POLITECNICO DI TORINO

Facoltà di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria informatica

Tesi di Laurea

Analysis And processing Of Information Transmitted By Vessels

Relatori:

Prof. Maurizio Morisio

Enrico Baccaglini

Daniele Brevi

Candidato:

Getahun TesfayeAlemu

ANNO ACCADEMICO 2017-2018

Section with acronyms

AIS Automatic Identification System
CSTDMA Carrier Sense Time Division Multiple Access
DSC Digital Selective calling
HELCOM Baltic Marine Environment Protection Commission
ITU International Telecommunications Union
IMO International Maritime Organization
IEC International Electrotechnical Commission
IALA The International Association of Marine Aids to Navigation and Lighthouse Authorities
LOCODE United Nations Code for Trade and Transport Locations
MMSI Maritime Mobile Service Identities
MarNIS Maritime Navigation and Information Services
NMEA National Marine Electronics Association
ORM Object-relational mapping
PAS Digital selective calling
PSS Publicly Available Specifications
SOLAS Safety of Life at Sea
SOTDMA Carrier Sense Time Division Multiple Access
TDMA Time-division multiple access
TCP Transmission Control Protocol
UTC Coordinated Universal Time
VHF Very high frequency
VDL VHF Digital Link
VTS Vessel Traffic Service

Acknowledgments

First of all, I would like to thanks to T.V.(CP) Antonio Vollero Comando Generale del Corpo delle Capitanerie di Porto that launches the idea and provides the AIS flow, which allowed me to develop an interesting thesis proposal.

Special thanks are addressed to the members of the Multi Layer Wireless Solutions research area of Istituto Superiore Mario Boella, with whose collaboration this thesis has been realized. In particular, I would like to thank Enrico Baccaglini and Daniele Brevi for their continuous support whenever I had a question about my research. Their patient guidance, excellent supervision and constant support was significant throughout my work. Particularly they allowed this thesis to be my own work but steered me in the most fruitful direction whenever they thought I need it. Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.
kali your are my challenge, my inspiration and my hope!! love you at most.

Table of contents

1 Introduction **4**

2 State of the art **8**

2.1 Automatic Identification System 8

2.1.1 Class A And Class B Automatic Identification System 8

2.1.2 AIS Recommendation and Standards 9

2.1.3 Technical Specification of AIS 10

2.1.4 ITU-R M.1371 and Message Type 15

2.2 Open challenges and issues 18

2.3 Possible Approach 20

2.3.1 Algorithm for distance metrics 20

2.3.2 Regular Expressions 20

3 Proposed Implementation **21**

3.1 Architecture for the System 21

3.2 Decoding AIS message 23

3.2.1 Edit Distance 27

3.3 Java - Regular Expressions 28

3.4 Hibernate ORM 35

3.4.1 Benefits of Using Hibernate 35

3.4.2 Hibernate Sessions 36

3.5 The Work Flow Of the Implementation 39

3.6 Data Base Architecture 48

3.6.1 Migration of Data Base 48

3.6.2 PostgreSQL Database 48

4 Experimental results **51**
4.1 Preparation of the Data 51
4.2 Evaluation of the Algorithm 52
4.3 Analysis of the Results 52
4.3.1 Port Name as an Input 53
4.3.2 Port Code as an Input 55
4.3.3 Country Code with Port Name as an Input 56
4.3.4 Country code with Port Code as an Input 58

5 Conclusion, Limitations and Future Work **60**

Bibliography **63**

Abstract

The work carried out in this thesis is in a collaboration with Institute Superiore Mario Boella, and Capitaneria di Porto, to identify the most likely value for source and destination ports by processing information which is sent by vessels using the AIS (Automatic Identification System) system.
The project is a part of the AIS system by which vessels transmit and receive a set of static and dynamic information using a VHF transponder on board. In Shore side, a network of AIS Base Station can be found, operated by Guardia Costiera in accordance with Legislative Decree 19 August 2005, No.196. The management of this a network is charge of a central system which interacts with external systems by means of web services.
The AIS is a collaborative navigational aid system that allows for the automatic identification of naval units equipped with special VHF transponders, made mandatory on particular naval units and certain sizes [from Chapter V of the London Convention November 1, 1974 Safety of Life at Sea (SOLAS)] and it allows to capture information on ship behavior, supporting, the monitoring activities aimed at ensuring the safety of navigation within Maritime. The system require data collection based on continuous data exchange between ships and base stations via two VHF channels.

The AIS system board sends a series of messages by means of information containing various fields and internationally standardized by regulations ITU-R M.1371. This information is encoded in AIS messages which include, among the other fields, source and destination ports. ITU does not provide a world-wide accepted standard for port name encoding and therefore it is left to the user's choice the way to insert this information. As an example, different ships all going to Genova (Italy) may have the destination port field coded as GENOA, ITGEO, etc. Such heterogeneity of encoding way does not allow to collect reliable statistics in an automated manner.

The main objective of this work is to propose a system which automatically parse the AIS data stream, looks for the source or destination fields and automatically proposes the best matching port name.
The problem is that, those fields are compiled in different ways to the same destination and source ports such that ships bound for the port of Genoa and each could forward destination in a different ways: GENOA, ITGEO, GENOVAIT, ITALYGE etc. Such variety of information does not allows us to carry out checks and statistics in automated way. Therefore, some activity are necessary for data normalization of those fields. The activities to be carried out will consist in the analysis of the problem and to identify the best algorithm that allows us to obtain the basis of statistical evaluation on the ports, choosing and evaluating different types of algorithms mainly Levenshtein distance and related algorithms.

The thesis is organized as follows:

- Chapter 1 :
  Is an introduction on the Automatic Identification System (AIS), and the navigation system with its main characteristics.

- Chapter 2:
  Illustrates the state of art in the navigation system for vessels. Current architectures and system components are described together with open challenges and current issues with the most recent and innovative solutions with a possible approach to solve the problems.

- Chapter 3 :
  We discuss the proposed architecture. A description of a possible implementation for extracting data and processing the fields is given. The algorithm to evaluate the distance between words (Edit distance) is described together with the design approach and class diagrams.

- Chapter 4:
  Shows some experimental results in the developed graphical user interface

together with some examples. Finally, in Chapter 5 we summarize the conclusions and propose future work.

Chapter 1

Introduction

In recent years the marine transportation system has come under increasing attention. Safety and efficiency are among the most important open issues of interest, Research is carried out for ship analysis and tracking to avoid e.g. illegal operations. Moreover, the mariner need a better and real time information about waterway conditions that have been increased due the raise of the number of ships and the increase of traffic density.

In addition, the previous generations of collision avoidance electronics were not able to identify any given radar target when multiple contacts are being tracked, especially at night or in reduced visibility, when it is impossible to verify a ship's identity visually. This issue has been cited as a contributing factor to many collisions at sea in the past years.

To resolve these problems, the Automatic Identification System (AIS) has been developed and much work has been done to define AIS technical and communication requirements. These efforts have resulted in worldwide mandatory carriage requirements for AIS-equipped vessels. Indeed, the International Maritime Organization (IMO) recognized that it was important to use systems of navigation for maritime safety and the prevention of the marine pollution. Thus, the AIS was set up as a regulation project which would require that any ship that comply with the International Convention for the Safety of Life at Sea (SOLAS) must be equipped with an embedded system enabling them to determine their position automatically and to transmit this information to all the interested people, at sea or shore, regularly and

in an automatic way. Therefore, AIS system appears to be very practical because of the great importance of the AIS market and the immense to approve the whole maritime environment but also to improving the organization of the ports and the operations. AIS information includes vessel position, source and destination stations (ports), course, speed and other dynamic information as well as the vessels name, type and other static information. Due to the widely use of AIS, crew and maritime management authorities can quickly access to real-time information of waterway traffic situation and statistics. A growing number of scholars and external tracking application use the AIS information to implement their research and application through the use of the vessel position information in the AIS message which can be used to identify the waterway traffic trend, to restored vessel track, analyze accidents and study traffic flow distribution . The AIS system board sends a series of

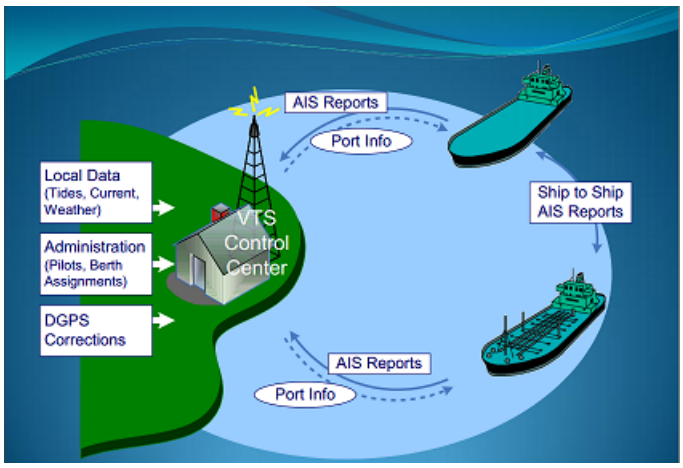


Figure 1.1. Usage of Port info

messages by means of information containing various fields and internationally standardized by regulations ITU-R M.1371. The information sent by ships and received from the network installed along the Italian coast contains useful field for vessel identification e.g. source and destination ports. The destination field provides the port name in which the naval unit is leading to. The recommendation ITU does not provide pre-formatted fields and therefore these fields are filled with free text. As a result destination field will usually compiled in different ways due to the fact that these fields can be compiled freely by the user can mislead the receivers of this

	Abe	Def
De	2	1
Abe	0	3

Table 1.1. distance between words

information. As an example, the authority may not be aware of the real destination of the vessels and security of waterway may be at risk. An Automatic port name recognition is thus very desirable to minimize human errors when typing this data.

In this work we propose an architecture to automatically identify and processing AIS streaming. The objective is to develop a system able to automatically map the incoming port names to a set of well-known values. The system is based on the Levenshtein's edit distance [Levenshtein 1966] because of it's flexible, fast and easy to implement by Dynamic Programming, and customized it to deal with some kinds of errors and attributes of database application domain. The algorithm, called WordMatchEditDistance, has proved to be efficient in terms of accuracy and performance.
One of the main problems in modeling a database that contains information of the port in the world is the difficulty in ensuring a unique identification for them, as there is no a unique world-wide identification for ports. Hence one has to rely on semi structured information like port name, country, port code, country code and other properties of the AIS information, in order to try to identify them. This information can be obtained by UN/LOCODE code list by country.

However, the information obtained from incoming AIS stream is not always trustworthy, because the names and codes of ports may have typing errors, missing words, transposition of words, acronyms and abbreviations, bringing forward the problem of inaccuracy in the application of exact record matching. In order to test the efficiency of the solution, a database containing a very large set of country and port name has been migrated and used in our system.

One of the main contributions of this work is then to significantly improve the safety and anti-collision capability of vessels. In fact, by adding the proposed functionalities to the AIS system, it is possible to aid vessel's manager to track other vessels and reduce workload for captains.

Chapter 2

State of the art

2.1 Automatic Identification System

AIS is an automatic system for the exchange of navigational information between suitably equipped vessels and shore stations using distinct messages and operating on two designated marine VHF channels. AIS allows for an improved awareness for navigators by overcoming the inherent limitations of sight, voice over VHF and radar for collision avoidance.

2.1.1 Class A And Class B Automatic Identification System

AIS information is divided into two classes classes A and B depending on the AIS transponder transmitting the AIS information. These classes are of great importance to the capabilities of the AIS. There is a great difference between the two classes, both in terms of extent, complexity and price.

