K.; EHRENFREUND, P.; RICCO, A.; HERTZFELD, H. Cubesats: cost-effective science and technology platforms for emerging and developing nations. **Advances in Space Research**, v. 47, p. 663-684, 2010. doi:10.1016/j.asr.2010.10.009.

XHAFA, F.; HERRERO, X.; BAROLLI, A.; BAROLLI, L.; TAKIZAWA, M. Evaluation of struggle strategy in genetic algorithms for ground stations scheduling problem. **Journal of computer and system sciences**, v. 79, p. 1086-1100, 2013. doi:10.1016/j.jcss.2013.01.023.

YAESU. **G-5500, complete az-el rotation system.** Available on:
<<http://www.yaesu.com/indexVS.cfm?cmd=DisplayProducts&ProdCatID=104&encProdID=79A89CEC477AA3B819EE02831F3FD5B8&DivisionID=65&isArchived=0>>. Accessed on: 11 Apr. 2016.

YAMADA, T. CCSDS telemetry/telecommand standards restructured as communications protocols. In: SPACEOPS 2002, 2002, Houston Texas. **Proceedings...** Reston, VA: Institute of Space and Astronautical Science. Doi: 10.2514/6.2002-T5-11.

YOUNGBLOOD, G. **A software-defined radio for the masses, Part 1.** QEX: the forum for communications experimenters. Austin, TX, 2002. p. 13-21.

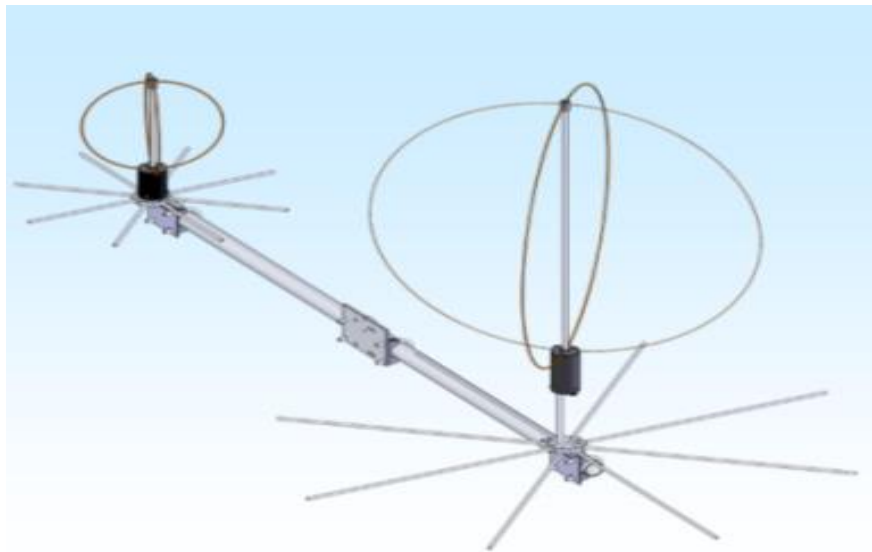
APPENDIX A - THE ADSGS-ADAPTED GROUND STATION HARDWARE

The following is the suggested hardware shown in the chapter 3 figure 3.3, for creating an ADSGS type ground station:

A.1. Antennas

They have the function to broadcast and receive radio waves, to convert the guided wave by the transmission line (cable or waveguide) into electromagnetic waves that can be transmitted in free space. In this architecture are used amateur radio frequencies: VHF for 145 MHz, UHF for 436 MHz and S-band for 2.4 GHz. These antennas of 70 cm and 2 m respectively are crossboom type, as shown in Figure A.1; they are omnidirectional, vertically polarized for long range FM, horizontally stacked to increase the gain of the system by up to 3dB.

Figure A.1 - VHF-UHF CrossBoom antenna



Source: M2 Antenna systems, (2016)

A.2. The S-band antenna

It is a 1.8 m plate with circular polarization from the brand Shaanxi Probecom Microwave Technology Co. Ltd. a specialized company engaged in the design

and manufacture of satellite communication ground station antenna with a selected range from 1.8m to 16m (see Figure A.2) (Probecom, 2016).

Figure A.2 - S-band antenna



Source: Probecom (2016)

A.3. Preamplifier

This device is used to boost the signal strength, increasing the voltage of the input signal. The preamplifier levels the electric voltage from every audio source; then, when evened, send them as an input signal to another device. Two preamplifiers are required: a 145 MHz for VHF and a 436 MHz for UHF. This device features the following technical specifications for both VHF and UHF: (AR2, 2016)

- Low noise figure
- Automatically switches off during transmit
- 160 watt through power capability
- No transceiver modification necessary
- High immunity to overload
- Completely shielded
- Rugged custom weatherproof enclosure
- Mounts to 1-1/4" - 3-1/4" diameter masts

This series of preamplifiers, shown in Figure A.3, are specifically designed for mast mounting. Locating the preamplifier close to the antenna reduces feedline losses, locating it ahead of the preamplifier reduces the system noise figure.

Figure A.3 - VHF/UHF Preamplifier



Source: AR2 (2016)

A.4. 2.4 GHz Converter

It's an electronic device able to convert an analogic signal into a digital one throughout a quantifier and commonly it is coded into a particular binary code, which represents a single voltage value. In this case, it is going to be used a down-converter UEK-3000 from SSB Electronic (see figure A.4), because it meets every satellite operators' requirements and it's available in 144 MHz or 432 MHz frequencies. This converter offers a noise rate below 0,7 dB, conversion gain of 30 dB and needs no preamplifier.

Figure A.4 - S-band 2,4 GHz converter



Source: SSB (2016)

A.5. Radio transceiver IC-7100

It is a device with both a transmitter and a receiver, with common circuitry used to transmit or receive both analog and digital signals. In this architecture is recommended to use ICOM IC-7100 multi-frequency radio, using frequencies of 145 MHz for RX and 436 MHz for TX.

This IC-7100 radio (see Figure A.5), fully covers the HF, 50, 144, 430/440 MHz amateur bands in multiple modes, providing 100W on HF/50MHz bands, 50W on 144MHz band and 35W on 430/440MHz band (ICOM, 2016).

Figure A.5 - IC-7100 Transceiver



Source: ICOM (2016)

A.6. Rotor

This is a servomotor required to move the antennas in both azimuth (360°) and elevation (180°), for VHF and UHF antennas. A Yaesu G-5500 (see Figure A.6) was chosen for this architecture, due to its features and price. This rotor provides a complete Azimuth-Elevation rotation system that's ideal for most all Amateur applications. The dual controller allows keeping track of all aspects of the antenna system's positioning, and the G-5500 includes an interface jack for computer control, supported by the optional GS-232A Computer Controlled as well as many aftermarket control programs and interfaces (YAESU, 2016).

Figure A.6 - Yaesu G-5500 Rotor



Fuente: YAESU (2016)

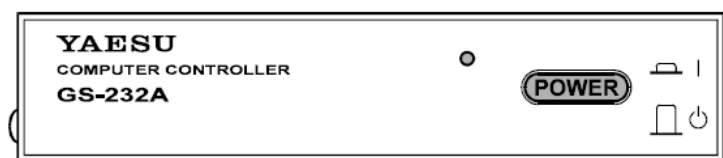
A.7. VHF/UHF Rotor Interface

The GS-232A, shown in Figure A.7, provides digital control of most models of Yaesu antenna rotators from the serial port of an external personal computer. The GS-232A contains its own microprocessor with ROM and RAM (memory), and a 10-bit analog-to-digital (A-D) converter. The 3-wire async serial line can be configured for serial data rates from 150 to 9600 baud. The GS-232A has a

DB-9 “male” connector for connection to the (RS-232C) COM port of a computer (YAESU, 2016).

Firmware on the GS-232A supports either direct keyboard control, or commands from programs written specifically to support it (software is not supplied by Yaesu). In addition to reading and setting antenna angle and rotation speed, the firmware includes clocked positioning routines to automatically step the antenna through up to 3800 angles at programmable intervals, such as for tracking band openings or satellites (with an elevation rotator) (YAESU, 2016).

Figure A.7 - GS-232A Control Interface



Source: YAESU (2016)

A.8. S-band antenna rotor

Based on good practices for this architecture it is recommended to use an S-band rotor different from the one used for VHF/UHF antennas. The BIG-RAS/HR type rotor, shown in Figure A.8, from RF Hamdesign features a precise rotor system along with multiple user configurable functions.

The BIG-RAS/HR rotor is a two engine rotor with two counting-position sensors installed (transmission 0,0058 degree on impulse) counting will be 0,1 degree on impulse. It also features its own controller with USB connection.

Figure A.8 - S-band BIG-RAS/HR rotor



Source: RF HAMDESING (2016)

A.9. S-Band rotor interface

The SPID set, BIG-RAS/HR and MD-01 type controller features a build in track interface with USB connection and to have a high precision (track) rotator system. It controls the S-band antenna rotor and allows highly precise movement in both elevation and azimuth.

Rotor Controller MD-01 shown in Figure A.9, has many functions which can be configured by the user, a dual voltage power supply PS-01 is available and a switcher for external devices.

Figure A.9 - MD-01 Rotor Controller



Source: RF HAMDESING (2016)

A.10. Relay

It is an electrically operated switch that commonly uses an electromagnet to mechanically operate it. Relays are used to control a circuit by a low-power

signal, or where several circuits must be controlled by one signal. Henry Radio is a distributor of coaxial relays from Japanese brand Tohtsu. These relays are typically used as antenna relays for any RF switching. This architecture uses a CX-600N relay, shown in Figure A.10, with the following specifications:

Specifications (HENRYRADIO, 2016):

RF Impedence: 50 Ohms

Voltage:

Pull in: 10 Vdc - 14 Vdc (12 Vdc Nominal)

Drop out: 3.0 Vdc minimum

Current: 160 mA at 12 Vdc at 20 degrees C

Coil resistance: 75 Ohms +/-10% at 20 degrees C.

Switching Time:

Pull in: 20 ms typical

Drop out: 15 ms typical

Life Expectancy: 1,000,000. Operation min.

Mechanical:

Contact arrangement: SPDT

Operational temperature: -25 to +50 degrees C.

Weight: Approx 230 grams

Mounting: Two screws

Cable connections: 3 N Type female connectors

Figure A.10 - CX-600N relay



Source: HENRYRADIO (2016)

A.11. AirSpy SDR

The Airspy, shown in Figure A.11, is a Software Defined Radio (SDR) (REED, 2002), communication receiver with high performance for TX signals in VHF/UHF bands between 24 MHz and 1,8 GHz with 10 MHz of instantaneous view. It also features a tight integration with SDR# software. It is used for VHF/UHF/S-band receiving.

Figure A.11 - AirSpy SDR



Source: AIRSPY (2016)

Table A.1 shows commercial values for this architecture, as shown on providers' websites.

Table A.1 - Element values for AMSAT-BR

| Ref. | Description | Item | USD |
|----------------|---|------|----------|
| EB144/432 | VHF/UHF Antennas: SATPACK #1 EB144/EB432/CROSSBOOM | 1 | 510,00 |
| MSP144 | VHF Mast mount preamplifier: MSP144VDG-160 | 1 | 280,00 |
| MSP432 | UHF Mast mount preamplifier: MSP432VDG-160 | 1 | 280,00 |
| UEK-144 | 2.4 GHz Mast mount downconverter: UEK- 3000-144 | 1 | 580,00 |
| G-5500 | VHF/UHF antenna rotor: Yaesu G-5500 | 1 | 750,00 |
| MD01 | Dish antenna rotor: SPID, type BIG- RAS/HR Azimuth & Elevation | 1 | 2,750,00 |
| CX600N | Coaxial Relays: Tohtsu CX600N | 4 | 460,00 |
| IC-7100 | VHF/UHF Transmitter: ICOM IC-7100 | 1 | 1,600,00 |
| SDR- AirsPy | SDR Receivers: AirSpy | 1 | 220,00 |
| Sband19 | S-Band antenna: 1.9m dish | 1 | 270,00 |
| | Plugs and cables | 20 | 250,00 |
| | Total | | 7,950,00 |

Source: Author

The cost of these elements does not exceed \$ 8.000 USD, thus contributing to reduce the costs of a ground station.

APPENDIX B - SOFTWARE DEFINED RADIO

B.1. Concept

SDR is a radio system supported by DSP where hardware components are implemented via software with the advantage of improving operational capacity and reducing the costs of communication devices. SDR is featured by the flexibility and the ability to change its operation. Developing radio through software allowed the upgrade of newer and better ways to improve performance rate without frequently changing the hardware (YOUNGBLOOD, 2002).

B.2. SDR Hardware

Figure B.1 shows a hardware board DVB+T+DAB+FM, used as SDR and created for TV reception through FM. This kind of device is currently used by amateur radio community and it is known as *Dongle* and has become help for software developers in digital data processing. Among the most popular devices are *FUNcube Dongle*¹³ y *GAUSS Ground Dongle*¹⁴.

Figure B.1 - SDR - Hardware Board










Source: Superkuh (2016)

¹³ Available on: <<http://www.funcubedongle.com/>>. Accessed on: 22 May. 2016

¹⁴ Available on: <<http://www.gaussteam.com/services/satellite-subsystem/radio/>>. Accessed on: 07 Jun 2016

Table B.1 shows a comparison between the most common SDR hardware nowadays. It is going to be compared: the lowest and maximum tone, the transmission bandwidth, ADC resolution, transmitting ability and approximate market value.

Table B.1 - Comparisons with other common SDR

| SDR | Image | Low Tune (MHz) | Max Tune (MHz) | RX Bandwidth (MHz) | ADC Resolution (Bits) | Transmit? (Yes/No) | Price (\$USD) |
|--------------------|---|----------------------|----------------------|--------------------------|-----------------------------|-----------------------|------------------|
| RTL-SDR (R820T) |  | 24 | 1766 | 3.2 | 8 | No | ~20 |
| Funcube Pro+ |  | 0.15 410 | 260 2050 | 0.3192 | 16 | No | ~200 |
| Airspy |  | 24 | 1800 | 10 | 12 | No | 199 |
| SDRPlay |  | 0.1 | 2000 | 8 | 12 | No | 149 |
| HackRF |  | 30 | 6000 | 20 | 8 | Yes | 299 |
| BladeRF |  | 300 | 3800 | 40 | 12 | Yes | 525 |
| USRP 1 |  | DC | 6000 | 64 | 12 | Yes | 700 |

Source: Adapted from (RTL-SDR.COM, 2016)

One of the advantages of SDR is re-configurability, which results in its flexibility. The key part of an SDR system is the software, which can be modified and updated at any time (RADIXON GROUP, 2016).