AIS information from a class A transponder will always be prioritized and, thus, be shown to other ships in the area, whereas AIS information from a class B transponder will not be shown until or if there is room on the AIS channel. AIS of class A In order to avoid that the ships AIS systems all speak at the same time, large ships use an AIS system of class A, which is called SOTDMA (Self-Organized TDMA).

An algorithm ensures that the AIS transmitter of a ship first notices how other ships transmit their messages and, subsequently, adjusts its own transmission pattern to that of the others. In case there are more ships fitted with AIS of class A in

an area than the capacity of the bandwidth, the system will automatically limit the area of coverage so that the remotest ships in the area are discarded. AIS of class B Small vessels fitted with AIS, such as recreational craft, can use a less expensive AIS station of class B, which transmits less frequently. This system is called CSTDMA (Carrier Sense TDMA). A class B station will listen for a couple of milliseconds to hear whether a large ship is transmitting before it transmits its own message.

2.1.2 AIS Recommendation and Standards

There are four primary international standards for AIS equipment. They are developed jointly by International Maritime Organization (IMO), International Telecommunications Union (ITU), International Electrotechnical Commission (IEC) and other organizations. The shipboard AIS equipment must meet the provisions of the following documents:

- IMO Resolution MSC.74 (69), Annex 3, [1]: establishes carriage requirements for AIS and performance requirements for the shipboard equipment.

- ITU-R Recommendation M.1371-1[2]: This document presents technical characteristics for universal ship borne AIS using Time Division Multiple Access in the Maritime.

- Mobile Band: This standard specifies in detail the technical characteristics of the system, the operational requirements of the performance standard and it provides the technical criteria for the AIS.

- IEC(International Electrotechnical Commission) Standard 61993-2[3]: presents the universal shipborne AIS, specifies the minimum operational and performance requirements, methods of testing and required test results conforming to the performance standards contained in IMO Resolution MSC.74 (69), Annex 3, and it also incorporates the technical characteristics contained in ITU-R M.1371-1. IEC 62287-1 Ed.1[4] [5]: It defines a new access technique: CSTDMA used by class b ship, and specifies the minimum operational and performance requirements, methods of testing and required test results conforming to the performance standards.

AIS is a broadcasting system, functioning in VHF maritime band, able to transmit ship information, such as the identity, the position, the course, the length, the width, the type and the draught of the ship and the information of dangerous cargo, to other ships as well as to base stations. Three modes of operation are foreseen namely assigned, autonomous, continuous and polled. The default mode is autonomous and may be switched to and or from other modes as required by a competent authority.

Each station transmits and receives over two radio channels to avoid interference problems. The system provides for automatic collision resolution between itself and other stations, and communications integrity is maintained even in overload situations. If a station operates on the SOTDMA mode, it shall determine its own transmission schedule, based upon data link traffic history and knowledge of future actions by other stations. A position report from one AIS station fits into one of the 2250 time slots established every 60 seconds. AIS stations continuously synchronize themselves to each other, to avoid overlap of slot transmissions. Slot selection is randomized within a defined interval, and tagged with a random timeout between 4 and 8 frames. When a station changes its slot assignment, it pre-announces both the new location and the timeout for that location. On the other hand, an AIS station which functions on the CSTDMA mode determines its own nominal transmit slots after monitoring the two channels for one minute and continues to transmit upon these slots according to its own schedule until its report rate changes or it stops. Figure 2.1 shows a scheme for the AIS VHF data link. [1]

2.1.3 Technical Specification of AIS

AIS network The AIS network is subdivided into mobile and fixed stations. This subdivision determines the intended purposes of the AIS stations and thereby the capabilities associated with these stations. On one hand, mobile stations are intended to be used by mobile participants of the AIS, such as vessels, SAR (Search and Rescue) aircrafts and in particular floating Aids-to-Navigation. On the other hand, fixed AIS stations which exhibit a much superior functionality in terms of

[1] Essaadali, Riadh and Jebali, Chokri and Grati, Khaled and Kouki, Ammar. (2015). AIS data exchange protocol study and embedded software development for maritime navigation. 10.1109/CCECE.2015.7129519.

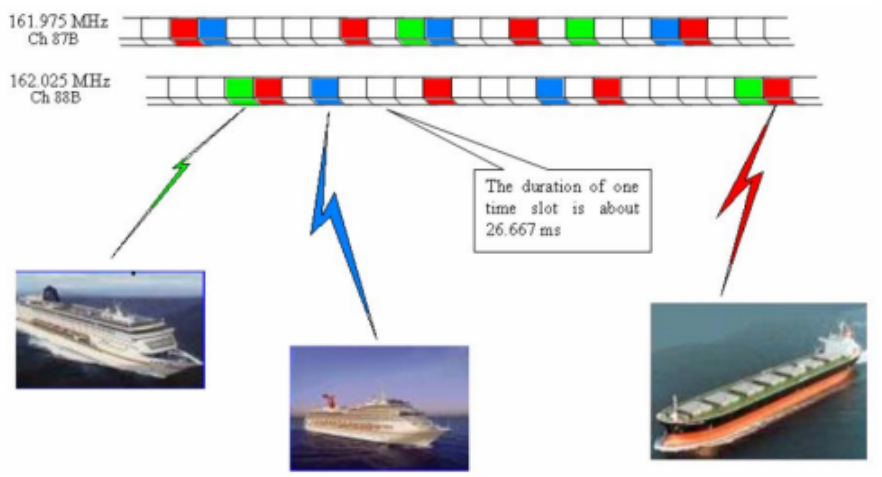


Figure 2.1. AIS VHF data link

controlling the AIS VDL(VHF Digital Link) than mobile AIS stations are intended to be used by the shorebased competent authority when setting up its AIS Service.

AIS ship mobile equipment is the principal component parts of a ship borne mobile AIS station. The GPS receiver supplies the coordinated universal time (UTC) to the AIS station to correct its own clock in order to synchronize all transmissions such that there are no collisions or overlaps which would degrade the information being transmitted. The VHF transceiver transmits and receives radio signals that form the VHF Data Link and interconnects the AIS stations to each other. The DSC VHF receiver is fixed to channel 70 to receive channel management commands for regional area designation. The processor manages the time slot selection process, the operation of the transmitters and receivers, the processing of the various input signals and the subsequent distribution of all the output and input signals to the various interface plugs and sockets, and the processing of messages into suitable transmission packets. A built-in integrity test continuously controls any failure or malfunction of the unit. The keyboard inputs all required voyage-related information. The display shows all received vessels, views and acknowledges the alarm and indicates to the operator when a safety related message has been

received.

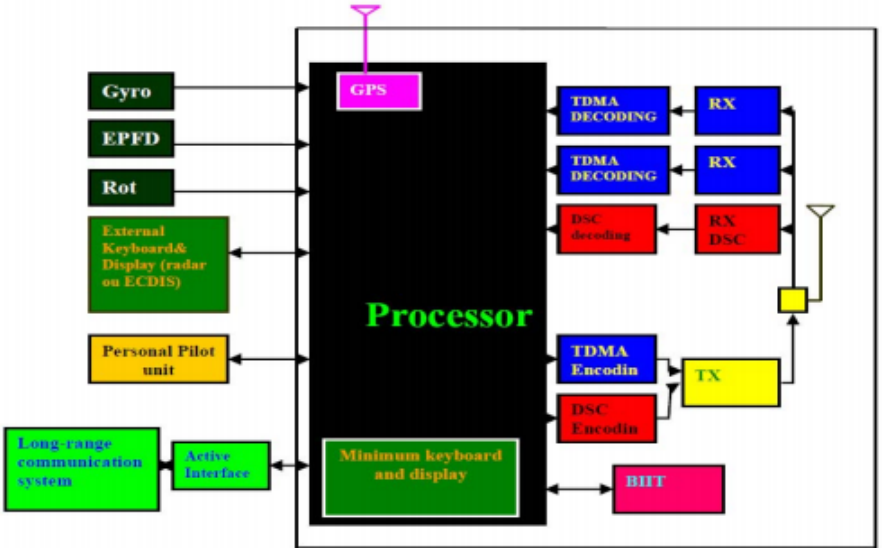


Figure 2.2. Diagram of a ship borne mobile AIS station

Specific issues for Class B Ship borne Mobile AIS stations The AIS class B can transmit the message 18 for the position report including the MMSI (Maritime Mobile Service Identity which is a series of nine digits which are sent in digital form over a radio frequency channel) of the vessel and the message 19 which gives supplementary information about the ship like its dimension and type. In addition, Messages with identifier 24 (defined by IEC) provides other information about the ship e.g. its name. This message is not readable by class A equipment but can be read by other class B CSTDMA and suitably equipped Base Stations. The IEC standard indicates a specific report rate for class B between 5s and 3 min.

Specific issues for Class A Ship borne Mobile AIS stations The AIS class A can transmit the message 1, 2 or 3 for the position report and the message 5 for information of ship according to a report rate which depends on the state and on the speed over ground of the ship. Concerning its composition as shown in Fig.2.2 In addition to the external GPS, it requires an internal differentiate GPS as a tool of position provider emergency, it also requires many sensors as heading, gyro

compass and rate of turn provider, a minimum keyboard display, an interface with the personal pilot unit and an active interface with long range communication.

AIS Data Communication AIS can transfer data via binary messages and Provides provides a means to encode, decode and send messages.

AIS is a collaborative navigational aid system that allows the identification Automatic naval units equipped with special transponders VHF, made mandatory on particular naval units and certain sizes, from Chapter 5 of the London Convention November 1, 1974 Safety of Life at Sea (SOLAS).

As shown in Fig.2.3 a typical AIS is composed of a VHF transmitter, Three receivers VHF (AIS1 AIS2, the third reserved for the reception of DSC messages,An antenna and from the connecting link with ,Satellite receiver which allows timing and location (eg. GPS) On-board sensors, and Visual display for the information received.

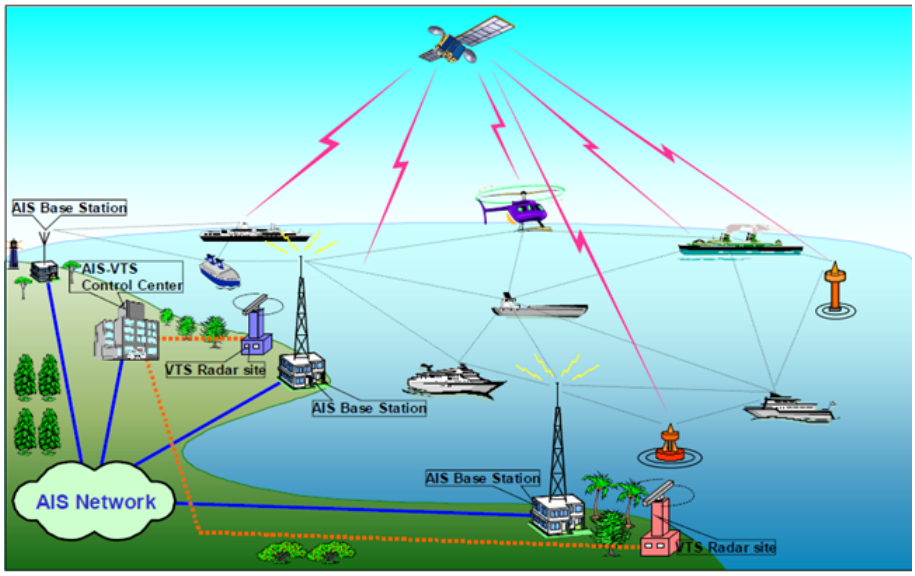


Figure 2.3. AIS system

AIS Data transmission The transmitted data is encoded using a GMSK modulation [2]. Data on static information is transmitted every six minutes while the dynamic information is sent with a frequency that varies depending on the vessel's speed as shown in Table 2.1.

Ship's dynamic conditions	Reporting interval
ship at anchor or moored and not moving faster than 3 knots	3 min
ship at anchor or moored and moving faster than 3 knots	10 s
ship 0-14 Knots	10 s
ship 0-14 Knots and changing course	3 1/3 s
ship 14-23 Knots	6 s
ship 14-23 Knots and changing course	2 s
ship >23 Knots	2 s
ship > 23 Knots and changing course	2 s

Table 2.1. AIS Data Transmission (GMSK mod.)

Two receivers for transmitted data through the channels and AIS1, AIS2, the third reserved for the reception of DSC messages (Digital selective calling) which is an important safety feature that can be found in the apparatus of marine VHF transmission that operates on channel 70. By pressing a single button (usually red) the system will transmit the identification of the boat together with the location if the system is connected to a GPS receiver.

The architecture of the AIS network, In Italy the AIS network is based on 50 coastal base stations (AIS-BS) that ensure data acquisition continuously transmitted by vessels equipped with an AIS transponder. Each BS is controlled by a logic device station, essentially consisting of a server with a software process that is responsible for processing the data collected by PSS(power system stabilizer) acting as an interface to and from the applications. Each edge server, which can control one or more physical drives, in addition to the collection, processing and storage of AIS

[2]K. Murota and K. Hirade, "GMSK Modulation for Digital Mobile Radio Telephony," in IEEE Transactions on Communications, vol. 29, no. 7, pp. 1044-1050, July 1981. doi: 10.1109/TCOM.1981.1095089

data, also takes care of networking, access and transfer data to the central (national) server which is responsible for the monitoring of maritime traffic and therefore plays the role of AIS Service Management (As shown in figure 2.4)

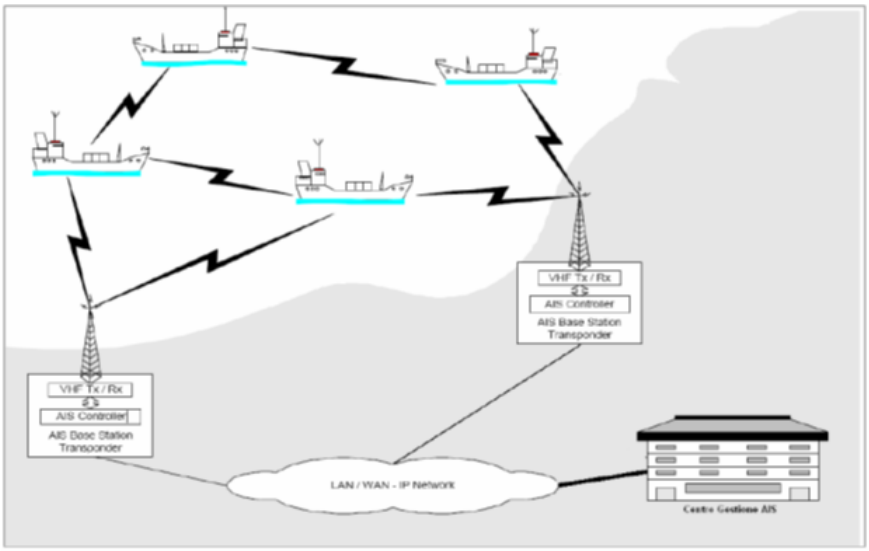


Figure 2.4. AIS Service Management

Three types of information can be exchanged with AIS messages namely static information (ship's name, MMSI number, size and type of the ship), information (location, speed and route), journey's information (presence of danger goods, destinations, expected time of arrival).

The AIS system board sends a series of messages by means of information containing various fields and internationally standardized by regulations ITU-R M.1371.

In this work, we consider the extended vessel positions message from [25]. The destination field is 20-char long, Using 6-bit ASCII(Identifier of message type) and Country code and Names from our Data Base as a reference (more detail figure 2.5).

2.1.4 ITU-R M.1371 and Message Type

It is a recommended protocol which provides the technical characteristics of an automatic identification system (AIS) using time division multiple accesses in the very high frequency (VHF) maritime mobile band. AIS stations should provide

static, dynamic and voyage-related data using this protocol. In the following we briefly describe message types and reporting interval.

1. Short safety related messages

   - Class A: ship borne mobile equipment should be capable of receiving and transmitting short safety related messages containing important navigational or important meteorological warning.
   - Class B :ship borne mobile equipment should be capable of receiving short safety related messages.

   A Class A ship-borne mobile station can be interrogated for message identifiers 3 and 5, by another station and A Class B ship-borne mobile station can be interrogated for message identifiers 18 and 19, by another station.

2. Reporting interval:

   - Static information: Every 6 min or, when data has been amended, on request.
   - Dynamic information: Dependent on speed and course alteration. Every 3 min for long-range broadcast message.
   - Voyage related information: Every 6 min or, when data has been amended, on request. Safety related message: An example is shown in Fig.2.6 for message type 5.

Message 5 format Message 5 format: Ship static and voyage related data should only be used by Class A ship borne and SAR aircraft AIS stations when reporting static or voyage related data. In Fig.2.6 we report an example for message type 5 which refers to ship static and voyage related data. This data should be used only by Class A ship borne, SAR aircraft and AIS stations. This information is sent every 6 minutes.

Paramètre	Number of bits	Description
Message ID	6	Identifier for this Message 5
Repeat indicator	2	Used by the repeater to indicate how many times a message has been repeated. Refer to § 4.6.1, Annex 2; 0-3; 0 = default; 3 = do not repeat any more
User ID	30	MMSI number
AIS version indicator	2	0 = station compliant with Recommendation ITU-R M.1371-1 1 = station compliant with Recommendation ITU-R M.1371-3 (or later) 2 = station compliant with Recommendation ITU-R M.1371-5 (or later) 3 = station compliant with future editions
IMO number	30	0 = not available = default – Not applicable to SAR aircraft 0000000001-0000999999 not used 0001000000-0009999999= valid IMO number; 0010000000-1073741823 = official flag state number.
Call sign	42	7 x 6 bit ASCII characters, @@@@@@@ = not available = default. Craft associated with a parent vessel, should use "A" followed by the last 6 digits of the MMSI of the parent vessel. Examples of these craft include towed vessels, rescue boats, tenders, lifeboats and liferafts.
Name	120	Maximum 20 characters 6 bit ASCII, as defined in Table 47 "@@@@@@@@@@@@@@@@@@@@" = not available = default. The Name should be as shown on the station radio license. For SAR aircraft, it should be set to "SAR AIRCRAFT NNNNNNN" where NNNNNNN equals the aircraft registration number.
. . .		
ETA	20	Estimated time of arrival; MMDDHHMM UTC Bits 19-16: month; 1-12; 0 = not available = default Bits 15-11: day; 1-31; 0 = not available = default Bits 10-6: hour; 0-23; 24 = not available = default Bits 5-0: minute; 0-59; 60 = not available = default For SAR aircraft, the use of this field may be decided by the responsible administration
Maximum present static draught	8	In 1/10 m, 255 = draught 25.5 m or greater, 0 = not available = default; in accordance with IMO Resolution A.851 Not applicable to SAR aircraft, should be set to 0
Destination	120	Maximum 20 characters using 6-bit ASCII; @@@@@@@@@@@@@@@@@@@@ = not available For SAR aircraft, the use of this field may be decided by the responsible administration
DTE	1	Data terminal equipment (DTE) ready (0 = available, 1 = not available = default) (see § 3.3.1)
Spare	1	Spare. Not used. Should be set to zero. Reserved for future use
Number of bits	424	Occupies 2 slots

Figure 2.5. Partial Message type 5

2.2 Open challenges and issues

As previously described, one of the main issues of the AIS architecture is that the destination field may be compiled in many different ways. The same destination port in fact may be written in different ways according to an arbitrary user decision. An example is shown in Fig.2.6 where the same destination port (Genoa, Italy) is written in two different ways (with and without the letter 'v'). Such variety of information does not allow to carry out checks and statistics in automated manner to capture information on ship traffic.

Nowadays AIS became an important source of information about the marine traffic for national and regional monitoring networks. This also resulted in establishing in the Community a vessel traffic and information (HELCOM) with the view to enhancing the safety and efficiency of maritime traffic improving the response of authorities to dangerous situations at sea. Special attention has been paid to traffic organization, contributing to a better prevention and detection of environmental pollution by ships. Coordinated traffic surveillance includes also search and rescue actions as well as security improvement aspect With a set of static, dynamic and voyage-related data broadcasted through an AIS signals a mariner's can see a target ship's location(Source and destination), name, size, draught, navigational status, speed and course over ground, heading, rate of turn and able to identify that vessel or to agree maneuvers.

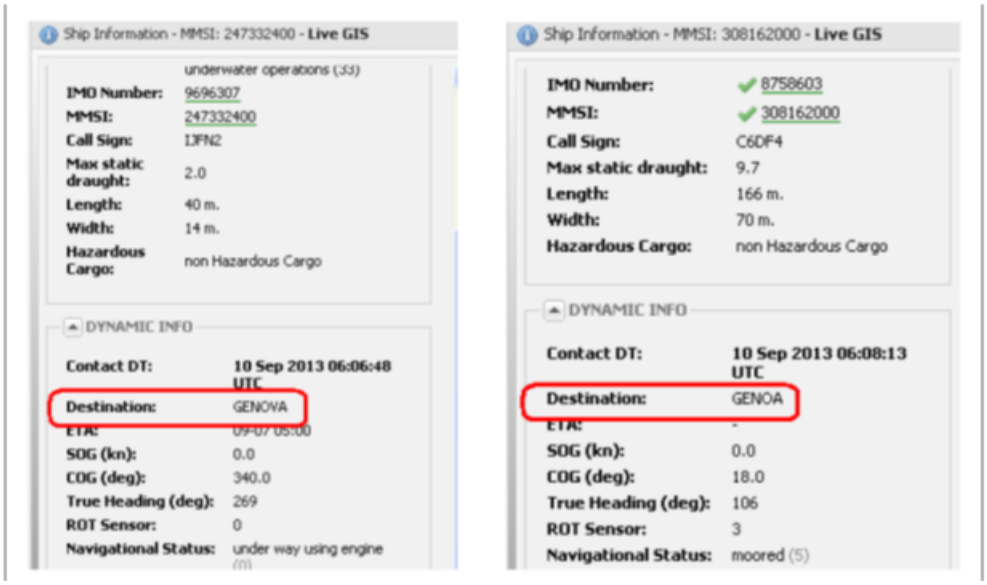
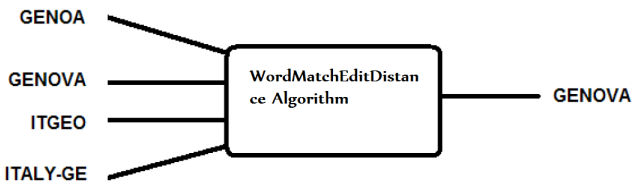


Figure 2.6. An example of port name coding in different ways



Algorithm.png

Figure 2.7. An example of different coding for the same port name

Vulnerability of AIS AIS protocol was designed with seemingly zero security considerations, these are the major issues:

- Lack of Validity Checks. It is possible to send an AIS message from any location for a vessel at another location e.g. you can send a message from a location near New York for a vessel that claims to be in the Gulf of Mexico, and it will be accepted without question. No geographical validity checks are carried out.

- Lack of Timing Checks. It is also possible to replay existing (valid) AIS information, because no time stamp information is included in the message e.g. you can replicate the position of a vessel.

- Lack of Authentication. There is no authentication built into the AIS protocol. That means that anyone who can craft a AIS packet can impersonate any other vessel on the planet, and all receiving vessels will treat the message as fact.
- Lack of Integrity Checks. All AIS messages are sent with out encryption and unsigned form, making them trivial to intercept and modify.