B.3. SDR Software

Both hardware and software used in SDR is open source, have developed several applications like SDR#, FlexRadio Systems, HDSDR, PowerSDR, among others. The GNU Radio software is free and open source; it is used as a tool to develop SDR (CLARK, 2008). Most of this software is compatible with the list as described in the table B.2

Below there is a list of the most used software and a brief description of each one. Later, in Table B.2 there is a comparison between different SDR compatible software made so far.

1. **SDR#** (pronounced “SDR Sharp”) is the most popular free RTL-SDR compatible software in use at the moment. It is relatively simple to use compared to other SDR software and has a simple set up procedure. SDR# is a simple to use program that also has some advanced features. It has useful modular plugin type architecture, and many plugins have already been developed by third party developers. The basic SDR# download without any third party plugins includes a standard FFT display and waterfall, a frequency manager, recording plugin and a digital noise reduction plugin. SDR# also decodes RDS signals from broadcast FM.
2. **HDSDR**. (High Definition Software Defined Radio) Is based on the old WinRAD SDR program. HDSDR supports the RTL-SDR through use of an ExtIO.dll module. DSDR has some extra advanced features; users will also find an Audio FFT and waterfall display on the bottom of the screen. The output audio can also be bandpass filtered by dragging the filter borders on the display. Bandpass filtering the audio can really help clean up a noisy signal. The audio processing also supports placing of notch

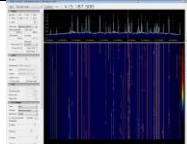
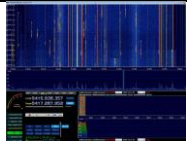
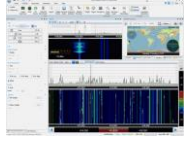
filters either manually or automatically. There are also noise reduction and noise blanker features and an automatic frequency centering algorithm which will automatically center the signal, avoiding the need of clicking exactly in the center of a signal. Traditional amateur radio operators also enjoy the S-units signal strength meter and the built in frequency manager.

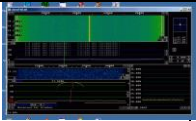

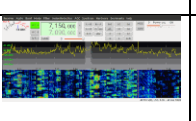
3. **SDR-RADIO.COM V2.** It is a popular SDR program with many advanced features. As such is it a fair amount more difficult to learn and use compared to SDR# and HDSDR.
4. **Linrad.** It is a free advanced SDR program with a steeper learning curve compared to most other SDR programs. Many users prefer to use Linrad as it can be set up to have a very high amount of information density on the screen and it also has an advantage with receiving signals with the E4000 tuner as it can be made it run in a special linearity mode which significantly improves the dynamic range.
5. **Studio1.** Is a non-free commercial SDR receiver with advanced DSP capabilities, and is claimed to have the lowest CPU usage of any general purpose SDR software. Studio1 supports the RTL-SDR through use of an ExtIO.dll module. Studio1's main selling point is that it claims that its spur/alias free dynamic range is greater than 130 dB compared to the 70 dB that most free SDR software programs have (spurious free dynamic range influences the capability of the receiver to resolve small signals in presence of in-band strong signals). It also claims that it has a very efficient DSP engine that can run well on older Pentium 4 machines.
6. **SDR Touch.** This software was the first Android based RTL-SDR software receiver available. There is a free restricted trial version and the full version can be bought from the Google Play store. To run SDR Touch it is necessary a modern Android 4.0+ device with decent processing power and USB OTG support. It will also need a USB OTG

cable to connect to the Android device to an RTL-SDR dongle. SDR Touch has several standard features such as FFT spectrum and waterfall displays, WFM/FM/AM/SSB tuning and a frequency manager.

7. **PowerSDR.** This is an SDR receiver intended for use with the FlexRadio SDR hardware. It is compatible with the RTL-SDR through use of the RTL_HPSDR translation server program. PowerSDR is capable of supporting up to four simultaneous RTL-SDR dongles.
8. **QtRadio.** Is an open source SDR receiver that is based on a client-server architecture from the ground up. It is designed so that multiple clients can connect to a single server. QtRadio comes with a special RTL-SDR server which is necessary download and run.
9. **SeeDer.** Is a general purpose SDR GUI that has a focus on decoding data from satellites. Legality Note: SeeDeR may be an illegal clone of SDR#.

Table B.2 - Comparisons with other common General Purpose RTL-SDR

| SDR | Image | Free? | OS | Web site |
|------------------|---|-------|---------|---|
| SDR# |  | Free | Windows | http://airspy.com/ |
| HDSDR |  | Free | Windows | http://www.hdsdr.de/ |
| SDR-RADIO.COM V2 |  | Free | Windows | http://v2.sdr-radio.com/ |

| | | | | |
|-----------|---|---------------|-----------------------|---|
| Linrad |  | Free | Windows/Linux/ Mac | http://www.sm5bsz.com/linuxdsp/linrad.htm |
| Studio1 |  | Paid | Windows | http://www.sdrapplications.it/ |
| SDR Touch |  | Trial Paid | Android | https://play.google.com/store/apps/details?id=marto.androsdr2&hl=en |
| PowerSDR |  | Free | Windows | http://www.flexradio.com/support/ |
| QtRadio |  | Free | Windows Linux | http://napan.ca/ghpsdr3/index.php/QtRadio_Installation |
| SeeDeR |  | Free | Windows | http://www.southernstars.com/skycube/seeder.html |

Source: Author

According to the research, SDR # and HDSDR is the most used software by amateur radio community.

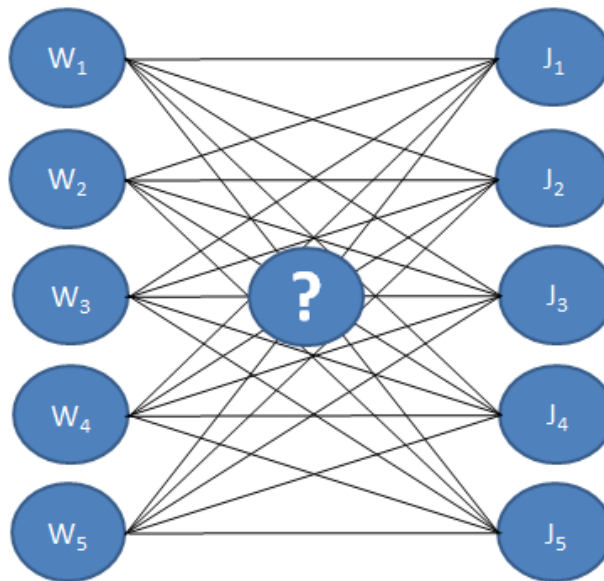
APPENDIX C - HUNGARIAN ALGORITHM

C.1. Concept

The Hungarian method, created by Hungarian mathematicians Dénes Kőnig and Jenő Egerváry, is formally a multi-step procedure for applying its theorem given a cost-matrix of order n and obtaining another matrix with nonnegative entries containing an assignment consisting entirely of zeros. (Kuhn, 1955).

A classical problem solved by the Hungarian algorithm is the one when it is necessary to assign many jobs to many workers where it is required to optimize the minimum time to accomplish the job. Figure C.1 shows the depicted situation. The solution of the problem consists of optimizing the time so that only one worker can do one job, but which one?

Figure C.1 - A classical problem solved by the Hungarian Algorithm



Source: Author

This problem uses a Hungarian algorithm where the following theorem is proposed: *“If a number is added to or subtracted from all entries in a row or*

column of a cost matrix, then an optimal assignment of tasks to the resulting cost-matrix is also an optimal task assignment for the original cost matrix”.

The Hungarian algorithm consists of the four steps below. The first two steps are executed once, while Steps 3 and 4 are repeated until an optimal assignment is found. The input of the algorithm is an n by n square matrix with only non-negative elements.

The steps are¹⁵:

Step 1: Subtract row minima. For each row, find the lowest element and subtract it from each element in that row.

Step 2: Subtract column minima. Similarly, for each column, find the lowest element and subtract it from each element in that column.

Step 3: Cover all zeros with a minimum number of lines. Cover all zeros in the resulting matrix using a minimum number of horizontal and vertical lines. If n lines are required, an optimal assignment exists among the zeros. The algorithm stops.

If less than n lines are required, continue with Step 4.

Step 4: Create additional zeros. Find the smallest element (call it k) that is not covered by a line in Step 3. Subtract k from all uncovered elements, and add k to all elements that are covered twice.

Some important observations:

- The cost matrix must be square. When it does not happen, a fictitious row or column of zeros must be entered.

¹⁵ Available on: <<http://www.hungarianalgorithm.com/>>. Accessed on: 20 Oct. 2016

- Cost matrix inputs must be integers. When it does not happen, the matrix-cost must be multiplied by a suitable power of ten. The problem must be minimization.
- The problem of maximizing the sum of inputs of a cost matrix is easily converted into a problem of minimizing the sum of inputs by multiplying each input of the matrix by -1.

C.2. Result

This section content the result of the running the Hungarian Algorithm online from, with the following results:

Fill the matrix with the problem of cost allocation, complete with 10 cells where not applicable, and 100 cells where no values in the case of the satellite 4 and 5 satellite. The optimal assignment will be determined and a step by step explanation of the Adapted Hungarian Algorithm will be given.

This is the original cost matrix (table C.1):

Table C.1 - Original matrix

| | | | | |
|----|----|----|-----|-----|
| 1 | 10 | 3 | 100 | 100 |
| 1 | 2 | 3 | 100 | 100 |
| 10 | 2 | 10 | 100 | 100 |
| 1 | 2 | 10 | 100 | 100 |
| 10 | 10 | 10 | 100 | 100 |

Source: Author

Subtract row minima. In the table C.2 we subtract the row minimum from each row:

Table C.2 - Subtract row minima

| | | | | | |
|---|---|---|----|----|-------|
| 0 | 9 | 2 | 99 | 99 | (-1) |
| 0 | 1 | 2 | 99 | 99 | (-1) |
| 8 | 0 | 8 | 98 | 98 | (-2) |
| 0 | 1 | 9 | 99 | 99 | (-1) |
| 0 | 0 | 0 | 90 | 90 | (-10) |

Source: Author

Subtract column minima. In the table C.3 we subtract the column minimum from each column:

Table C.3 - Subtract column minima

| | | | | |
|---|---|---|-------|-------|
| 0 | 9 | 2 | 9 | 9 |
| 0 | 1 | 2 | 9 | 9 |
| 8 | 0 | 8 | 8 | 8 |
| 0 | 1 | 9 | 9 | 9 |
| 0 | 0 | 0 | 0 | 0 |
| | | | (-90) | (-90) |

Source: Author

Cover all zeros with a minimum number of lines. There are 3 lines required to cover all zeros (table C.4):

Table C.4 - Cover all zeros

| | | | | | |
|---|---|---|---|---|---|
| 0 | 9 | 2 | 9 | 9 | |
| 0 | 1 | 2 | 9 | 9 | |
| 8 | 0 | 8 | 8 | 8 | |
| 0 | 1 | 9 | 9 | 9 | x |
| 0 | 0 | 0 | 0 | 0 | x |
| x | | | | | |

Source: Author

Create additional zeros. The number of lines is smaller than 5. The smallest uncovered number is 1. We subtract this number from all uncovered elements and add it to all elements that are covered twice (see table C.5):

Table C.5 - Create additional zeros (a)

| | | | | |
|---|---|---|---|---|
| 0 | 8 | 1 | 8 | 8 |
| 0 | 0 | 1 | 8 | 8 |
| 9 | 0 | 8 | 8 | 8 |
| 0 | 0 | 8 | 8 | 8 |
| 1 | 0 | 0 | 0 | 0 |

Source: Author

Cover all zeros with a minimum number of lines. There are 3 lines required to cover all zeros (table C.6):

Table C.6 - Cover all zeros with a minimum number of lines (a)

| | | | | |
|---|---|---|---|---|
| 0 | 8 | 1 | 8 | 8 |
| 0 | 0 | 1 | 8 | 8 |
| 9 | 0 | 8 | 8 | 8 |
| 0 | 0 | 8 | 8 | 8 |
| 1 | 0 | 0 | 0 | 0 |
| x | x | | | |

Source: Author

Create additional zeros. The number of lines is smaller than 5. The smallest uncovered number is 1. We subtract this number from all uncovered elements and add it to all elements that are covered twice (see table C.7):

Table C.7 - Create additional zeros (b)

| | | | | |
|---|---|---|---|---|
| 0 | 8 | 0 | 7 | 7 |
| 0 | 0 | 0 | 7 | 7 |
| 9 | 0 | 7 | 7 | 7 |
| 0 | 0 | 7 | 7 | 7 |
| 2 | 1 | 0 | 0 | 0 |

Source: Author

Cover all zeros with a minimum number of lines. There are 4 lines required to cover all zeros (table C.8):

Table C.8 - Cover all zeros with a minimum number of lines (b)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 0 | 7 | 7 | |
| 0 | 0 | 0 | 7 | 7 | |
| 9 | 0 | 7 | 7 | 7 | |
| 0 | 0 | 7 | 7 | 7 | |
| 2 | 1 | 0 | 0 | 0 | x |
| x | x | x | | | |

Source: Author

Create additional zeros. The number of lines is smaller than 5. The smallest uncovered number is 7. We subtract this number from all uncovered elements and add it to all elements that are covered twice (table C.9):

Table C.9 - Create additional zeros (c)

| | | | | |
|---|---|---|---|---|
| 0 | 8 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 7 | 0 | 0 |
| 0 | 0 | 7 | 0 | 0 |
| 9 | 8 | 7 | 0 | 0 |

Source: Author

Cover all zeros with a minimum number of lines. There are 5 lines required to cover all zeros:

Table C.10 - Cover all zeros with a minimum number of lines (c)

| | | | | | |
|---|---|---|---|---|---|
| 0 | 8 | 0 | 0 | 0 | x |
| 0 | 0 | 0 | 0 | 0 | x |
| 9 | 0 | 7 | 0 | 0 | x |
| 0 | 0 | 7 | 0 | 0 | x |
| 9 | 8 | 0 | 0 | 0 | x |

Source: Author

The optimal assignment. Because there are 5 lines required, the zeros cover an optimal assignment (table C.11):

Table C.11 - The optimal assignment

| | | | | |
|---|---|---|---|---|
| 0 | 8 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 7 | 0 | 0 |
| 0 | 0 | 7 | 0 | 0 |
| 9 | 8 | 7 | 0 | 0 |

Source: Author

The table C.12 corresponds to the following optimal assignment in the original cost matrix:

Table C.12 - The optimal assignment, original matrix

| | | | | |
|----|----|----|-----|-----|
| 1 | 10 | 3 | 100 | 100 |
| 1 | 2 | 3 | 100 | 100 |
| 10 | 2 | 10 | 100 | 100 |
| 1 | 2 | 10 | 100 | 100 |
| 10 | 10 | 10 | 100 | 100 |

Source: Author

APPENDIX D - BRIEF CONCEPTS ON ARTIFICIAL INTELLIGENCE

Bellow there are some of the artificial intelligence techniques that were studied in the preparation of this proposal:

D.1. Rational Agent

From a functional point of view, it can be established that "An agent is anything that can perceive its environment with the help of sensors and act upon that environment using actuators" (RUSSELL; NORVIG, 2004).