2.3 Possible Approach

2.3.1 Algorithm for distance metrics

Available String Distance algorithm

In order to check the distance between two strings (e.g. Genoa and Genova) different approach and metrics have been proposed in literature. The Hamming distance evaluates the distance between two strings of equal length as the number of positions for which the corresponding symbols are different. This distance evaluation can be easily implemented but it is worth noting that in the case of this work strings may have different length. The Levenshtein distance (also called the edit distance) is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character and finally the Damerau-Levenshtein distance is an extension of the previous one that counts transposition as a single edit operation. Strictly speaking, the Damerau-Levenshtein distance is equal to the minimal number of insertions, deletions, substitutions and transpositions needed to transform one string into the other.

2.3.2 Regular Expressions

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. It can be used to search, edit, or manipulate text and data.

Chapter 3

Proposed Implementation

The developed system use MVC framework in Java application development for desktop applications, the basic programs to enterprise solutions written in Java.

MVC framework is used to separate the data access layer, business logic code and the graphical user interface that has to be defined and designed to let the user interact with the application. This application has three parts:

- Model
  It is adopted to store the data of the application, such as databases, text data, files and/or other resources.
- View
  This is the graphical user interface of the application which contains buttons, text boxes and other controls.
- Controller
  The actual back-end code constitutes the controller of the framework. A controller controls the data coming from the users, or going to the user from a model.

3.1 Architecture for the System

A very good feature of the MVC framework which is the system architecture is that it hides the data access layer from the users. That is, the data access layer or the

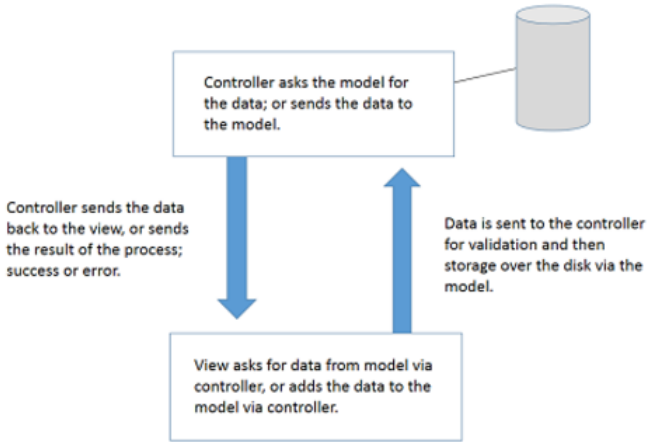


Figure 3.1. shows MVC pattern.

data is never actually called directly by the user from the interface. In this way, to access data, the user has only a limited set of action he can perform. This feature allows the developers to create groups or roles of users that are allowed to access the data such as Admins, Guests etc. Another good thing about this framework is

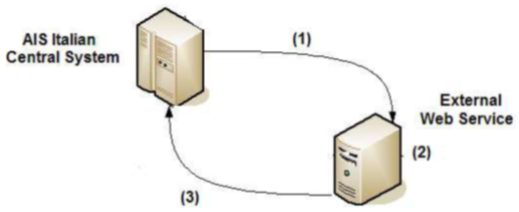


Figure 3.2. Overview of Expected system at High level

that it doesn't let the application get so complicated, and all the three segments of the application interfering with each other in a single source code package.

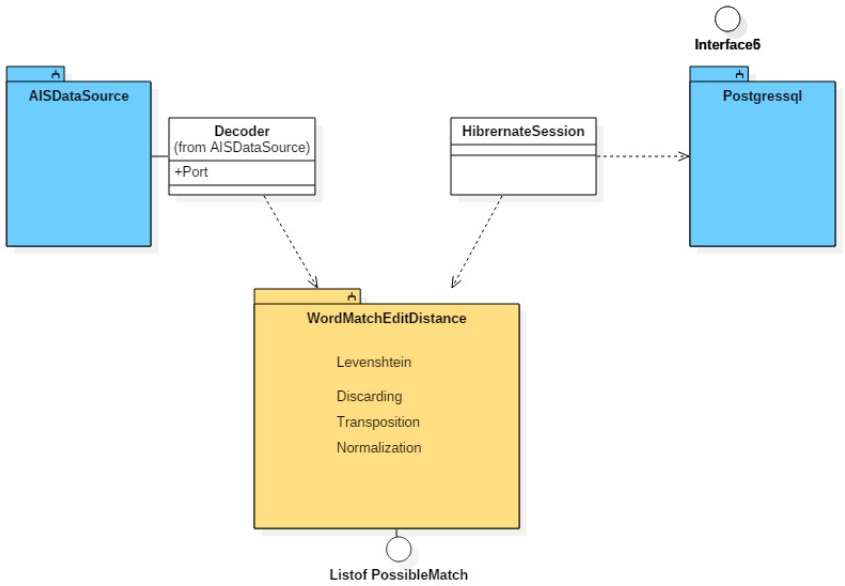


Figure 3.3. Implemenation Overview

3.2 Decoding AIS message

AIS data is binary code that can be extracted by the Standard of NMEA . It uses two sentences AIVDM (Accepted Data from other vessels) and AIVDO (ship own information). The AIS transmitters periodically broadcast their position and course using TDMA (Time Division Multiple Access). The interested reader can refer to SOLAS [1] for further details. AIS receivers report ASCII data packets as a byte stream over serial or usb lines, using the NMEA 0183 or NMEA 2000 data formats. [2] The RS422 variant of serial specified as a physical layer by NMEA 0183 is common in marine navigation systems there may be a pilot plug which converts to usb. Alternatively, newer AIS receivers may report directly over RS232 or usb.
AIS packets have the introducer AIVDM or AIVDO; AIVDM packets are reports from other ships and AIVDO packets are reports from your own ship.
AIS consists of several types of records and the messages are divided into four file

[1] http://www.imo.org/en/about/conventions/listofconventions/pages/international-convention-for-the-safety-of-life-at-sea-(solas),-1974.aspx
[2] http://www.gpsinformation.org/dale/nmea.htm

types(as shown in figure 3.4) Some of them are about the position of the vessel (where the message id is 1, 2, 3 or 21) whereas others are about voyage related issues (where the message id is 5). The third is all other AIS messages which is possible to decode, and the fourth is data that we could not decode. The records were written to different sets of files but for our purpose we only want the position of the vessel and the others to the voyage of the vessel.

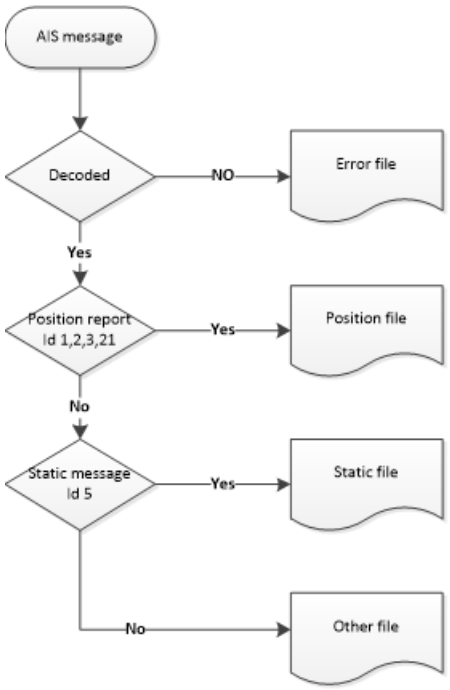


Figure 3.4. Decoding AIS messages into four types of files

AIS Standards and Information Sources The IALA(International Association of Marine Aids to Navigation and Lighthouse Authorities) Technical Clarifications at IALA and the Coast Guard's AIS pages at NAVCEN [3] describe AIS message payloads type 1-24 almost completely. Certain specialized binary messages of types 6 and 8 defined by the International Maritime Organization are described in IMO236 and IMO289 [4] The detail information on payload formats in this document

[3]https://www.navcen.uscg.gov/
[4]https://www.e-navigation.nl/content/imo-documents

is mostly derived from these public sources.
Kurt Schwehr is a research scientist at the Center for Coastal and Ocean Mapping at the University of New Hamphire. His work blog at Schwehr [5] contains sample messages and descriptions of AIS operation in the wild that shed light on various obscure corners of the specification. He also communicated some critical information from IEC-PAS, and supplied information about new messages and fields in ITU-1371-3.

Schwehr includes links to a collection of Python scripts for decoding and analyzing AIVDM sentences. Kurt Schwehr warns that this is research code rather than a production tool.There is also a project on [GNU AIS] [6] at SourceForge ,in this work, a Java JAR utility to decode AIS messages in NMEA format has been developed(sample message is listed below)

```
!AIVDM,1,1,,B,177KQJ5000G?tO‘K>RA1wUbN0TKH,0*5C
!AIVDM,1,1,,A,18UG;P0012G?Uq4EdHa=c;7@051@,0*53
!AIVDM,1,1,,A,Dh3OvkOnIN>4,0*38
```

into Java objects with easily accessible properties. AIS messages comes with three different classes to receive NMEA encapsulated AIS messages: one of them is the NMEAMessageSocketClient which demonstrates decode AIS messages by connecting to a TCP/IP socket and starting to receive and decode.It is available in the Maven Central repository.

AIVDM/AIVDO Sentence Layer has two-layer protocol. The outer layer is a variant of NMEA 0183 [NMEA], the ancient standard for data interchange in marine navigation systems.

AIVDM data packet: In the following we describe the meaning of a received string see [7]for further detail

[5]http://vislab-ccom.unh.edu/ schwehr/papers/2010-IMO-SN.1-Circ.289.pdf
[6]http://gnuais.sourceforge.net/[GNU AIS]
[7]http://catb.org/gpsd/AIVDM.html

```
AIVDM,1,1,,B,177KQJ5000G?tO`K>RA1wUbN0TKH,0*5C
```

Field 1 AIVDM, identifies this as an AIVDM packet.

Field 2 (1 in this example) is the count of fragments in the currently accumulating message. The payload size of each sentence is limited by NMEA 0183's 82-character maximum, so it is sometimes required to split a payload over several fragment sentences.

Field 3 (1 in this example) is the fragment number of this sentence. It will be one-based. A sentence with a fragment count of 1 and a fragment number of 1 is complete in itself.

Field 4 (empty in this example) is a sequential message ID for multi-sentence messages.

Field 5 (B in this example) is a radio channel code. AIS uses the high side of the duplex from two VHF radio channels: AIS Channel A is 161.975Mhz (87B); AIS Channel B is 162.025Mhz (88B). In the wild, channel codes 1 and 2 may also be encountered, the standards do not prescribe an interpretation of these but it's obvious enough.

Field 6 (177KQJ5000G?tO`K¿RA1wUbN0TKH which is the data payload.

Field 7 (0) is the number of fill bits requires to pad the data payload to a 6 bit boundary, ranging from 0 to 5. Equivalently, subtracting 5 from this tells how many least significant bits of the last 6-bit nibble in the data payload should be ignored. Note that this pad byte has a tricky interaction with the [ITU-1371] requirement for byte alignment in over-the-air AIS messages; see the detailed discussion of message lengths and alignment in a later section.

The symbol * do separated suffix (*5C) is the NMEA 0183 data-integrity checksum for the sentence, preceded by "*". It is computed on the entire sentence including the AIVDM tag but excluding the leading "!".

By means of the Java library for AIS message, it is possible to decode in specific message type 5 which includes ship static and voyage related data. These messages are usually used only in Class A ship borne and SAR aircraft AIS stations when reporting static or voyage related data.

3.2.1 Edit Distance

Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other.

Edit distances find applications in natural language processing, where automatic spelling correction can determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low distance to the word in question. In bioinformatics, it can be used to quantify the similarity of DNA sequences, which can be viewed as strings of the letters A, C, G and T.

The edit distance dis(x, y) of two strings x and y is a string similarity measure, which is defined by the minimum number of edit operations to make x be identical to y, where the edit operations are inserting, deleting or substituting one character. The edit distance is a basic concept for many applications such as noises in signal processing and mutations in genome or protein sequences, for which generalized versions of the edit distance are employed

Different definitions of edit distance allow different sets of string operations. For instance:

- The levenshtein distance allows deletion, insertion and substitution. Being the most common metric, the levenshtein distance is usually what is meant by "edit distance".
- The longest common subsequence (LCS) distance allows only insertion and deletion, not substitution. The Hamming distance allows only substitution, hence, it only applies to strings of the same length.
- The damerau levenshtein distance allows insertion, deletion, substitution, and the transposition of two adjacent characters.
- The Jaro distance allows only transposition.

Classical Levenshtein Distance As described in A modified edit-distance algorithm for record linkage in a database of companies [8]

[8]Woltzenlogel Paleo, Bruno and Ghedini, Cinara and Lima, Joubert and Lima, C and H C Ribeiro, Carlos and Filho, Jorge. (2006). A modified edit-distance algorithm for record linkage in a database of companies.

In the Levenshtein Distance, the allowed operations are insertion, deletion and substitution of characters, all of them with cost 1. For example, the distance between "Parq" and "Parma" is equal to 2, because it's necessary to substitute 'q' by 'm' and insert 'a'.

Levenshtein Distance has been widely accepted in part grouping as the distance measure between strings . It is defined as the minimum number of inserts, deletes and substitutions needed to change source operation sequence into target one. When comparing two operation sequences, the most obvious type of difference between them is the substitution of one operation for another at the same position in the sequence. Such differences are called substitutions or swaps. There are other important types of differences, however, such as deletion of operations and insertion of operations. Dealing with differences between sequences due to substitution, deletion and insertion, is the central theme of operation sequence comparison

3.3 Java - Regular Expressions

Regular expressions are a language of string patterns built in to most modern programming languages, including Java 1.4 onward; they can be used for: searching, extracting, and modifying text.

Java provides the java.util.regex package for pattern matching with regular expressions. Java regular expressions are very similar to the Perl programming language and very easy to learn.

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. They can be used to search, edit, or manipulate text and data.
Regular expressions are both terribly awkward and extremely useful. Their syntax is cryptic, and the programming interface JavaScript provides for them is clumsy. But they are a powerful tool for inspecting and processing strings.

Usage of Regular Expression When it comes to searching unstructured data, regular expressions are a very useful and powerful tool. The power provided by popular regular expression libraries does come with a significant performance

cost in some cases though, both when compiling regular expressions into automata and when using these automata to match input. These constraints are usually acceptable for individuals needing to extract information from data sets located on personal computers or internal servers. The value of being able to extract the right information is often more important than the time it takes to search for it. However when search capabilities are offered as a service to thousands of users, constraints get much tighter: service maintainers cannot afford to let any user perform searches involving very inefficient algorithms as this may cause the service to become unresponsive to all users.
Re2 is a regex library that avoids the state explosion problem as deterministic automata are not required to be built. It also provides guarantees regarding performance of searching with regex by limiting the set of available features (for instance, backreferences are not possible with Re2). This approach is sill very practical though as the set of features offered still allows for interesting queries to be performed. Traditionally Logentries used a JNI wrapper over Re2 in order to integrate it into its Java code base. Recently a library called Re2J, which is a port of Re2 to Java, was released and has now replaced Re2 Java in Logentries' code base as it has proven to be faster at matching events than the JNI calls made by Re2 Java (see below for performance comparison below).
Finally Brics is a regex library which takes advantage of deterministic automata in order to perform fast regex matching.

Re2-Java Vs Re2J Vs Brics In terms of features, Re2J and Re2 have similar sets. Some of these features are not supported by Brics such as special escaped characters (e.g. " d"," s", etc). In terms of safety of use, Re2−Java and Re2J both avoid the state explosion problem while Brics does not. Finally regarding performance while matching events/input, Brics outperforms Re2−Java and Re2J, especially when considering large datasets(see fig 3.5).

These comparisons were performed on regular expressions that all 3 libraries can handle and that are typical to Logentries customers. So even though Figure 1 data is respective to specific queries and not fully general, it does show the trend that was mostly observed:

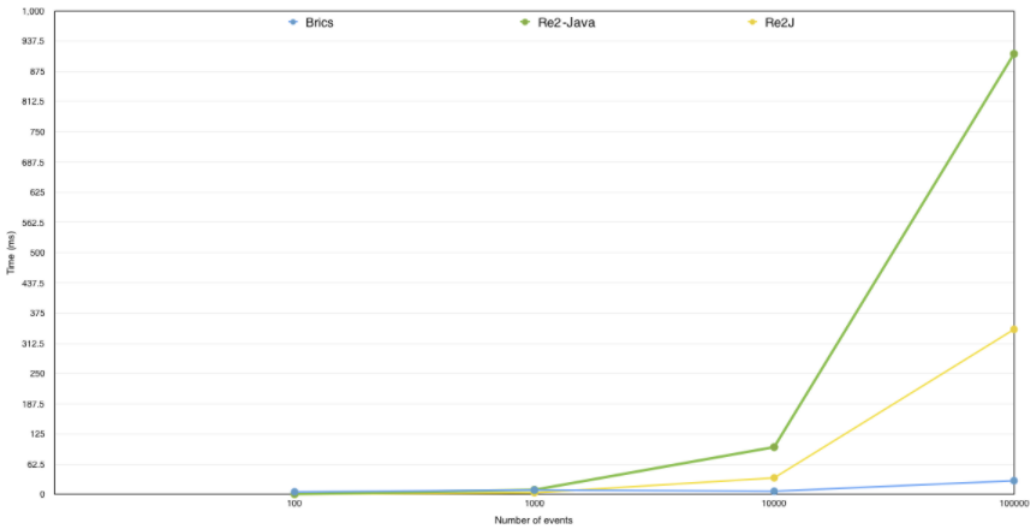


Figure 3.5. performance while matching events/input, Brics outperforms Re2-Java and Re2J, especially when considering large datasets.

- Re2-Java, Re2J and Brics tend to perform similarly on small datasets, i.e. less than 1000 events.
- Differences in performance start being significant when considering ten's of thousands of events, with Brics outperforming Re2J, itself outperforming Re2 Java.

When applying regular expressions on large sequence of events, many events may not match the regex. next Figure 3.8 shows the result of comparing Re2−Java, Re2J and Brics when no events are matched. The trend is similar to the one observe in above Figure 3.5 , with Brics being faster than Re2J, which is itself faster than Re2−Java.

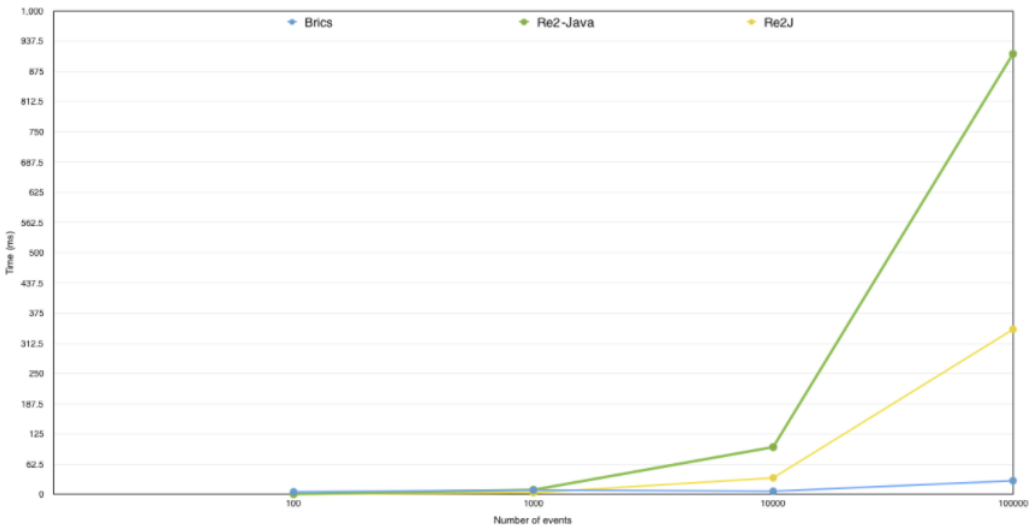


Figure 3.6. shows the result of comparing Re2-Java, Re2J and Brics when no events are matched. .

From the above figures (Figure 3.8 and 3.9), Re2J appears to be more performant than Re2−Java for a similar feature set and also Brics appears to be a good choice when it comes to performance while matching large sequences of events. However it suffers from the state explosion problem by building deterministic finite state machines and is not as feature complete as Re2J and Re2−Java. So in order to take advantage of Brics whenever possible, the following approach can be considered for a given regular expression that fits within Brics feature set:

1: Try to compile the regex with Brics.

2: Abort if the number of states of the automata being built goes beyond a given threshold(and go to Point 4).

3: If the size of the automata remains within the threshold, use the automata to perform matching (benefiting from Brics matching performance).

4: If the size of the automata does not remain within the threshold, then fall back onto using Re2.

Of course, for regular expressions that cannot be handled by Brics, Re2J can then be used.

In order to implement this approach, two questions remain to be answered:

1 How to determine what threshold to be used?

2 How to determine whether a regular expression fits within Brics feature set?

For Question 1, lets considered typical regular expressions used by our users and compiled the ones that fit within Brics feature set, while monitoring the maximal size of the automata being built during this process. next Figure (fig 3.10) illustrates the distribution that can be obtained.

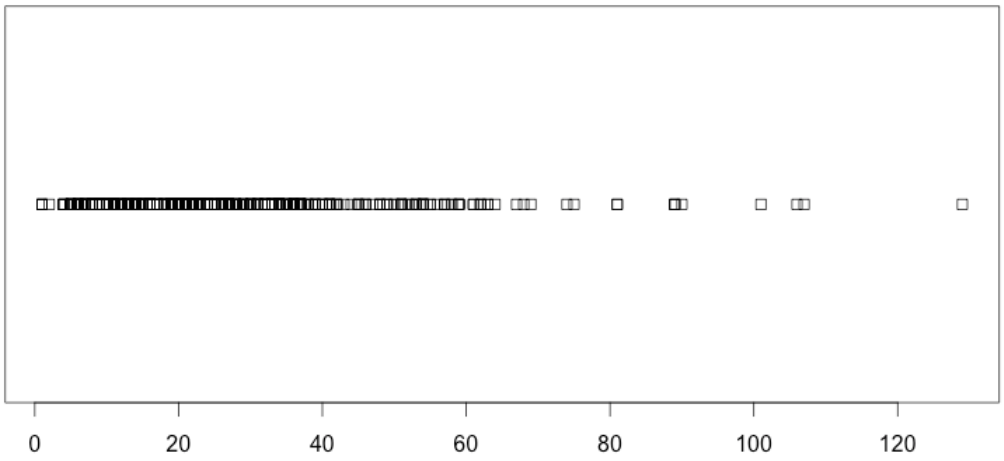


Figure 3.7. shows that a threshold of 80 states would capture for most of users use cases.

Question 2 is answered by leveraging the regex parser. Search queries are parsed using ANTLR. Beside generating both a Java and Javascript parser from one single language grammar, ANTLR also makes it much easier to determine whether the regular expression being parsed is valid and which features it contains. For instance considering a PCRE grammar, special escaped characters are detected and the use of Brics to apply the provided regular expression can be ruled out in this case as we know it does not handle such characters.