Thus, the behavior of the agent is given by the function of the agent which projects an action, with a given perception. This function is implemented by the agent program.

The purpose of a rational agent is doing the right thing, or in other words, does your best. Then, it is necessary to include a way to measure the success or failure of an agent. This way of measuring the success of an agent is called performance measures, which include the criteria that determine success in the agent's behavior.

According to Russell and Norvig (2004) a rational agent is defined as: are computational artifacts designed to build and maintain long-term, social-emotional relationships with their users. Central to the notion of relationship is that it is a persistent construct, spanning multiple interactions, thus Relational Agents are explicitly designed to remember past history and manage future expectations in their interactions with users. The above definition implies that the rational agent seeks to maximize the expected return (rationality) and not the actual result (omniscience). Similarly entails that the agent perform actions in order to modify future perceptions (information gathering) to learn as much as possible of the environment performance, thus making it independent of initial knowledge provided by the agent designer. (WASSERMANN, 1985)

The duty of Artificial Intelligence is to design the agent program that implements the agent function, which projects perceptions in actions. It is assumed that this program will run on any computer with physical sensors and actuators, this is known as architecture. According to this: Agent = architecture + program (RUSSELL; NORVIG, 2004).

Obviously, the chosen program must be appropriate to the architecture. Overall, the architecture is in charge of making that perceptions of sensors available for the program, implement programs, and have to put in place the actuators generated actions. (NIKOLOPOULOS, 1997)

D.2. Neural Networks

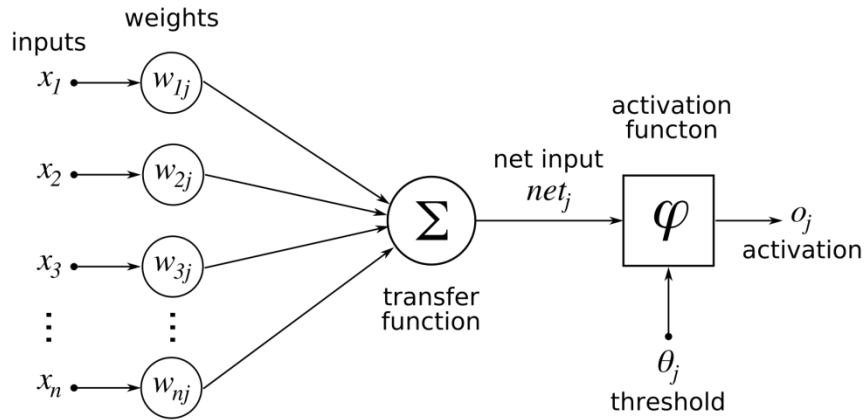
An Artificial Neural Network (ANN) is a computational model inspired by biological neural networks which can be considered as a data processing system with features like learning through examples, adaptability, robustness, capability of generalization and fault tolerance.

The ANN can be defined as a distributed structure with parallel processing, made of artificial neurons (also called processing elements), interconnected by a large number of connections (synapses), which are used to store knowledge that is available to be used (HERTZ et al., 1991).

Structure of Artificial Neuron

An artificial neuron is an information processing unit of neural networks. Figure D.1 shows the neuron model known as McCulloch-Pitts.

Figure D.1 - Artificial Neuron



Source: (HERTZ et al., 1991)

It can be seen that N input signals are represented by the variables $x_1, x_2 \dots x_N$ which are associated with weights that are represented by w_{ij} variables which determine the level of influence of the neuron j to neuron i .

There are two processing steps for each neuron: addition and activation.

- Addition. Input signals x_j and weights w_{ij} are combined by summation:

$$y_i = \sum_{j=1}^N w_{ij} x_j$$

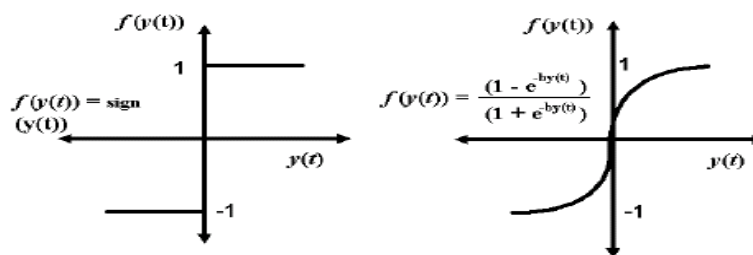
Where y_i is called the internal state of the neuron i . $x_i = f(x_i)$

- Activation. The neuron output is generated through the application of an activation function call, where the output of the neuron is represented by $f(x_i)$ corresponds to the activation function applied to the internal state of the neuron, which aims to limit the level of activation between $[-1, 1]$ or $[0, 1]$, if x_i is a continuous value and if x_i is discrete then it can be:

$$\{-1,1\} \text{ o } \{0,1\}$$

Two of the most used activation functions are shown in Figure D.2 the degree function and the hyperbolic tangent. As seen in the first figure the output of a neuron can be the input of another. Generally, a neural network is formed by many neurons in some coupled manner.

Figure D.2 - Activation Functions



Source: (HERTZ et al., 1991)

Back-propagation neural networks

The method of back-propagation (backward or training) is an automated system for training neural networks with hidden layers, perfected in the 80's (BROWN; ROTHARY, 1993).

When training such networks, the problem is that they only know the network output and input, so an adjustment to the synaptic weights associated with the hidden layer neurons cannot be done, since it can't be inferred from the state of the output layer as it should be done from the state of the hidden layers.

The training system using back-propagation:

- Begins with either synaptic weights (usually chosen at random).
- Enters input data (the input layer) randomly chosen from the input data to be used for training.

- Allows the network data to generate an output vector (forward propagation).
- Compares the output generated by the network with the desired output.

The difference obtained between the generated output and the desired (named error) is used to adjust the synaptic weights of the neurons of the output layer.

The error is propagated backwards (back-propagation) to the previous layer neurons, and is used to adjust the synaptic weights in this layer until the input layer is reached. This process is repeated with different training data.

The method by which the error is propagated backwards was discovered by DE Rumelhart, G. E. Hinton and R.J. Williams in 1986, and almost at the same time by Parker (1985) and Le Cun (1986) (ALONSO; BECERRIL, 1993).

Neural networks applications.

The characteristics of neural networks make possible a wide variety of applications. Some of the most prominent applications are:

- Pattern recognition classification.
- Voice recognition of handwritten characters.
- Analysis and recognition of images, shapes.
- Clinical diagnosis.
- Time series analysis and prediction:
- Weather models.
- Behavior prediction.
- Forecasting Time Series.

D.3. Genetic Algorithms

According to Russel and Norvig (2004) on their book about Artificial Intelligence, they make a state of the art about genetic algorithms and start talking about the

work from Sewall Wright (1931) about the concept of the suitability of a landscape, this was an important precursor to the development of genetic algorithms. In the 50's, some statisticians, including Box (1957) and Friedman (1959), used evolutionary techniques to optimization problems, but it wasn't until Rechenberg (1965, 1973) introduced the evolutionary strategies to solve optimization problem to airfoils where this approach became popular. In the 60's and 70's John Holland (1975) defended the genetic algorithms as a useful instrument and as a method to broaden our understanding of biological and some other way of adaptation (Holland, 1995).

The artificial life movement (Langton, 1995) takes this idea a little further by considering the products of genetic algorithms as organisms rather than problem solutions. The work of Hinton and Nowlan (1987) and Ackley and Littman (1991) in this field have done much to clarify the implications of Baldwin effect. To a deeper and more general approach to evolution Smith and Szathmáry (1999) is recommended.

It can be said that genetic algorithms are search algorithms based on natural selection and natural genetics mechanisms. They combine the survival of the more compatible ones between the chain structures, with an information structure already randomized and exchanged to build a search algorithm with some of the innovation capacities of human search (GOLDBERG, 1989).

A genetic algorithm is a variation of the stochastic beam search where the successor states are generated by combining two parent states rather than modifying one state.

According to Banzhaf and Reeves (1999), the simplest form of a genetic algorithm uses three types of operators: selection, crossing and mutation.

- Selection or reproduction: This operator chooses chromosomes among the population to make reproduction. The more able chromosome, will be selected to reproduce.

- Crossing: Is about an operator whose work is to choose a place and change the sequences before and after that position between two chromosomes, to create new offspring (e.g. strings 10010011 and 11111010 can be crossed after the third place to produce the offspring 10011010 and 11110011). It imitates the biological recombination between two haploid organisms.
- Mutation: This operator produces random variations in a chromosome (e.g. the string 00011100 can mutate its second position to lead to string 01011100). Mutation can be given in each bit position of a string, with a normally very small probability (for instance 0.001).

According to Mitchell (1998), a genetic algorithm differs from traditional search and optimization methods in four essential issues:

1. It works with a code of the parameters set, not with the set itself (they require that the parameters set of the optimization problem is coded in finite string upon a determined alphabet). By working in code level, and not with the functions and control variables like the other methods, it is harder to be “tricked”.
2. It looks for a point population, not a unique point. By keeping a well-adapted population of sample points, the probability to fall into a false top is reduced.
3. It uses the objective function, does not need derivatives or any other complementary information, so hard sometimes to get. Thus they gain in efficiency and generality.
4. It makes use of stochastic transition rules, not deterministic. Genetic algorithms make use of random operator to guide the search for the best points; it might look strange, but nature is full of precedents on behalf.

Genetic algorithms have had an extended impact on optimization problems like circuit arrangement and scheduled work in shops.

D.4. Fuzzy logic

According to Fuller (1995) fuzzy sets were introduced by Zaden in 1965 as a kind of representation and manipulation of fuzzy data instead of precise one. This logic offers a type of inference that allows the human being to think in a different way and thus making knowledge based systems.

Fuzzy theory uses the thinking and the reasoning to stay away from math and captures the uncertainties associated to human cognitive processes, due to the lack of conventional ways to represent knowledge since the concepts are fuzzy. As a consequence, there is no existence of adequate conceptual framework to represent knowledge, that as known, it is lexical, unprecise and cannot be categorized.

The entire development of fuzzy logic was motivated to a large degree to the necessity of a conceptual framework to approach the matter of uncertainty and lexical imprecision.

Some of the essential features of fuzzy logic are referred to: (MACHADO; ROCHA, 1992)

- The exact reasoning is seen like a limit case of approximate reasoning.
- In fuzzy logic, is all about degree.
- The knowledge is interpreted as an elastic collection, in an equivalent way.
- Inference is seen as an elastic restrictions propagation process.
- Any logical system can become a fuzzy one.

There are two main features of fuzzy systems that give them a better performance for specific applications: (FULLER, 1995)

- Fuzzy systems are adequate for uncertain or approximate reasoning, especially for the system with a difficult derivative mathematical model.
- Fuzzy logic allows decision-taking with estimated values that can be incomplete or with uncertain information.

D.5. Knowledge Based System.

According to Bernard (1998), it can be known under the names of Knowledge Based Systems (Knowledge Based Systems, KBS) or Expert Systems SE (Expert Systems), computer programs that:

- Possess knowledge of a human expert or a particular area of knowledge.
- They are able to provide recommendations inferred from their knowledge.
- They can justify their own conclusions.
- They have no knowledge implicitly encoded within the program.

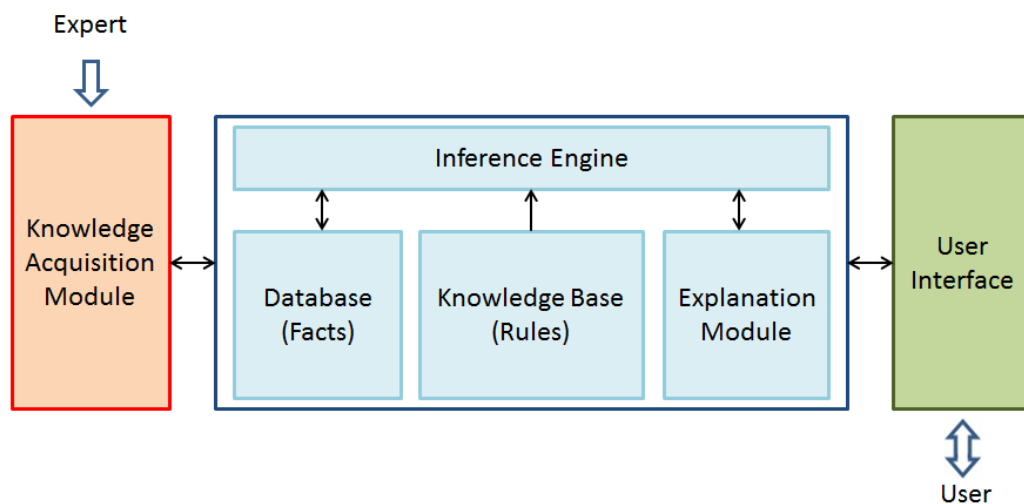
Expert systems are artificial intelligence technique first used in systems analysis of credit risk. However, many expert systems are in full force and service in various institutions, and their chances have been enriched with new approaches that have contributed mostly to other emerging techniques, as induction systems (LIAO, 2005).

Rule-based expert system structure:

As noted by several authors (MISHKOFF, 1988; HARMON; KING, 1988), due to the great diversity of rule-based expert systems it can't be talked of a single structure.

A rule-based system must have a knowledge base, a fact base, an inference engine, an explanation subsystem and a user interface. Figure D.3 shows some of the components of a rule-based system.

Figure D.3 - Rule-Based System



Source: Adapted from Pelaéz; Velazco (2016)

However, in most of them it is possible to identify the following basic components: (NEGNEVITSKY, 2011)

- Knowledge Base: Contains information about the knowledge domain that is referred to the expert system. Inside it can distinguish between declarative knowledge (facts) and procedural (rules).
- Database, Report or Model Situation: An auxiliary memory containing information about the problem to solve (baseline) and system status along the inference process (intermediate data).
- Inference engine: The inference engine or control structure is part of the expert system that is responsible for performing inference processes that relate the information in working memory with the knowledge base, in order to reach conclusions. Every rule-based system must perform three

operations: 1. recognize what the rules are, 2. decide which rules to implement, and 3. enforce the chosen rules (PALAZÓN, 1991).

- **User Interface:** The user interface and query subsystem is the part of the system that enables communication between the user and the inference engine. It enter/receives the information needed by the system and the responses to inform the user of the expert system.
- **Module of Justification or Explanation subsystem:** The part of the system that explains the steps performed by the inference engine to reach conclusions, indicates why the system to uses certain rules and not others, and why certain questions were raised during the dialogue with the user.
- **Knowledge acquisition subsystem:** It is an interface that facilitates the introduction of the knowledge base and inference mechanisms in the inference engine. It also checks the accuracy and consistency of facts and rules that are introduced into the knowledge base.

Basing on the literature review and as a summary, Table D.1 shows a description of the advantages and disadvantages of commonly used techniques.

Table D.1 - Artificial Intelligence Techniques

| Technical Artificial Intelligence | Advantages | Disadvantages |
|--|---|---|
| Rational Agents | <p>One of the important features of an agent is the flexibility it offers to act with its own environment and meet its goals.</p> <p>Inside the flexibility an agent can offer, it is appropriate to highlight its reactivity, which is the responsiveness of the changes</p> | <p>Environmental control, because as a rule, will never have total control of the environment; having in mind that the same action may not always have the same result.</p> <p>The agent can make mistakes, so they should be prepared to identify them and recover from them and</p> |

| | | |
|--------------------------|---|--|
| | detected; and its social ability, which is the ability to interact with other agents to accomplish their goals. | last but not least, an agent must make decisions based on its goal. |
| Neural networks | <p>Learning. Neural network have the ability to learn through a phase called learning. This provides the ANN input data while indicating which is the output (response) expected.</p> <p>Self-organization. An ANN creates its own representation of information inside, freeing the user of this task.</p> <p>Availability of increasingly powerful development tools.</p> | <p>Show weakness when it's needed to predict random or semi-random numbers.</p> <p>Have difficulty with the processing of large integers. For example, determine whether a long integer is prime or composite.</p> <p>Not have the ability to decrypt some good algorithm encrypted data.</p> |
| Genetic Algorithms (GA). | <p>Not need specific knowledge of the problem it attempts to solve.</p> <p>Operate simultaneously with several solutions instead of sequentially working with traditional techniques.</p> <p>When used for optimization problems (maximizing an objective function) they are less affected by local maxima (false solutions) than traditional techniques.</p> <p>It is extremely easy to run them on modern massively parallel architectures.</p> | <p>Use probabilistic operators instead of the typical deterministic operators of other techniques.</p> <p>May take some time to converge or not converge at all, depending on the parameters used (population size, number of generations, etc.).</p> <p>May converge prematurely due to a number of different problems.</p> |
| Fuzzy logic | The main advantage is the excellent results of a fuzzy logic based control system, provides quick and accurate outputs, thereby reducing transitions of key states which controls the | <p>Facing with a problem that can be solved by a mathematical model, we obtain worse results using fuzzy logic.</p> <p>It has the disadvantage that the expert or the technology can modify</p> |

| | | |
|------------------------|---|---|
| | <p>physical environment.</p> <p>Ease of implementation.</p> <p>Good results in nonlinear and difficult modeling processes.</p> <p>Mode of operation similar to human behavior.</p> <p>No need to know the mathematical model that governs its operation.</p> | <p>information. Thus the technology (mainly through neural networks) can reinforce initial heuristic rules of any control system based on this kind of logic.</p> |
| Knowledge Based System | <p>Ability of making decisions based on the knowledge stored.</p> <p>Low cost. Even if the initial cost is high, for ease of duplication at the end becomes an economic system.</p> <p>Duplicity. The expert system are programs that once installed on a computer, run a series of functions that can be copied and put on other machines.</p> <p>Quickly. An expert system has the capability of processing in less time than a human expert, a great variety of information.</p> | <p>Need for training for its management.</p> <p>The common sense. For an expert system nothing is obvious, everything must have structural connotation, because if it does not run that way, the answers will not be the most appropriate.</p> <p>Rigidity. An Expert System is extremely rigid when receiving information due to the structure of its programming.</p> |

Source: Author

APPENDIX E - EXPERT SINTA

In this appendix is detailed the way Expert SINTA performs each one of the tasks needed to create a rule-based expert system. This appendix is based on the user's guide of Expert SINTA.

Expert SINTA is software that uses artificial intelligence techniques in order to achieve automated generation of rule-based expert systems. This software uses a representational model of knowledge based on production and probabilities rules, aiming to simplify the work to implement expert systems using: a shared inference engine, automated generation of windows and menus, probabilistic treatment of production rules and the use of sensitive explanations to the context of the modeled knowledge base (Expert SINTA, 1996).

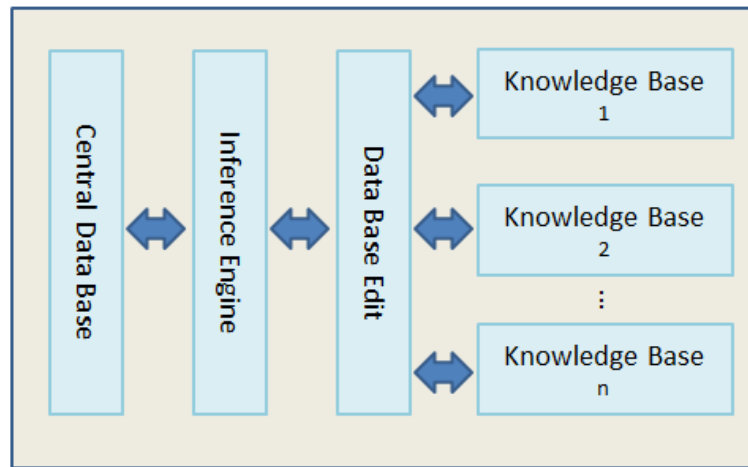
The user must interact to a menu sequence, and then the system is responsible to offer answers towards the windows pointed by the user. This system was developed in the Artificial Intelligence Lab of *Universidade Federal do Ceará – UFC*, Brazil.

Right below is explained Expert SINTA inner operation, the way knowledge is represented and probabilistic usage in the inference engine.

E.1. Expert SINTA Architecture

Expert systems generated with Expert SINTA follow the architecture observed in Figure E.1

Figure E.1 - Expert SINTA Simplified Architecture



Source: Adapted from Expert SINTA (1996)

In this architecture, the following items are shown:

Knowledge Base: It represents the information (acts and rules) that a specialized system uses, represented in computational forms.

Database Editor: Through this one, the Shell allows the implementation of desired knowledge bases.

Inference Engine: It's the part of the expert system responsible for deduction on the knowledge base.

Central Database: These are the evidences pointed by the user of the expert system during a query.

The goal of Expert SINTA is to simplify to the max the creational stages of a complete expert system. Therefore, it already offers a basic inference engine mainly relying on back chaining.

Expert SINTA was created in order to allow to the knowledge analyst to implement the desired base. The user of Expert SINTA doesn't require much

programming knowledge, only needs to know how to interact with visual environment.

A knowledge base in Expert SINTA involves the following attributes that must be declared by the creator of the knowledge base:

- Variables
- Rules
- Queries
- Objectives
- Additional information

E.2. Managing bases

Expert SINTA keeps track of knowledge bases generated, in *.BCM files. To open one of the existent bases, use the menu *Arquivo/Abrir*. To create a new base, use the menu *Arquivo/Novo*.

Use the menu *Arquivo/Salvar* to save the base every time needed. If desired it can be printed using the menu *Arquivo/Imprimir*. Figure E.2 shows the tool bar highlighting the available operations with the bases.

Figure E.2 - Tool bar for base-making

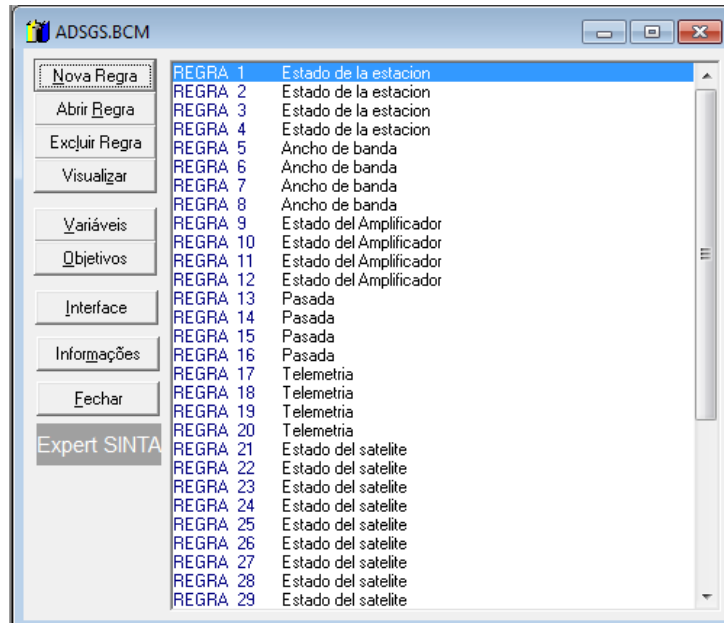


Source: Adapted from Expert SINTA (1996)

E.3. “Knowledge-in-a-box” (KIB) window

This window contains every edition options of the base, besides of a list with every existent rule. Among the options: new rule, open rule, exclude rule, visualize, variables, objectives, interface, information and close. Figure E.3 shows KIB window.

Figure E.3 - KIB window



Source: Generated by Expert SINTA

Expert SINTA also shows in the upper part the menu bar with every selectable option. (See Figure E.4)

Figure E.4 - Tools bar of Expert SINTA

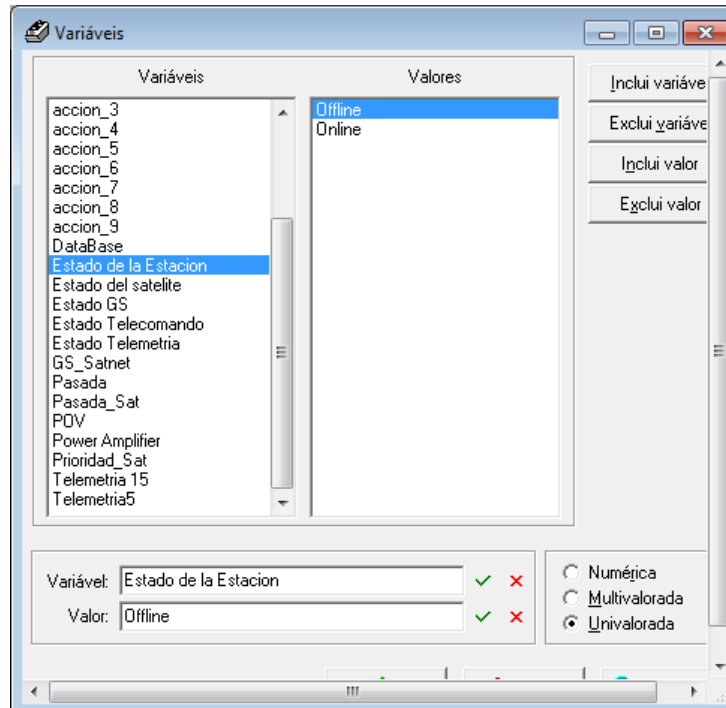


Source: Generated by Expert SINTA

E.4. Creating Variables

To create new rules, first it is necessary to create the variables with their correspondent values. A base can be organized through this mechanism, allowing easy maintenance and rule creation visually. Figure E.5 shows the variable creation window and initializing its values. The variables can be numeric, multi o univalued. This window allows to include/exclude variables and values.

Figure E.5 - Expert SINTA Simplified Architecture



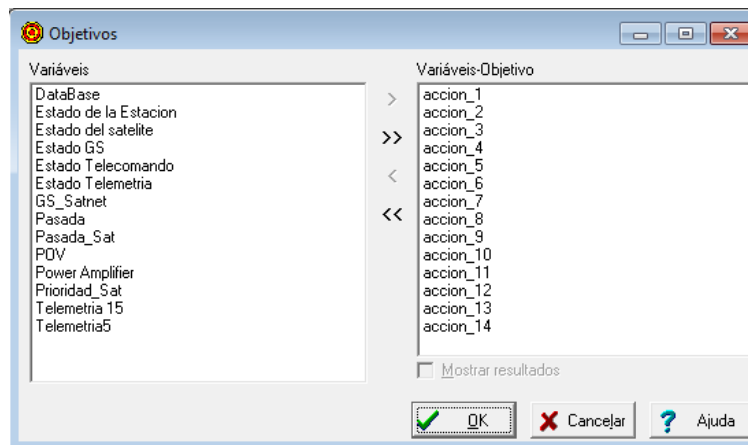
Source: Generated by Expert SINTA

E.5. Objectives definition

The objective of a specialist query is to find an answer for a particular problem. The expert system must also have well defined objectives. The difference is that in Expert SINTA the problems are represented by variables which should be defined to control the way the inference engine works.

Figure E.6 shows the window where variables are defined and then will be objectives. The system allows to include/exclude variables.

Figure E.6 - Objectives



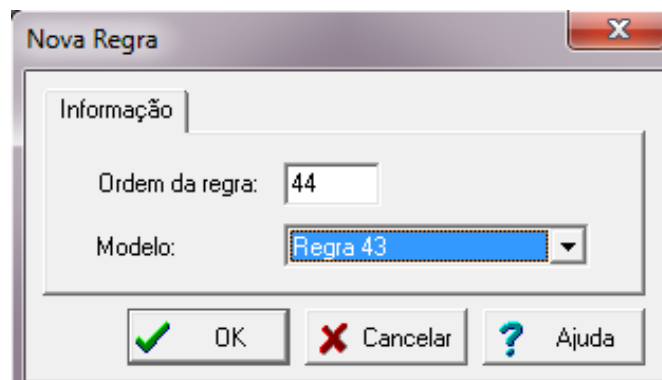
Source: Generated by Expert SINTA

E.6. Working with rules

Expert SINTA uses production rules to model human knowledge, turning ideal for selection problems where a particular solution should be achieved from a selections set.

Figure E.7 shows the way to create a new rule, which will have an order number affecting the system behavior. New rules can be created basing on previously created rules.