A primary concern when providing a service where users can perform searches using regular expression is to limit it to "safe" cases where the state explosion problem (inherent to automata computing deterministic automata) is avoided as well as slow matching algorithms. A Java library such as Re2J ensures such criteria. However, when applying regular expressions to millions or even billions of events, another level of performance is needed. Brics is a Java library that provides fast event matching. However, unlike Re2J, Brics may be subject to the state explosion problem. It also provides a more limited set of features. A possible strategy in this case is to consider two regex libraries and take advantage of a ANTLR parser in order to detect which library should be used for the regular expression submitted, performing fast matching when possible while having the ability to always fall back onto a safe (still quite performant) option, Safety first!

The java.util.regex package primarily consists of the following three classes

Pattern Class

A Pattern object is a compiled representation of a regular expression. The Pattern class provides no public constructors. To create a pattern, you must first invoke one of its public static compile() methods, which will then return a Pattern object. These methods accept a regular expression as the first argument.

Matcher Class

A Matcher object is the engine that interprets the pattern and performs match operations against an input string. Like the Pattern class, Matcher defines no public constructors. You obtain a Matcher object by invoking the matcher() method on a Pattern object.

PatternSyntaxException

A PatternSyntaxException object is an unchecked exception that indicates a syntax error in a regular expression pattern.

Basic Handling of String In pattern

The Java language includes a primitive data type char, which holds a 16-bit unicode character. You can hold multiple characters in a String object, or in a StringBuffer object.
Methods such as equals and equalsIgnoreCase, startsWith and endsWith allow you to test Strings against one another. Methods such as indexOf and substring allow you to perform operations on a String. parseInt and other similarly named methods allow you to extract a number (in this example an int) from a String, although you do need to remember to catch the exception that may be thrown.
In certain specialist applications, such as Bioinformatics, multiple characters can also be usefully held in a char array, where they're likely to be dealt with character-by-character in a loop, as in DNA and RNA sequencing.

The StringTokenizer class allows you to take a String and step through it element-by-element (token-by-token) to handle it in chunks or sections. You can choose what character or characters you use between the elements to break up the string in the way you want.
But, until Java release 1.4, the standard classes didn't include any way to ask "does this string look like xxxxxxx". Why would we want to? Well, we might want to ask, "Does this string look like an email address?" and go on to define (in simple terms) an email address as a series of non-spaces, followed by an @ character, followed by another series of non-spaces. Let's see how that is solved in Java 1.4:

java Reg1 email graham@wellho.net or
lisa@wellho.net for information
"email" is NOT an email address
"graham@wellho.net" IS a possible email address
"or" is NOT an email address
"lisa@wellho.net" IS a possible emailaddress
"for" is NOT an email address
"information" is NOT an email address

3.4 Hibernate ORM

ORM or Object Relational Mapping uses objects as a representation of relational databases.It is concerned with data persistence as it applies to relational databases.

3.4.1 Benefits of Using Hibernate

- persistence: Hibernate ORM is the best way to achieve persistence.

- ORM: you will map a database table with java object called "Entity". So once you map these, you will get advantages of OOP concepts like inheritance, encapsulation.

- Caching mechanism :

  provided hibernate means no need to hit database for similar queries, it is possible to cache it and use it from buffered memory to improve performance and productivity.

- Lazy loading: Supports Lazy loading (also called n+1 problem in Hibernate). Take an example-parent class has n number of child class. So When you want information from only 1 child class, there is no meaning of loading n child classes. This is called lazy loading (Load only thing which you want).

- Maintainability:

  It helps reduce the lines of code, makes system more understandable and emphasizes more on business logic rather than persistence work (SQLs). More important, a system with less code is easier to refactor.

- Portability:

  It abstracts our application away from the underlying SQL database and sql dialect. Switching to other SQL database requires few changes in Hibernate configuration file (Write once / run-anywhere).

3.4.2 Hibernate Sessions

A Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database. Persistent objects are saved and retrieved through a Session object. The session objects should not be kept open for a long time because they are not usually thread safe and they should be created and destroyed them as needed. The main function of the Session is to offer create, read and delete operations for instances of mapped entity classes. Instances may exist in one of the following three states at a given point in time:

- transient: A new instance of a persistent class which is not associated with a Session and has no representation in the database and no identifier value is considered transient by Hibernate.
- persistent: You can make a transient instance persistent by associating it with a Session. A persistent instance has a representation in the database, an identifier value and is associated with a Session.
- detached: Once we close the Hibernate Session, the persistent instance will become a detached instance.

A Session instance is serializable if its persistent classes are serializable.

```
 Session session = factory.openSession();
 Transaction tx = null;
try {
   tx = session.beginTransaction();
  do some work
    ...
   tx.commit();
   }
   catch (Exception e) {
if (tx!=null) tx.rollback();
e.printStackTrace();
 }
```

```
 finally {
   session.close();
}
```

The entire concept of Hibernate is to take the values from Java class attributes and persist them to a database table. A mapping document helps Hibernate in determining how to pull the values from the classes and map them with table and associated fields.

Java classes whose objects or instances will be stored in database tables are called persistent classes in Hibernate. Hibernate works best if these classes follow some simple rules, also known as the Plain Old Java Object (POJO) programming model. There are following main rules of persistent classes, however, none of these rules are hard requirements.

- All Java classes that will be persisted need a default constructor.
- All classes should contain an ID in order to allow easy identification of your objects within Hibernate and the database. This property maps to the primary key column of a database table.
- All attributes that will be persisted should be declared private and have getXXX and setXXX methods defined in the JavaBean style.
- A central feature of Hibernate, proxies, depends upon the persistent class being either non-final, or the implementation of an interface that declares all public methods.
- All classes that do not extend or implement some specialized classes and interfaces required by the EJB framework.

The POJO name is used to emphasize that a given object is an ordinary Java Object, not a special object, and in particular not an Enterprise JavaBean check port and customport POJO class An Object/relational mappings are usually defined in an XML document

Hibernate Config file The file hibernate-cgf.xml which configure the data base connection and Hibernate concrete class.

```
   <?xml version="1.0" encoding="UTF
<property name="hibernate.connection.driver
     $-$class">org.postgresql.Driver</property>
  <property name="hibernate.connection.password">123456</property>
  <property name="hibernate.connection.url">
  jdbc:postgresql://localhost/AISdata</property>
      <property name="hibernate.connection.username">
  postgres</property>8"?>

   <!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
   <hibernate-configuration>
   <session-factory>
   <property name="hibernate.connection.driver$-$class">
   org.postgresql.Driver</property>
     <property name="hibernate.connection.password">123456</property>
<property name="hibernate.connection.url
">jdbc:postgresql://\\ localhost/AISdata</property>
 <property name="hibernate.connection.username">postgres</property>
 <!--  JDBC connection pool -->
 <property name=" connection.pool-size">5</property>


 <!-- Defines the SQL dialect used in Hiberante's application -->
 <property name="dialect">org.hibernate.dialect.MySQLDialect</property>

   <!-- Enable Hibernate's automatic session context management -->


  <!-- Disable the second-level cache  -->
<property name="cache.provider-class">org.hibernate.cache.NoCacheProvider</property>
```

```
 <!-- Display and format all executed SQL to stdout -->
 <property name="show-sql">true</property>
 <property name="format-sql">true</property>

 <!--  echo all excuted sql to stdout-->
  <!--  <property name="hbm2ddl.auto">validate</property> -->
  <!-- List of XML mapping files -->
   <mapping class="Model.Ports"/>
   <mapping class="Model.Customport"/>
   </session-factory>
</hibernate-configuration>
```

3.5 The Work Flow Of the Implementation

The hibernate start with Lazy loading (it is a design pattern which is used to defer initialization of an object as long as it it possible)and create TCP socket with incoming AIS message (sample of incoming date fig 3.8)

Sentence	MMSI	Message Type	DAC	FI	ID	Vessel Name	Comments
!ABVDM	255803250	1					Position Report Class A (Scheduled)
!ABVDM	247131500	1					Position Report Class A (Scheduled)
!ABVDM	247601000	1					Position Report Class A (Scheduled)
!ABVDM	247601000	1					Position Report Class A (Scheduled)
!ABVDM	247044910	1					Position Report Class A (Scheduled)
!ABVDM	247044910	1					Position Report Class A (Scheduled)
!ABVDM	271000766	1					Position Report Class A (Scheduled)
!ABVDM	255803250	1					Position Report Class A (Scheduled)
!ABVDM	227212000	3					Position Report Class A (Special)
!ABVDM	271000766	1					Position Report Class A (Scheduled)
!ABVDM	311003500	1					Position Report Class A (Scheduled)
!ABVDM	311003500	1					Position Report Class A (Scheduled)
!ABVDM	247370400	1					Position Report Class A (Scheduled)
!ABVDM	247042850	1					Position Report Class A (Scheduled)
!ABVDM	319023600	3					Position Report Class A (Special)
!ABVDM	378365000	3					Position Report Class A (Special)
!ABVDM	247044910	1					Position Report Class A (Scheduled)

Figure 3.8. Sample of Incoming AIS message.

select based on message Type (Message Type 5) since all the data we need is found there.

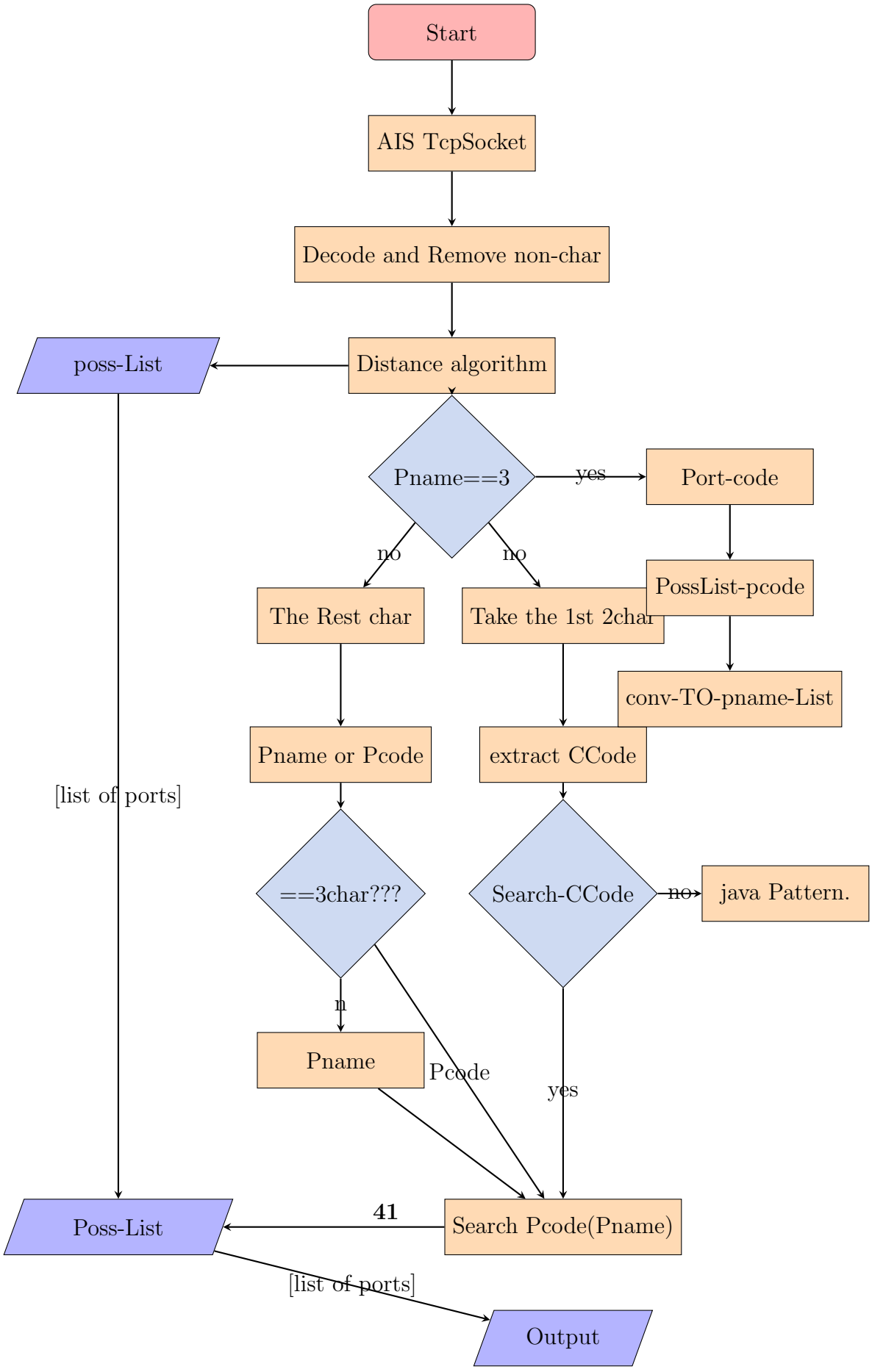
- Decode the Incoming AIS message using java free library(dk.tbsalling.aismessages) for further information simple demo applications in the [9]

-Clean the date by Removing non character and/or spaces etc.