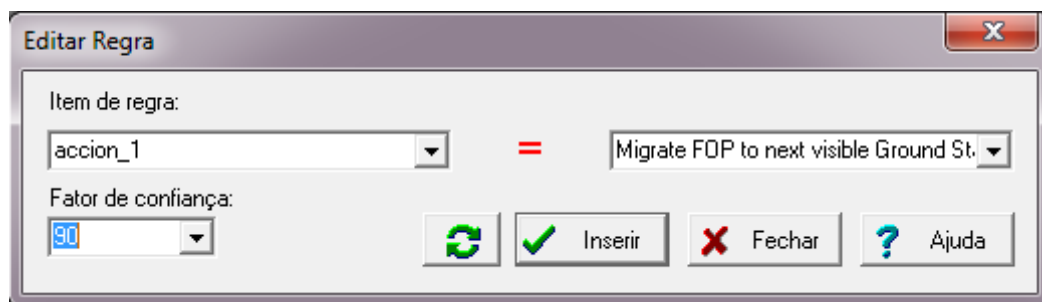
Figure E.7 - New rule creation



Source: Generated by Expert SINTA

When a new rule is created, Expert SINTA displays the edition window (see Fig E.8). The first list indicates the variable to be used; the last one indicates the value of a relation established. It is possible to identify the relation type: equality (“=”), difference (“<>”), superiority (“>” – only numeric variables), among others. In the case where the premise is not the first rule, the connector type (AND or OR) should be indicated; then the premise concatenates with its previous.

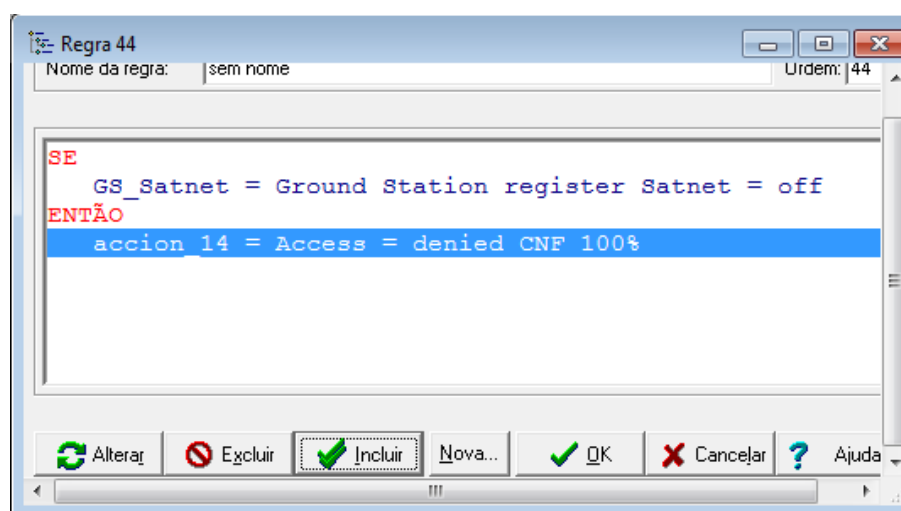
Figure E.8 - Rule Edition Window



Source: Generated by Expert SINTA

When hitting Inserir button, the systems shows the rule window (see Fig E.9). Here the rule uses the IF...THEN... format, the variable definition given and the correspondent degree of trust.

Figure E.9 - Rule window



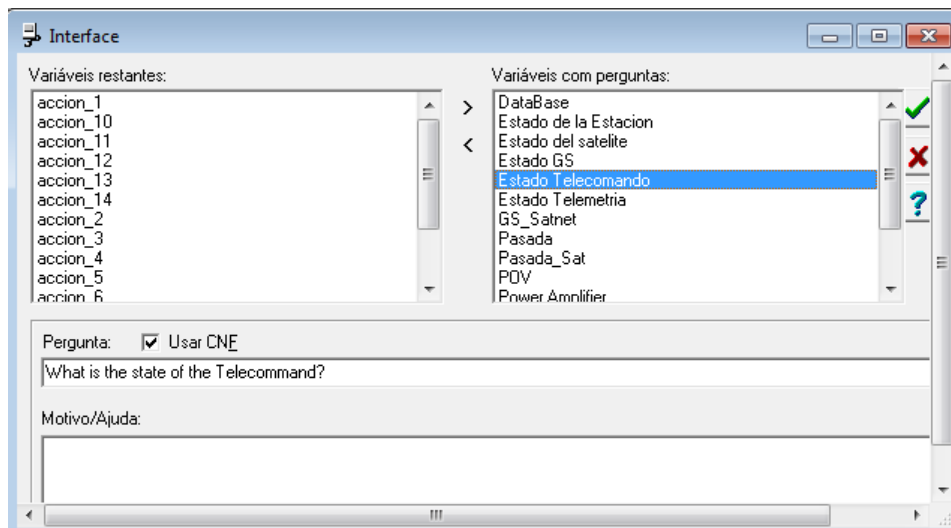
Source: Generated by Expert SINTA

E.7. Interface

An expert system implementing Expert SINTA interacts with the final user through multiple choices menus (or single option menu if the variable is univalued). These menus are automatically built by the Shell, but some details should be provided by the knowledge base creator.

A part of these details are the questions the system should do to the final user, Figure E.10 shows the window where the knowledge base creator establishes the questions for each variable.

Figure E.10 - Interface window



Source: Generated by Expert SINTA

E.8. Executing the system

There are two ways to perform a query in Expert SINTA, using the menu in the main screen or in the tool bar. To start a query, press the OK button. The system will display questions that the user should answer one by one; the degree of trust can be modified at any time, and if there is no absolute certainty about a fact it is displayed the default degree of trust.

Figure E.11 shows a sample question made by the system, if it is left unanswered the system understands that the user doesn't know the answer to that question.

Figure E.11 - Sample question

ADSGS

Which is the state of the Ground Station now?
(Marque somente uma alternativa)

Opção: _____ Grau de Confiança %:

☐ Offline

☒ Online

100

OK Por que?

Source: Generated by Expert SINTA

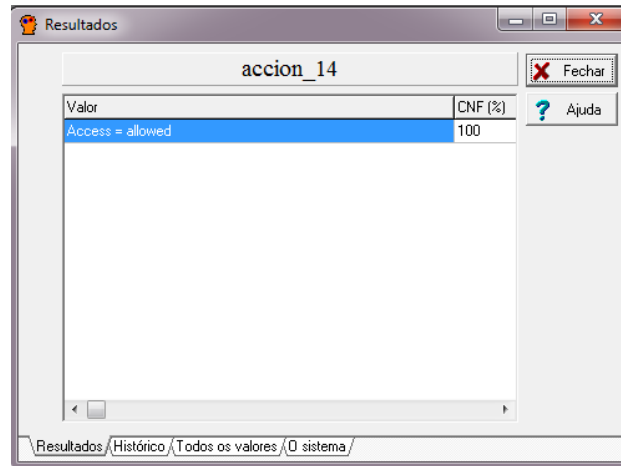
Choose the desired option with the checkboxes located on the left. Each answer can have its own degree of trust.

When the system finishes with the questions, it can be observed the query results, the record, all the values and the system. Each one of these are detailed below.

E.9. Results

After the last question, the system displays the results that would be presented according to the values reached by the objective, with the correspondent degrees of trust. In the case where the system didn't reach a solution, that message will be displayed. Figure E.12 shows the results given to a specific question.

Figure E.12 - Results window

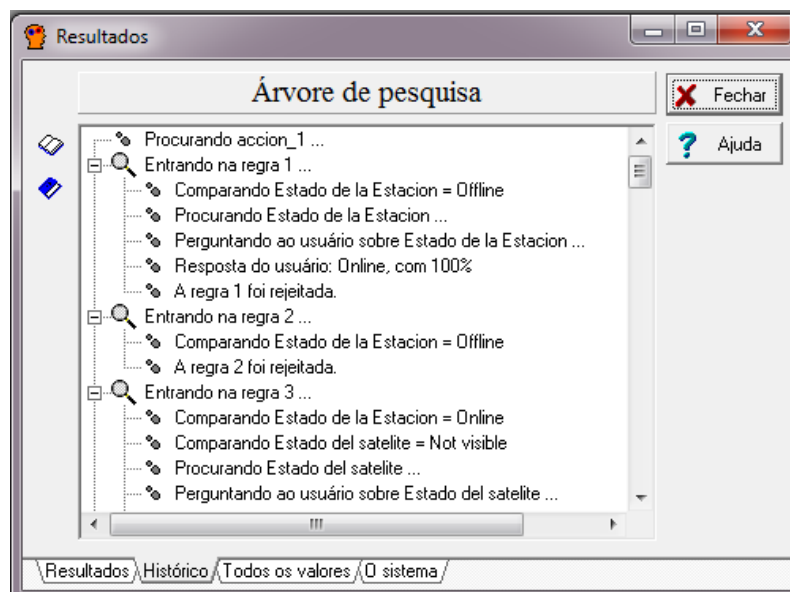


Source: Generated by Expert SINTA

E.10. Record

This window shows the path made by the expert system to get to the solution, every interaction performed by the system to get to a specific conclusion. The list can be displayed from up to down and vice versa. Figure E.12 shows the record window.

Figure E.12 - Record window



Source: Generated by Expert SINTA

E.12. All the values

This window displays every value for every variable, the system allows to show or hide the values for every variable; this option is only available if established by the knowledge base creator. Figure E.14 shows all the results.

Figure E.14 - All results

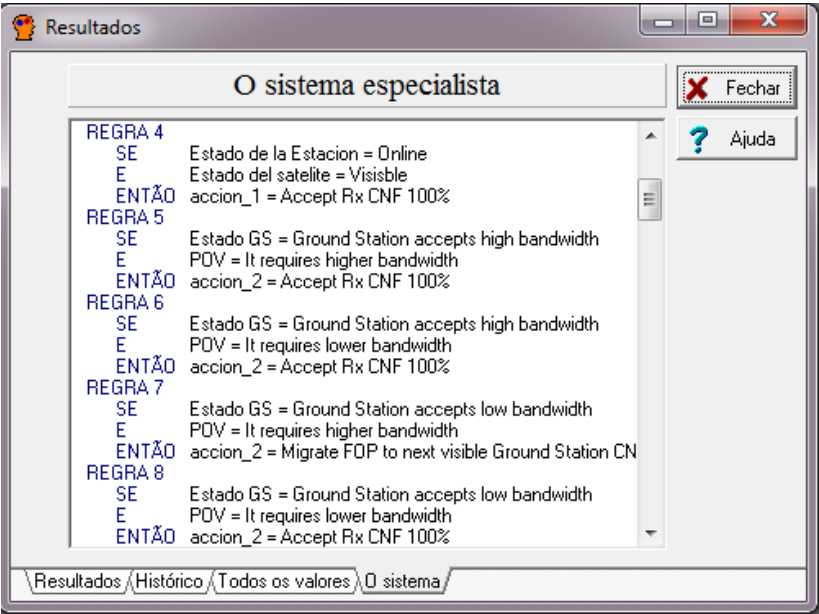


Source: Generated by Expert SINTA

E.13. The expert system

This window shows every rule of the expert system; this information is very useful to understand how the system achieved conclusions, altogether with the record. Figure E.15 shows the window with every rule defined on the expert system.

Figure E.15 - The Expert System



Source: Generated by Expert SINTA

APPENDIX F - THE ADSGS UML MODELING

This appendix to show the model 4+1 views of Kruchten (1995), each one of the diagrams suggested by Kontio (2008) to ADSGS UML modeling where the views are classified in logical view, process view, development view and physical view:

F.1. Logical view

The logical view is concerned with the functionality that the system provides to end-users. UML Diagrams used to represent the logical view include Class diagram, State Diagram, Sequence diagram (KONTIO, 2008).

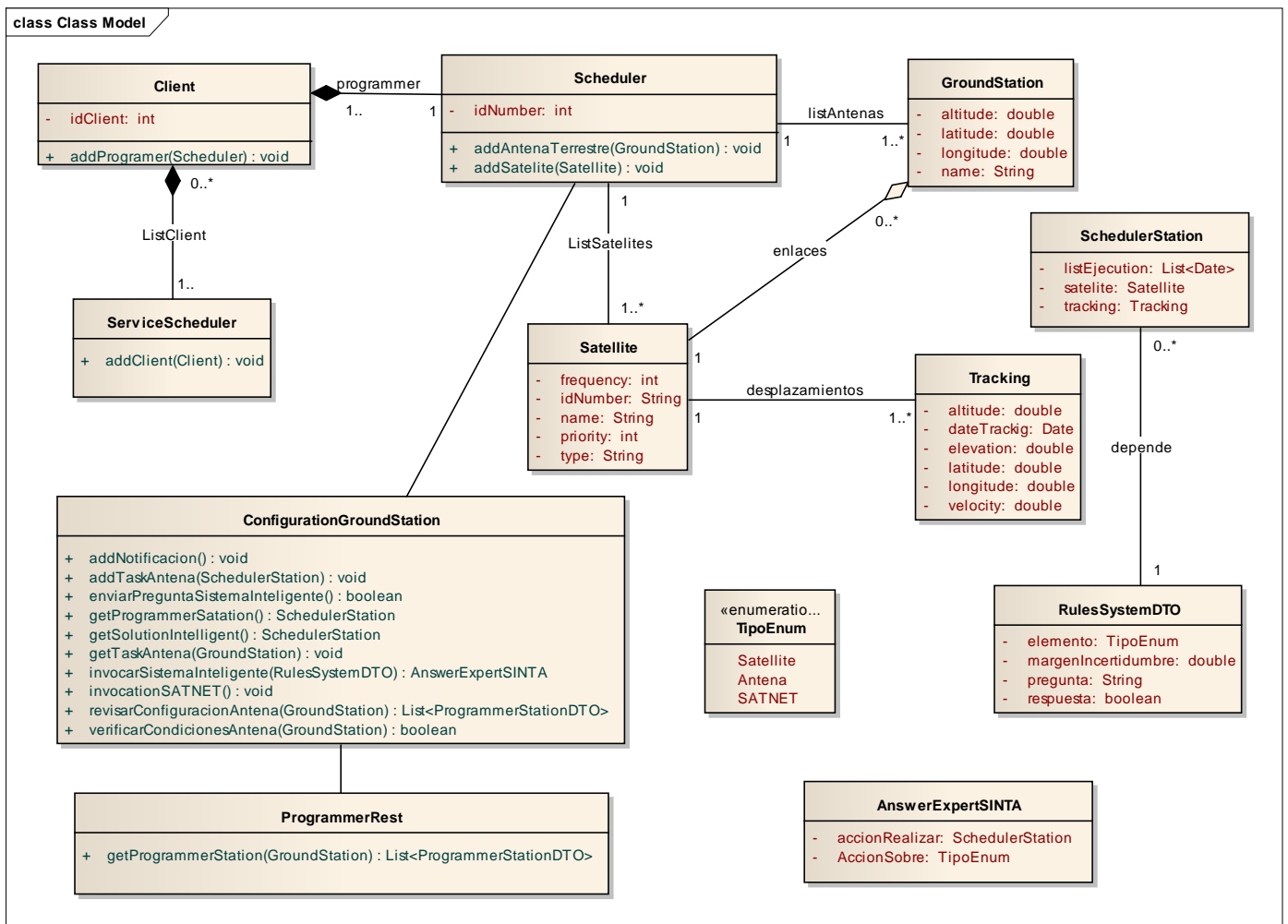
Once the information of the functional requirements of the application is determined, the diagrams are done: entity-relationship model, class diagram of JPA entities and object oriented solution model.

The ADSGS class diagram is shown in Figure F.1, with the different classes created to ADSGS system, among others: Client, ServiceScheduler, Scheduler, GroundStation, Satellite, SchedulerStation, Tracking, ProgrammerRest and ConfigurationGroundStation. It is also explained how is related the data model with the business logic creating a division to two independent layers.

This diagram represents the business where it is evident that there is a list of clients (person interested in managing an ground station), who wishes to program the actions to be performed in the ground station when a certain satellite passes over the signal window of that station, to this is the necessary information of each of the satellites, such as the frequency of download and upload, the priority that has to be attended and the information necessary to do the tracking. This information must be updated with certain periodicity so that the system is executed with a greater degree of precision.

With this data, the system invokes the ADSGS Agent who is responsible for making the most appropriate decisions according to the circumstances; this is done in real time, thus exposing a service Rest that is responsible for informing the ground station of the actions to carry out to make the tracking of the programmed satellite.

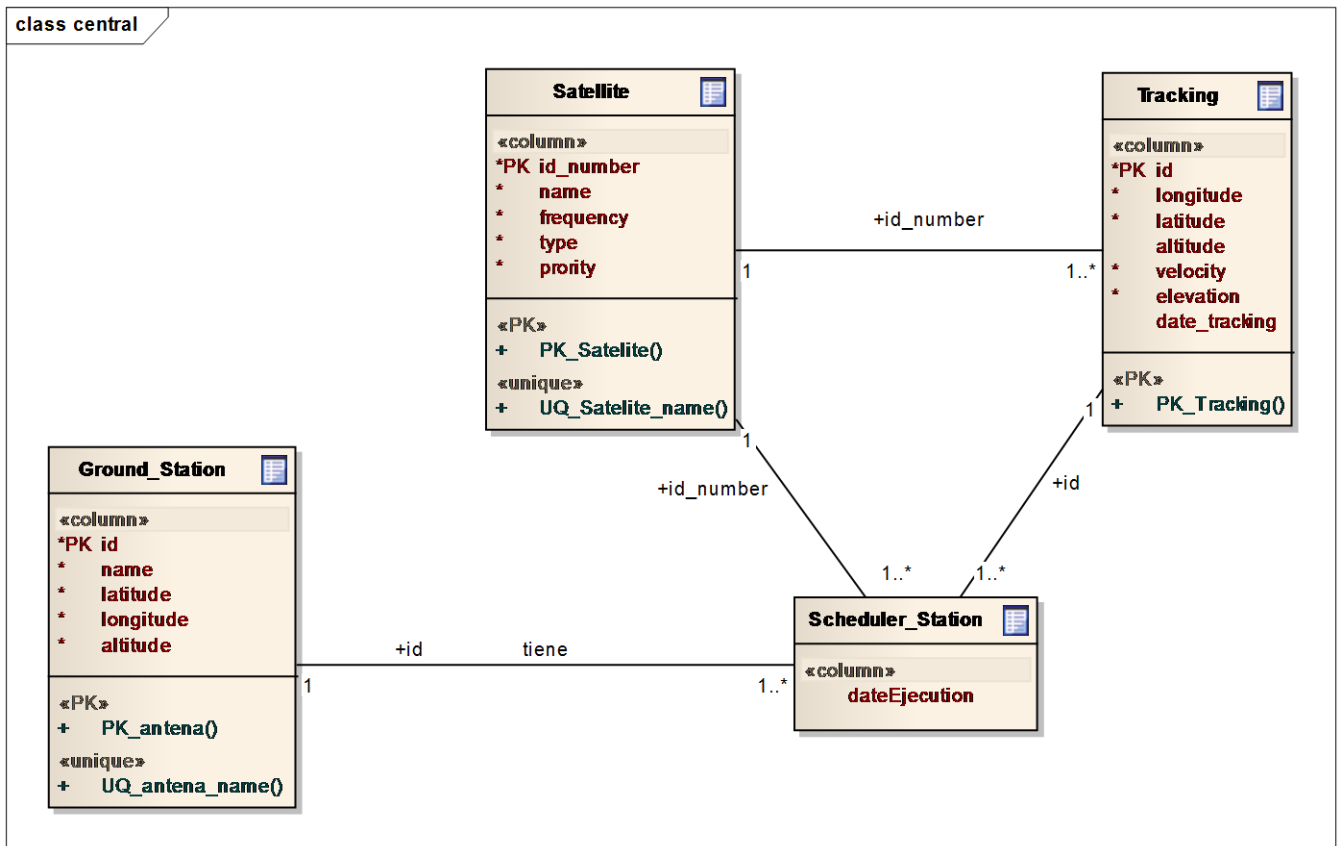
Figure F.1 - ADSGS - Class Diagram



Source: Author

The Entity-Relation model has the following elements: Satellite, Tracking, Ground_Station and Scheduler_Station and the relationship between them. The figure F2 show the different entity relationship.

Figure F.2 - ADSGS - Entity-Relationship Model Diagram



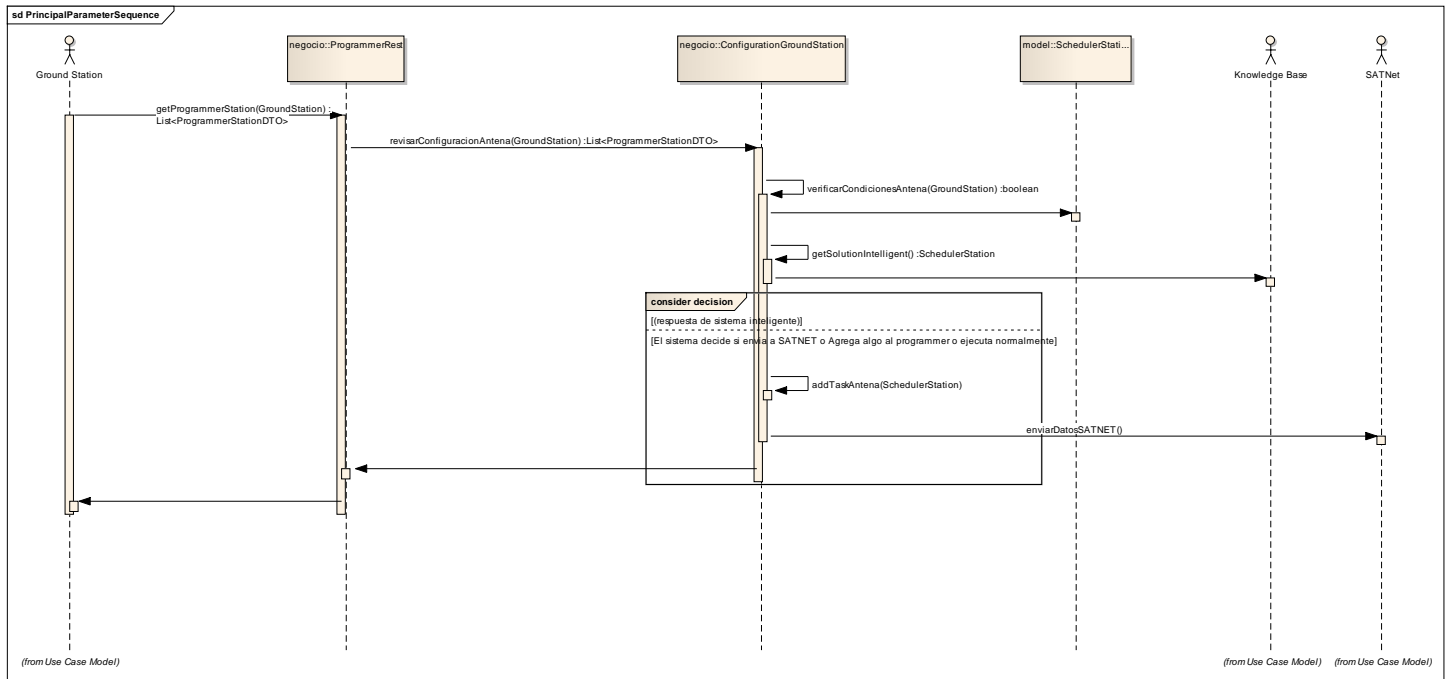
Source: Author

The logical view also frames the sequence diagrams that show the way it is pretended the classes interact with each one of the previously defined requirements.

The entity-relationship diagram has the functionality of saving the information of the data necessary for the System to execute the actions of the business, so there is an entity called **Ground_Station**, which stores the basic data of the stations that a customer wants to manage, each ground station has one or more scheduled tasks, which must be executed on a certain date, these tasks will have associated a particular satellite with its respective navigation route.

The sequence diagram (see Figure F.3) describes how ADSGS uses the knowledge base to give autonomy to the ground station and the steps followed to verify the antennas.

Figure F.3 - ADSGS - Main Parameters Sequence Diagram

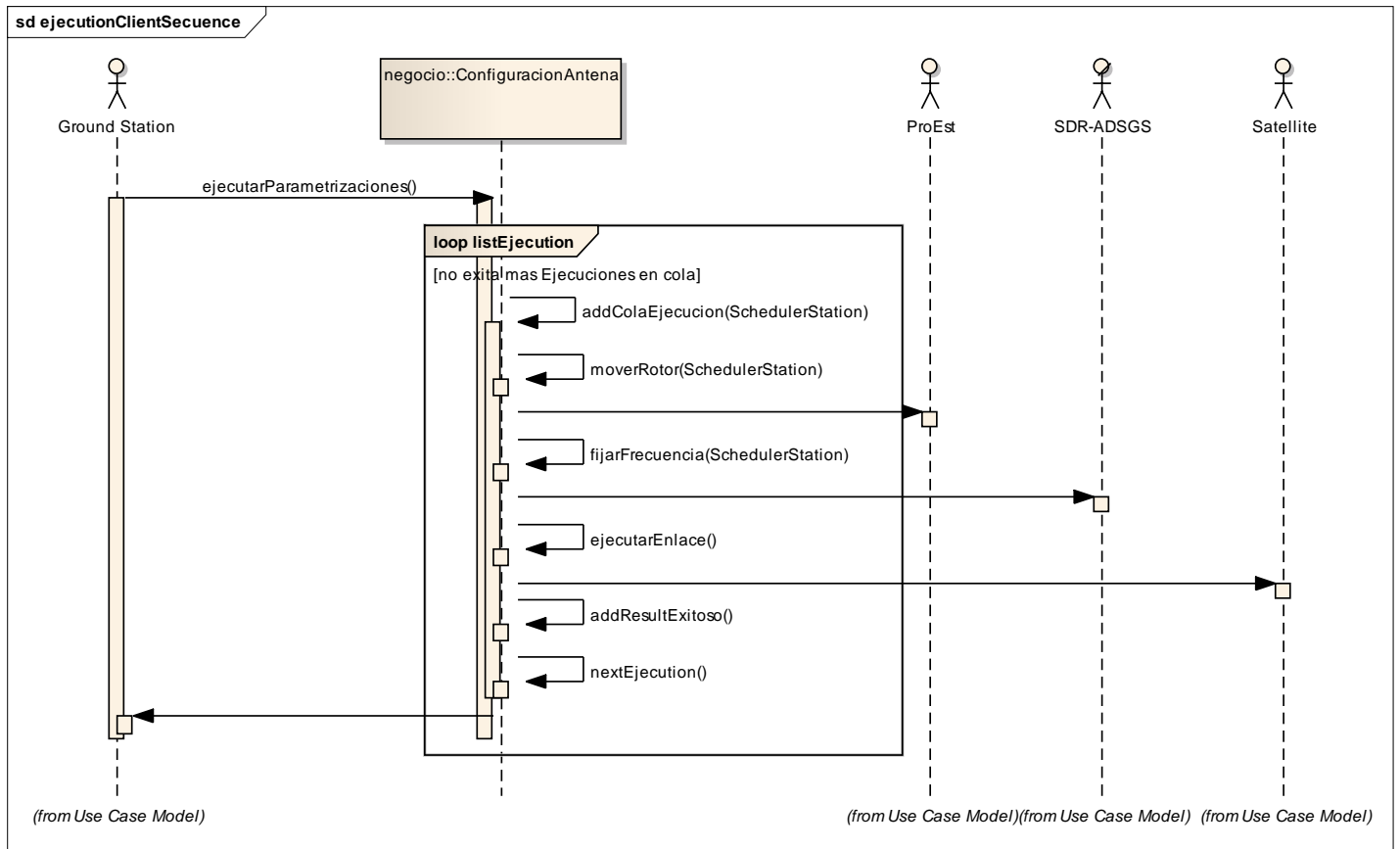


Source: Author

The previous sequence diagram shows how to execute the process of requesting actions from each ground station, starting from this principle there is an antenna, which communicates with the Central System, in that message should go the basic data of the antenna, Such as the current state of its artifacts and other data of interest, in this way the Central System sends this data to the ADSGS Agent so that it makes the decisions about the actions that it must realize or if it must communicate with SATnet and return the answer to the antenna that I request the service.

Similarly, Figure F.4 describes the sequence diagram of the execution of the system and the interaction between the Ground Station, ProEst, SDR-ADSGS and the Satellite.

Figure F.4 - ADSGS - Execution Client Sequence Diagram



Source: Author

This sequence diagram reflects the actions that are carried out after the antenna has communicated with the Central System and the ADSGS Agent, which have already made the relevant decisions, thus the response is not received in the ground station the software standalone begins to execute the action, where it starts moving the antenna through the rotor to locate it and leaves it ready to receive data of the next satellite pass, just finish the process of passing the antenna system must inform the Central System, whether the process was successful or failed and where you can find the information that was sent or received.