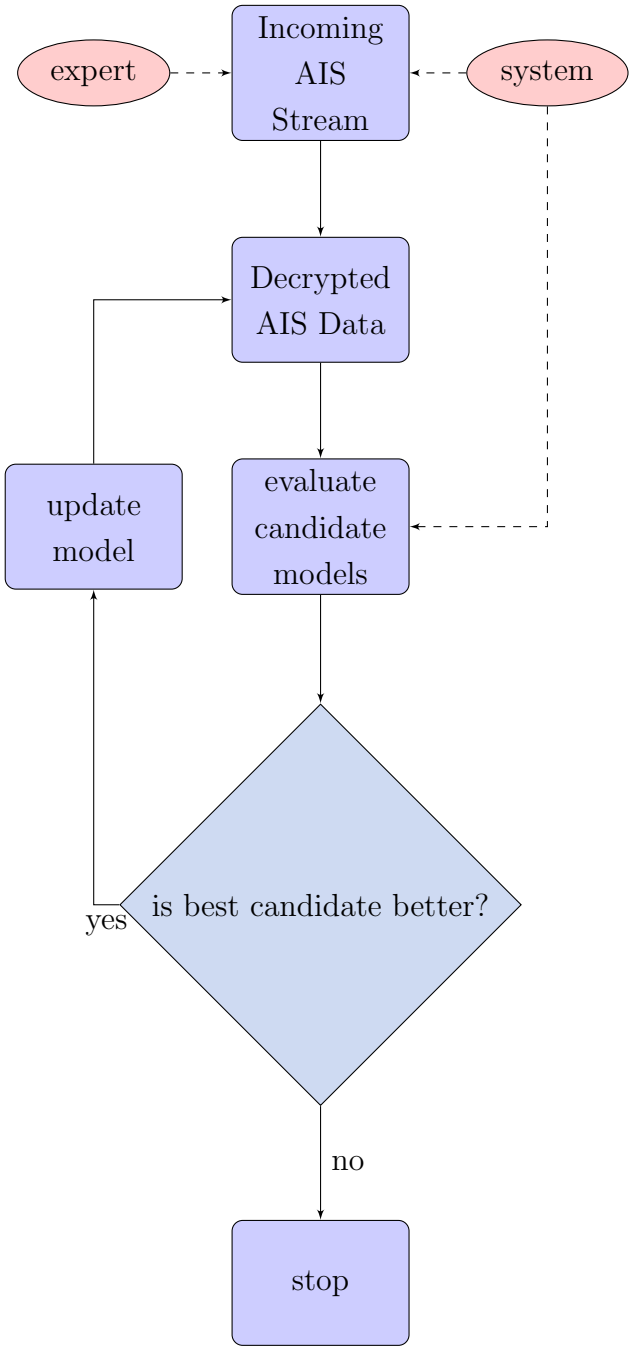
Now we have all the data(worlds port name from database and incoming data)and ready to apply Levenshtein or edit distance let denote incoming data(D) and data from the data base pname(port name), pcode(port code), ccode(country code). The first thing to do is that computing Distance with port name distance(D, pname), if distance is less than three put the result in the list(possible list of match found)and we keep searching if the length of D is three it is mostly likely a port code since max size of port code is three and find the distance(distance(D, pcode)).
If the length of D not three we will have two possibilities that D might be country code with port name or country code with port code. In order to check the county code take the first two character(all country code have two character) in order to search the country code from the data base as the same time the remaining string might be port code if only if it has three character other wise it is a port name, but having three character does not mean that it must be a port code the port name has also there character(ex. Gex, Lor in france) or user wrote it by mistake, in such cases we use two kinds of search the distance between them and java pattern, distance(remaining character, pcode)and distance(remaining character, pname)

Finally we have to sort and remove repeated name the one with higher distance will be removed and covert port codes to the corresponding name and update our lists for each different cases

[9]https://github.com/tbsalling/aismessages/tree/master/src/main/java/dk/tbsalling/aismessages/demo


Start
AIS TcpSocket
Decode and Remove non-char
poss-List
Distance algorithm
Pname==3
yes
Port-code
no
no
PossList-pcode
The Rest char
Take the 1st 2char
conv-TO-pname-List
Pname or Pcode
extract CCode
[list of ports]
==3char???
Search-CCode
no
java Pattern.
n
Pname
Pcode
yes
Poss-List
Search Pcode(Pname)
[list of ports]
Output

Our system will be used as a black box between the AIS system and the client(vessels/ships) which will listen the communication between to filter and provide the best service by updating the model(Decrypted AIS Data) as shown below.



Class Diagrams MVC separates out the application logic into three separate parts, promoting modularity and ease of collaboration and reuse. It also makes applications more flexible and welcoming to iterations

Control view Class diagram It contains logic which is the algorithm that updates the model and/or view in response to input from the users of the application or the Incoming AIS data. It includes the Hibernate utility for Session Factory and configuration setup and the most important class is the Compute Distance class which implement the algorithm and java pattern. as shown in figure 3.9.

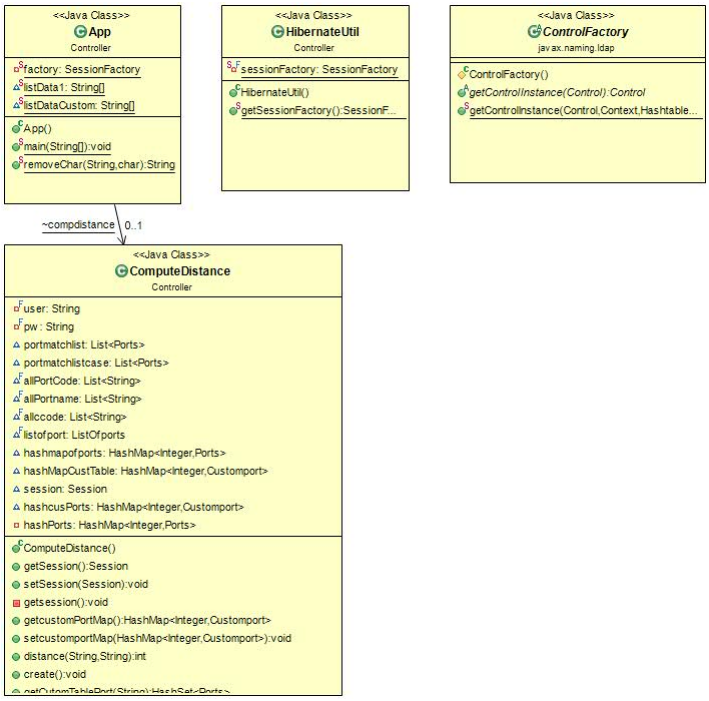


Figure 3.9. Control class diagram

Model Class diagram The Model is directly responsive for handling data. For example, the Model component accesses postgresql database. The Model should not rey on other components such as View or Controller. In other words, the Model does not care how its data can be displayed or when to be updated.

The data changes in the Model will generally be published through some event handlers. For example, the View model must register on the Model so that it understands the data changes. The most important task of the view as show in (figure 10.3) is to prepare list of port by lazy loading using hibernate

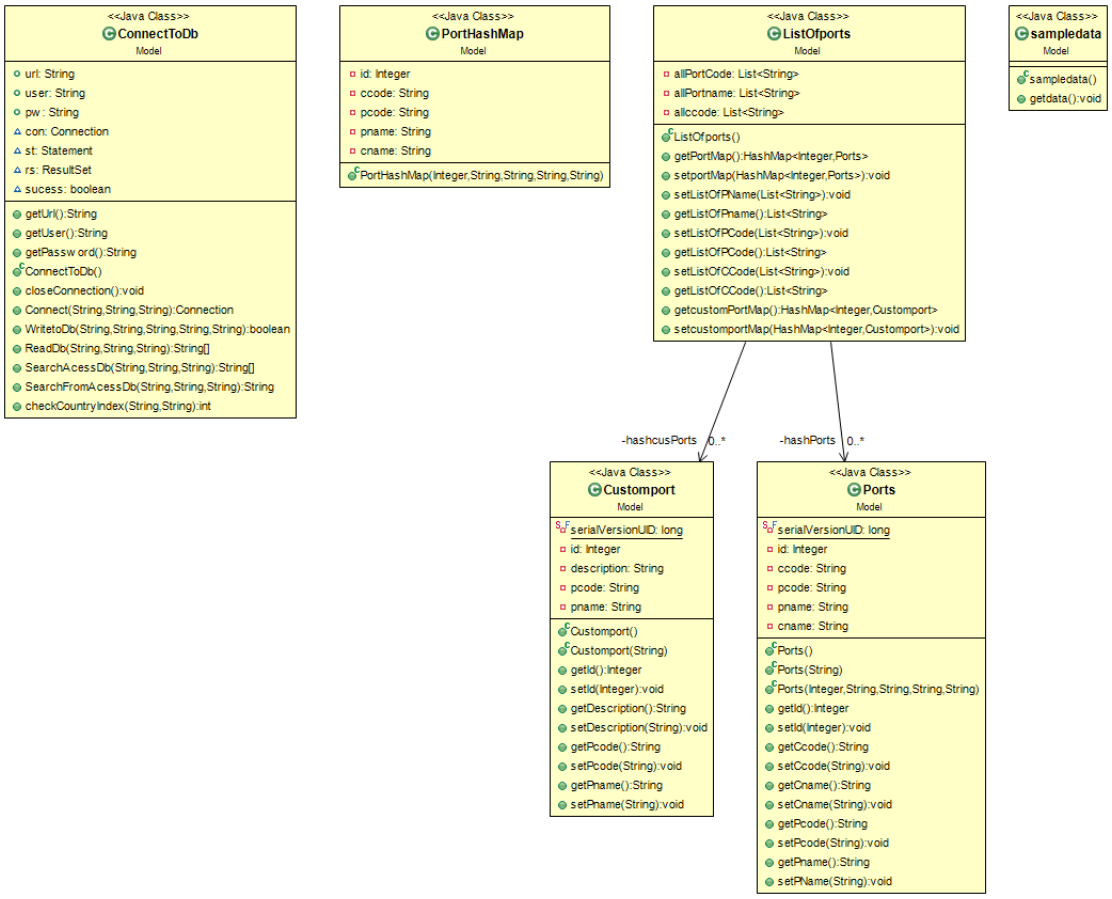


Figure 3.10. model class diagram

View Class diagram The view defines how the application data should be displayed.In our application, the view would define how the AIS Data (port names

and codes) are presented to the user, and receive the data from users in case of customized input which might be mandatory for specific port which will be saved and used for next search.(figure 3.11)