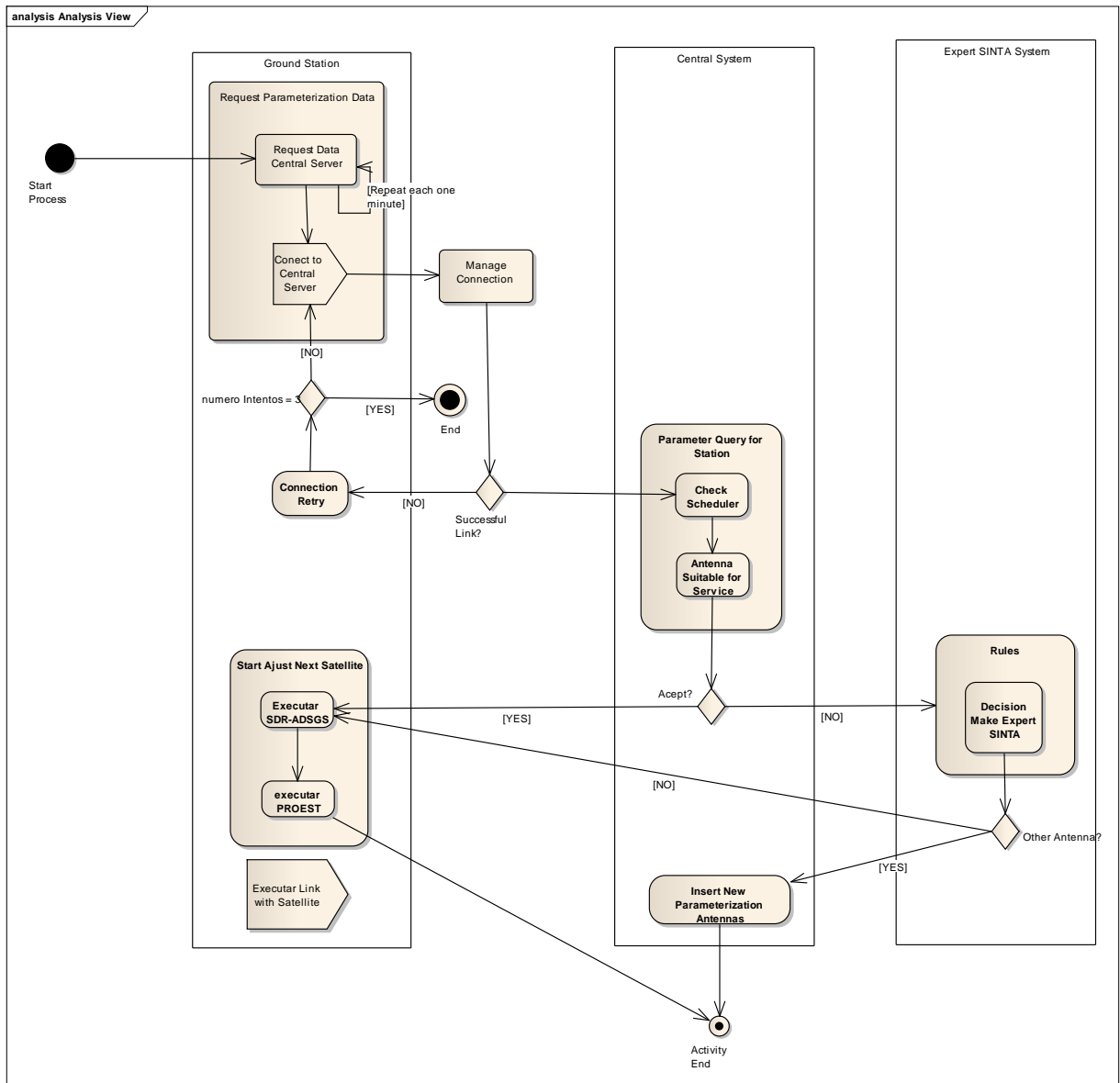
F.2. The Process view

The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behavior of the system. The process view addresses concurrency, distribution, integrators, performance, and scalability, etc. UML Diagrams to represent process view include the Activity diagram (KONTIO, 2008).

Figure F.5 shows the activity diagram, describing the interaction between the antennas system of the ground station, the central system and the rule-based expert system.

The activity diagram shows how the different components intervene in a given process, describes the activity performed to request the actions to be performed by a given station antenna. The process begins with the software of the ground station, which is programmed to communicate with the Central System every time interval to request information, thus the Central System with the parameters given, sends the information to the ADSGS Agent for to make a decision and these instructions return to the ground station, to be executed in real time, this process is repeated again and again, as scheduled by the system administrator.

Figure F.5 - ADSGS - Activity View Diagram



Source: Author

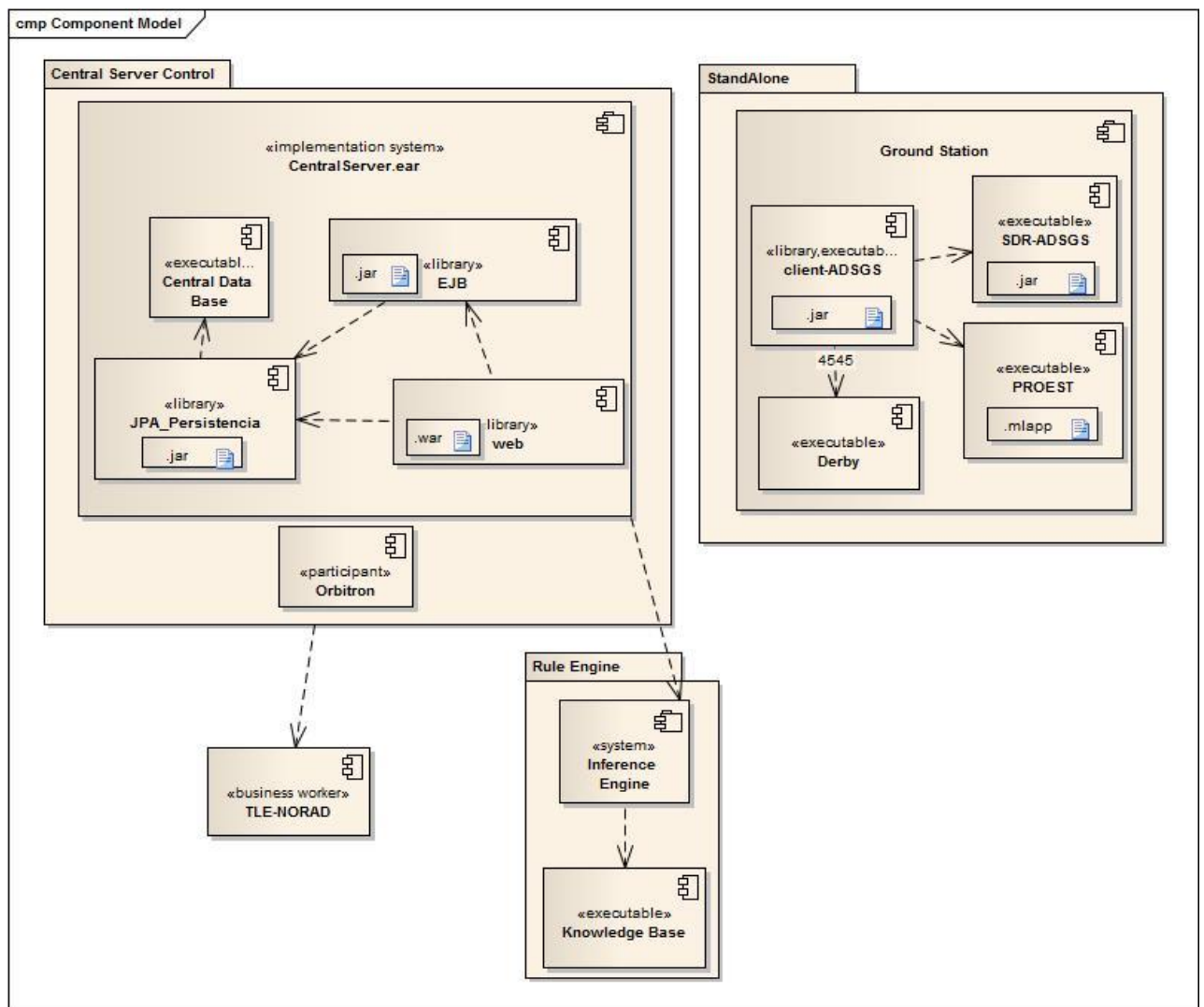
F.3. The Development View

The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML Component diagram to describe system

components. UML Diagrams used to represent the development view include the Package diagram (KONTIO, 2008).

The component diagram (Figure F.6) shows the interaction between the Central Control Server, the rule engine (the inference engine and the knowledge database), Orbitron participation, the required libraries to run ADSGS (EJB, JPA_Persistencia, Web) and the execution of the Central Database. Similarly in Standalone are executed Derby, SDR-ADSGS, ProEst and the library client-ADSGS.

Figure F.6 - ADSGS - Component Diagram



Source: Author

From the component diagram it is described briefly how the different technologies used in the application are connected. Initially the ground station applicative responsible for the integration of different components such as SDR-ADSGS and ProEst, features a connection with the local database which will rely upon Derby. The communication with the web server will be established through web services JAX-RS, which are “light” and do not require complex interface definition because they base their operation in communication via HTTP. The base of the web application must be upon Java EE7 which allows the integration of several of the previously mentioned technologies. The database will be hosted in a server upon MySQL and the connection between the database and the application will be through JPA.

The web application features web services definitions JAX-WS and JAX-RS which allow the connection with third party applications like SATNet. Such app relies upon a Wildfly 10 server which is one of the best servers regarding to compatibility and overall performance.

F.4. Physical view

The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. This view is also known as the deployment view. UML Diagrams used to represent physical view include the Deployment diagram (KONTIO, 2008).

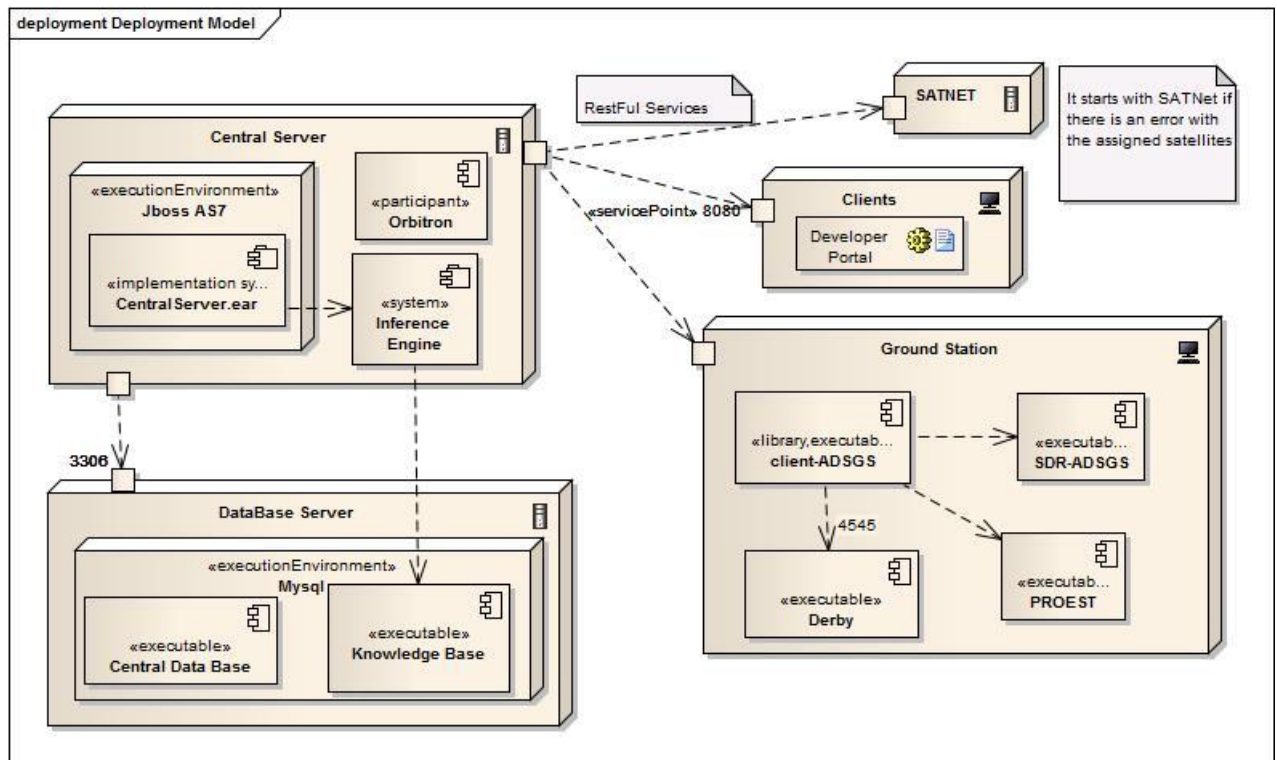
The development of the application is based on the management of the following technologies which together provide a precise technological solution adapted to the application requirements.

- Java SE 8
- MySQL
- Derby
- Java EE7

- Maven 3.0.4
- IDE Eclipse JEE Kepler
- Wildfly 10

The deployment diagram (Figure F.7), shows how the central server interacts with the database server the same as with the ground station, SATNet and the clients.

Figure F.7 - ADSGS - Deployment Diagram



Source: Author

The deployment diagram indicates how the different components of the system will interact with each other, since each has a special feature, for example if it is decided to use a Mysql database, the other components should be communicated with it through Port 3306, to query or modify information from the central database and the knowledge database, while the Central System

exposes its services through port 8080, where it communicates with the ADSGS Agent, the graphical interface of the client and with the software of the ground station that is a standalone application that manages the hardware that is in that station.

ADSGS UML Artifacts Modeling

In this section the main artifacts created to a prospective development of an ADSGS implementation are described: functional and non-functional requirements.

In functional requirements section, the functions identified in the software are presented. Non-functional requirements describe the behavioral conditions and restrictions. In the description of system stakeholders section, it is defined the environment users and their attributes. Finally, the architecture of the project describes the elements of ADSGS architecture.

The proposed process of requirements engineering adopts the concepts of requirements artifact review presented by Leffinwell and Widrig (2003).

F.5. Functional Requirements

They define the inner behavior of software: calculations, technical details, data management and other specific functionalities showing how use cases will be put into practice (MACHADO, 2011). Below are mentioned some functional requirements defined for ADSGS:

FR1. The operator shall access the application (Scheduler), where it must be indicated which communication ports will be used to connect to SATNet database, to connect to SDR, the Tx and Rx ports. These ports can be TCP, UDP or RS232.

FR2. The names of the satellites to be tracked from this station, will be gathered in the database or taken from SATNet, each satellite is an object and must have the following information:

- Name (string)
- Catalog number (numeric)
- Uplink frequency in Hz (numeric)
- Uplink schema (must be selected from a list using the following values AFSK, BPSK. It also allows selecting bitrate with values 1200 bps, 2400, 4800 or 9600 bps).
- Downlink frequency in Hz (numeric)
- Downlink schema (must be selected from a list using the following values AFSK, BPSK. It also allows selecting bitrate with values 1200 bps, 2400, 4800 or 9600 bps).
- The system should automatically take from NORAD website¹⁶, the TLE and display it the same as time and date of update.
- It should be possible to add, remove and edit a satellite.

FR3. It shall be displayed the next pass of the satellite at the top of the list and with the data obtained from TLE, display the start time of satellite tracking, the start time of pass, the pass ending time and the ending time of satellite tracking.

FR4. To start tracking a satellite you must first run the Orbitron software just like the SDR with the given values.

FR5. There shall be synchronization with ProEst to perform a real tracking to the satellite moving the rotors which in turn move the antennas.

FR6. The user shall capture the ground station location in terms of:

- Longitude (decimal format)

¹⁶ Available on: <<https://www.celestrak.com/NORAD/elements/>>. Accessed on: 1 Apr. 2016

- Latitude (decimal format)
- Elevation (given in meters)

FR7. The system shall query the database to perform decision making and synchronizing with SATNet give autonomy to the ground station.

FR8. ADSGS shall allow to send telecommands and receive telemetry and through SDR store this information to encode and decode the sent or received packets.

F.6. Non-Functional Requirements

They specify some criteria used to judge the system operation instead of its specific behaviors (MACHADO, 2011). Some non-functional requirements for ADSGS are mentioned below:

NFR1. The system should have an easy to use interface; the system should be organized in a way that minimizes the errors when operating the ground station.

NFR2. Advanced users shall be able to use all the functions of the system (add new rules to the knowledge base, create new communication ports, etc.)

NFR3. Software response time must be high, considering that ProEst requires time to move the antennas before the satellite pass over a ground station.

NFR4. There shall be compatibility between the knowledge base, SATNet database and ADSGS system.

NFR5. The system shall have high availability at least of 95%.

ATTACHMENT A - ABSTRACTS, POSTERS AND PAPERS

Below are the abstracts, posters and articles that were accepted in different events.

1. *Conference: V Congreso Internacional de Ciencia y Tecnología Aeroespacial - CICTA 2014. Automation of Ground Station State of Art.* November 18-20, 2014. Bogota, Colombia

Automation of Ground Station State of Art

Jorge Enrique Espindola Diaz, Mauricio Gonçalves Vieira Ferreira, Walter Abrahão dos Santos

Abstract: In this work the information found was classified in four areas: automation of ground station, ground station network, use of artificial intelligence and ground station virtual. This is because these issues are part of a ground system and ranking facilitates their study. The following describes the information found and an analysis of the advantages, opportunities and strong points, as well as the disadvantages, limitations and weak points. This information was taken from articles found both on the internet and in specialized magazines written in the last few years. Each selected article shows the authors, the contribution in this area of knowledge, the research institute or university where they worked, the year and last the country of origin.

Key Words: Artificial Intelligence, Automation of Ground Station, Ground Station Network, Ground station Virtual.

2. *Conference: 1st Latin American IAA CubeSat Workshop. A survey on State of Art in Ground Station for Miniaturized Satellites.* December 8-11, 2014. Brasilia, Brazil.

A Survey on State of Art in Ground Station for Miniaturized Satellites

Jorge Enrique Espindola Diaz, Mauricio Gonçalves Vieira Ferreira, Walter Abrahao Dos Santos

Every satellite, small, medium or large, needs to be monitored from Control Centers and Ground System also referred as Ground Station, designed for that purpose. These stations must be completely automated or else largely, in order to make their use more widespread, especially economic, and to guarantee easy access to every type of user. To deepen the knowledge on earth stations for pico-satellite control, it is necessary to know the state of the art of automation, the

evolution and the latest developments in this technology. Also, it is necessary to know used communication protocols, CubeSat projects that were developed or are currently being developed, organizations that regulate the use of pico-satellites, the design of ground stations, the most used software for tracking satellites and to look into known techniques for satellite tracking. This work is a compilation of the papers published concerning the evolution of ground system in the last 10 years. In addition it gets a summary and critical analysis about state of art to ground station. The objective of this work is to make a state of art in ground station for miniaturized satellites, verify the existence of the use of artificial intelligence techniques in the automation of ground stations and discover how we are working on ground station around the world.

3. Conference: 6^o. Workshop em Engenharia e Tecnologia Espaciais 2015, *Uma Abordagem Dinâmica e Autônoma para Operações Espaciais de Pequenos Satélites*. August 18-20, 2015. São Jose dos Campos, Brazil.

Uma abordagem Dinâmica e Autônoma para Operações Espaciais de Pequenos Satélites

ESPINDOLA, J. E.D., FERREIRA, M.G.V., Dos SANTOS, W.A.

¹Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Aluno de Doutorado do curso de Engenharia e Gerenciamento de Sistemas Espaciais.

²Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Tecnologistas

Jorge.espindola@inpe.br

Resumo. *Atualmente há uma enormidade de missões educacionais na área de Engenharia Aeroespacial sendo realizadas que devem também lidar com operações da missão em bordo.[Datta, 2012] A Rede Educacional Global para Operações de Satélite (GENSO, em Inglês, Global Educational Network for Satellite Operations) está formada por uma rede mundial de estações terrenas e satélites que podem interagir entre si através de um software padrão. A GENSO aumentou os pontos de contato com missões espaciais educacionais e mudou a maneira como essas missões eram originalmente empreendidas. Este trabalho tem por objetivo desenvolver uma camada middleware compatível com o padrão da GENSO, tentativamente, que permita a automação de alguns procedimentos em estações de terra.*

Palavras-chave: Picossatélites; Nanossatélites; Estação terrena; Inteligência Artificial; GENSO.

4. Poster: Spring 2015 CubeSat Developer's Workshop. *Autonomous and Dynamic System for Ground Station*. April 22-24, 2015. San Luis Obispo, CA, USA.

5. Conference: II Latin American IAA CubeSat Workshop. *Integration of the INPE Ground Station into the SATNet Network for Supporting Small Satellites Programs in Brazil*. February 28 to March 2, 2016. Ingleses Beach, Florianopolis, Brazil

IAA-BR-16-0S-0P

Integration of the INPE Ground Station into the SatNet Network

Jorge Enrique Espindola Diaz¹, Mauricio Gonçalves Vieira Ferreira², Walter Abrahão Dos Santos³.

The SatNet network aims at incorporating the capabilities of all the already deployed university Ground Stations into a single, coherent and usable resource. The approach used for this network is based on heterogeneity, allowing the integration of very different ground stations into a single ground system. Within this scope, each ground station is considered to contribute with its own communication capabilities, incorporating a set of new channels (modulations, bitrates, bandwidths, polarizations... etc.) from a different location - thus, extending the coverage of the ground system. The operational availability of each ground station will be defined by the owners, depending on issues such as their degree of autonomy, their maintenance schedule and reliability... etc. Satellite operators are expected to access this network with their own clients, registering in the network their satellites and their associated communications requirements. These will be used by the network to compute the compatibility with the ground stations registered. Combining the compatibility and the availability, spacecraft operators will only be allowed to schedule the utilization of remote ground stations during the passes of compatible satellites over ground stations that are available. In order to accommodate all this complex diversity, the network is composed of a central server that orchestrates the functioning of the system. Remote ground stations are only required to install a simple client that allows sharing its resources with remote spacecraft operators. Therefore, the key for the success of this approach relies on the capacity of integrating this ground station client into very diverse ground stations. The second release of the software already includes a generic client that eases the integration process of these ground stations through a process of adaptation to the specific hardware. The Ground Station at the INPE in Brazil has started working, together with the main developers of the SatNet network, in the integration process of the software client in their ground station. This paper presents the description of the integration process together with the results obtained.

6. Abstract: 13th Annual CubeSat Developer's Workshop. *Deployment of the SATNet Network at the INPE (Brazil) and University of Vigo (Spain)*

Facilities with a Central Server at the California Polytechnic State University (USA). April 20-22, 2016. San Luis Obispo, CA, USA.

Deployment of the SatNet Network at the INPE (Brazil) and University of Vigo (Spain) Facilities with a Central Server at the California Polytechnic State University (USA)

Ricardo Tubío-Pardavia, Jorge Enrique Espindola Diaz, Mauricio Gonçalves Vieira Ferreira, Walter Abrahao Dos Santos, Jordi Puig-Suari

The researchers of the Ground Station at the National Institute for Space Research[1] (INPE) in Brazil have started working, together with the main developers of the SATNet network, in the integration of their ground segment into that very same network. This paper presents the description of that integration process together with the results obtained and with a description of the current and upcoming small satellite projects in Brazil and in Latin America. The SATNet network aims at incorporating the capabilities of all the already deployed university Ground Stations into a single, coherent and usable resource. The approach used for this network is based on heterogeneity, allowing the integration of very different ground stations into a single ground system.

7. Abstract: 14th International Conference on Space Operations - SpaceOps 2016. *An Autonomous and Dynamic Approach to Ground Stations Control for Small Satellites*. May 16-20, 2016. Daejeon, Republic of Korea.

An Autonomous and Dynamic Approach to Ground Stations Control for Small Satellites

Jorge Enrique Espindola Diaz, Mauricio Gonçalves Vieira Ferreira, Walter Abrahão Dos Santos.

National Institute for Space research, São Jose dos Campos, SP, 12227/010, Brazil

Abstract. This research work aims to develop a layer that automates some procedures in ground stations to track small satellites. This layer, called ADSGS (Autonomous and Dynamic System for Ground Stations), is under development by the National Institute for Space Research (INPE) Brazil. Additionally, ADSGS allows the dynamic management of the different small satellites missions by using an infrastructure based

on the SATNet network originally developed by researchers at the California Polytechnic State University (CALPOLY), San Luis Obispo, USA, and the University of Vigo (UVIGO) in Spain. SATNet is composed by a worldwide network of ground stations that can interact via standard open software. The SATNet increased the return of educational space missions and changed the way original missions are managed, dramatically increasing the level of their accessibility to space. The ADSGS layer intends to employ new techniques for automating and improving the performance of ground stations, so that they can reduce their operating costs. Doing so, it contributes to the expansion of the space community by facilitating the access to science and aerospace technologies. This research work proposes a new architecture that provides control of one or more satellites without operator intervention. This is performed by a rule-based knowledge model which allows one to make decisions and run actions. A database that stores the satellite information, such as their flight plans and parameters, will be developed preferably using open source. The satellite operational flight plans can be monitored from automated ground station stored in a database created for this purpose. The ADSGS layer will manage the database which contains information from satellites in the SATNet network maintaining system integrity and providing security to the users. As a main objective of this proposal, it is the creation of an autonomy and dynamic system that will control the operations of a ground station in order to receive telemetry and send telecommand to small satellites. The ADSGS layer will be add at the Ground Station Client (G-Client) module available at SATNet for the operation of the ground station. This idea will become the current station into an intelligent ground station, be able to making its own decisions and thus reducing the operating costs of it.

8. Conference: *7o. Workshop em Engenharia e Tecnologia Espaciais 2016*, ADSGS - Autonomous Dynamic System Ground Station. September 23-24, 2016. São Jose dos Campos, Brazil.

ADSGS (Autonomous Dynamic System Ground Station)

ESPINDOLA, J. E.D.¹, FERREIRA, M.G.V.², Dos SANTOS, W.A.²

¹Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Aluno de Doutorado do curso de Engenharia e Gerenciamento de Sistemas Espaciais.

²Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Tecnologistas

jorge.espindola@inpe.br

Resumo. Neste trabalho é proposto a criação de ADSGS (Autonomous Dynamic System Ground Station), que é um “middleware” uma abordagem autônoma e dinâmica para a operação de pequenos satélites. ADSGS

desenvolve uma proposta definitiva para a estação terrena com componentes de hardware COTS (Commercial Off-the-shelf), esta estação terrena será usada para monitoramento e controle de pequenos satélites, (pico, nano e femtosatelites). O “middleware” proposto está entre o hardware e SATNet que é uma rede de estações terrenas desenvolvida pela equipe de pesquisa liderada pelo professor Yordi Puig, co-criador do padrão CubeSat, esta equipe de pesquisadores fazem parte da Universidade CalPoly California State Polytechnic - EUA, a Universidade de Vigo - Espanha e INPE (Instituto Nacional de Pesquisas Espaciais) - Brasil. Esta abordagem permitirá que as estações terrenas tenham autonomia e dinamismo no funcionamento de pequenos satélites e redução dos custos operacionais dos mesmos.

Palavras-chave: Pico satélites; Nano satélites; Estação terrena; Inteligência artificial.

9. Conference: *VI Congreso Internacional de Ciencia y Tecnología Aeroespacial - CICTA 2016*. ADSGS and SATNet. October 27-28, 2016. Cali, Colombia.

ADSGS and SATNet

Jorge Enrique Espindola Diaz, Mauricio Gonçalves Vieira Ferreira, Walter Abrahão Dos Santos

Abstract: In this work creating intends of ADSGS (Autonomous Dynamic System Ground Station), which it is a middleware as an autonomous, dynamic approach to the operation of small satellites. ADSGS develops a definite proposal for ground station with hardware components COTS (Commercial off-the-shelf) this earth station will be used for monitoring and control of small satellites, (pico, nano and femtosatelites). The proposed middleware will be between the hardware and SATNet which is a ground station network developed a working group led by Professor Yordi Puig, co-creator of Cubesat standard, these groups of researchers are part of the California Polytechnic State University CalPoly - USA, the University of Vigo - Spain and INPE (National Space Research Institute) - Brazil. This approach will allow ground stations have autonomy and dynamism in the operation of small satellites and operating costs thereof are reduced, one way to do this is by developing a middleware that allows the ground station is autonomous and dynamic in operating activities satellite tracking and monitoring.

Key Words: Pico-satellites; Ground Station; Artificial Intelligence.

10. Paper: Ground station for miniaturized satellites: a state of the art survey.
Magazine: *Redes de Ingenieria*. ISSN: 2248-762X

Ground Station for Miniaturized Satellites: a State of the Art Survey

Estado de arte sobre estaciones terrenas para pequeños satélites

Tipo de artículo: Artículo revisión

Fecha de recepción: 26/11/2016

Fecha de aprobación:

Abstract

In order to deepen the knowledge on ground stations for pico-satellite and nano-satellite control, it is necessary to know the state of the art of automation, the evolution and the latest developments in this technology. Also, it is necessary to know: used communication protocols; CubeSat projects that were developed or are currently being developed; organizations that regulate the use of pico-satellites; the design of ground stations; the most used software for tracking satellites, and to look into known techniques for satellite tracking. This work is a compilation of the papers published concerning the evolution of ground system in the last 10 years. In addition it provides a summary and critical analysis on state of the art to ground station. The objective of this work is to make a state of the art in ground station for miniaturized satellites, evaluate the existence of the use of artificial intelligence techniques in the automation of ground stations and identify how institutions are working on ground station around the world.

Keywords

Cubesat, Artificial Intelligence, Ground Station, Ground Station Network, Small Satellites.