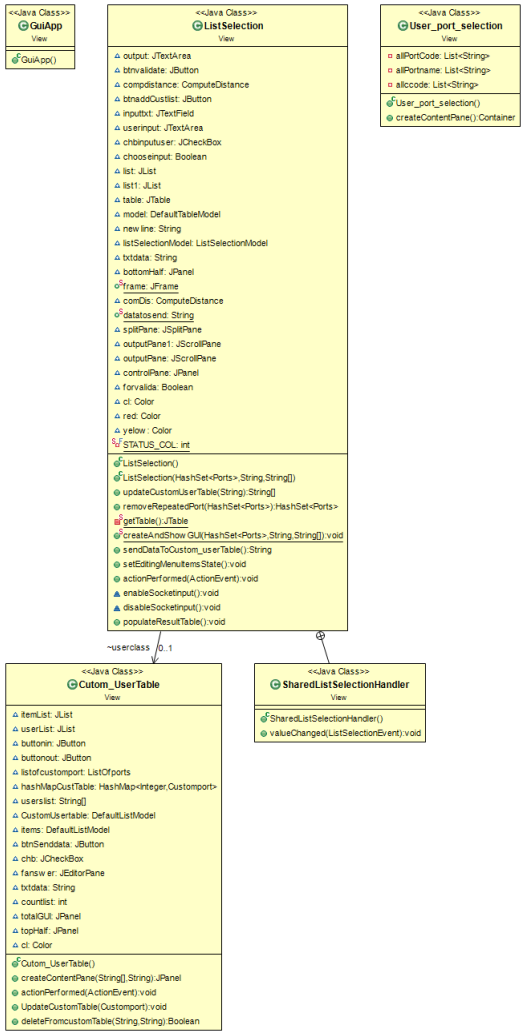


Figure 3.11. View class diagram

User Interface view

The view class has two basic categories for customized data which can be manipulated(edited) by the users and the rest will be the possible list that the algorithms found with an option of sorting the list add remove items ,save to database or send options.(figure 3.12)

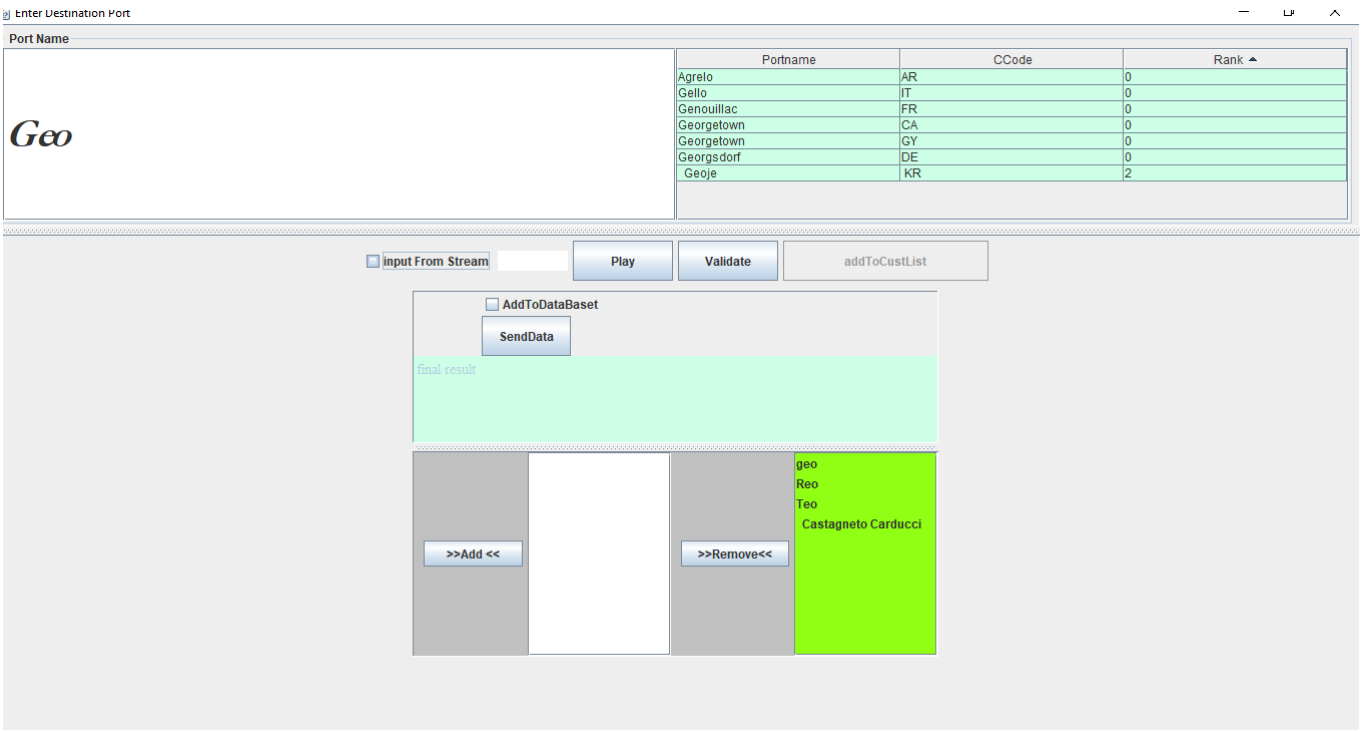


Figure 3.12. User Interface of View class diagram

All the three different class diagram (models) and there relationship which was explained separately shown below (figure 3.13).

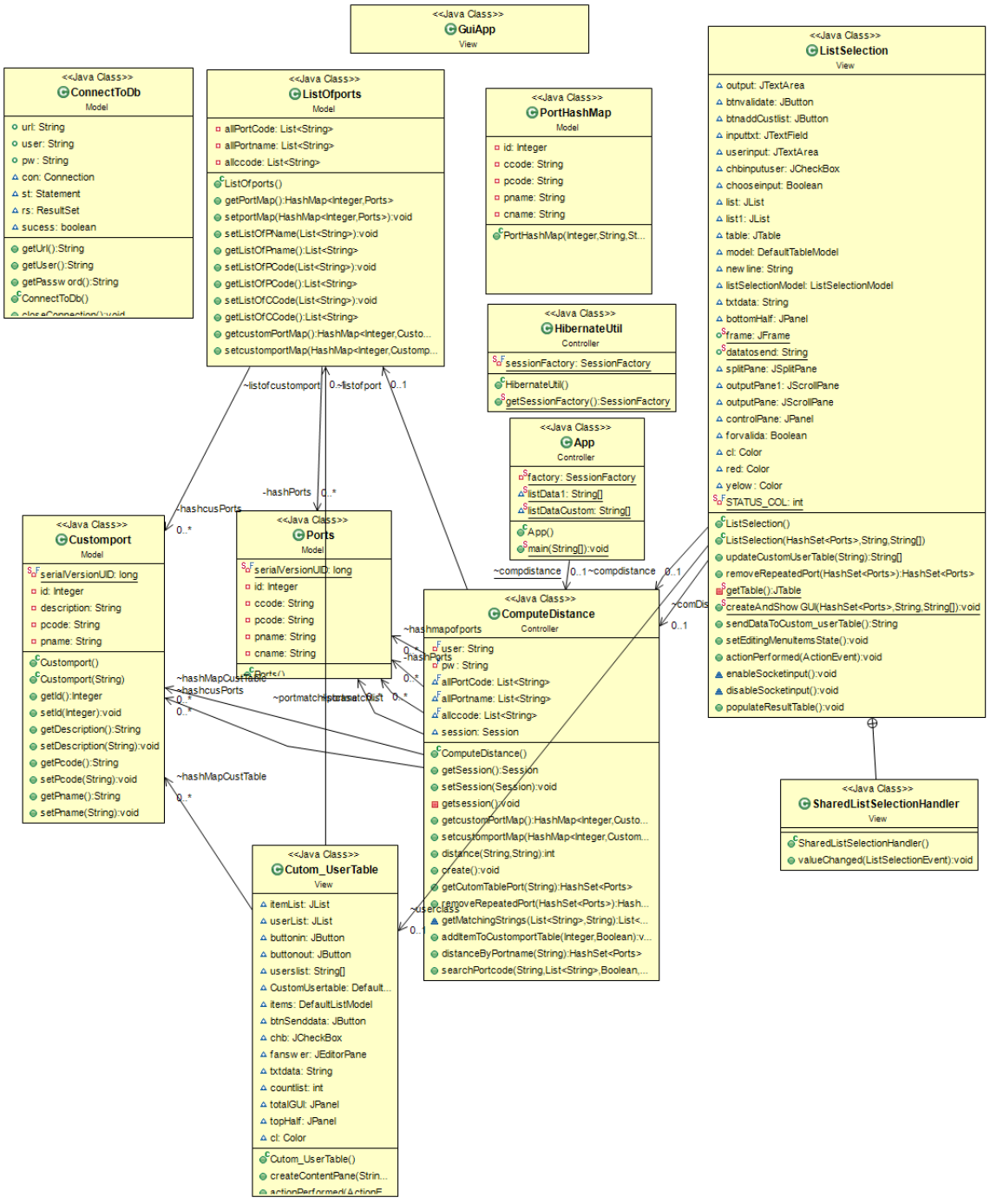


Figure 3.13. All class diagram

3.6 Data Base Architecture

3.6.1 Migration of Data Base

Before testing the application in a real scenario, a migration of the databases containing the port names has been required. The original file was in MS Excel format with the world country list, port names and codes. An example is given below for the first ten lines of the country list.

Country List			
Country Name or Area Name	ISO ALPHA 2 Code	ISO ALPHA 3 Code	ISO numeric Code
Afghanistan	AF	AFG	004
Aland Islands	AX	ALA	248
Albania	AL	ALB	008
Algeria	DZ	DZA	012
American Samoa	AS	ASM	016
Andorra	AD	AND	020
Angola	AO	AGO	024

3.6.2 PostgreSQL Database

PostgreSQL has a lot of capability. Built using an object-relational model, it supports complex structures and a breadth of built-in and user-defined data types and also provides extensive data capacity and is trusted for its data integrity.

Advantages of PostgreSQL It offers many advantages over other database systems such as:

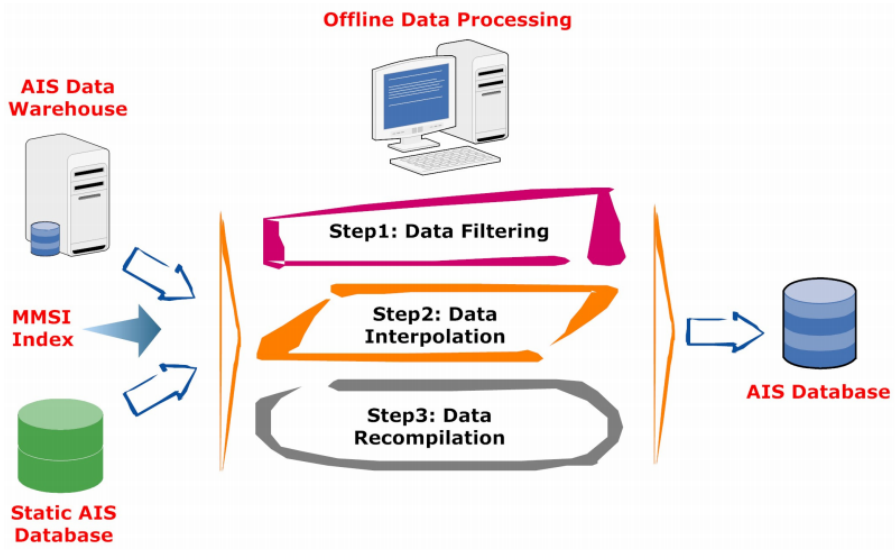


Figure 3.14. AIS Data Processing

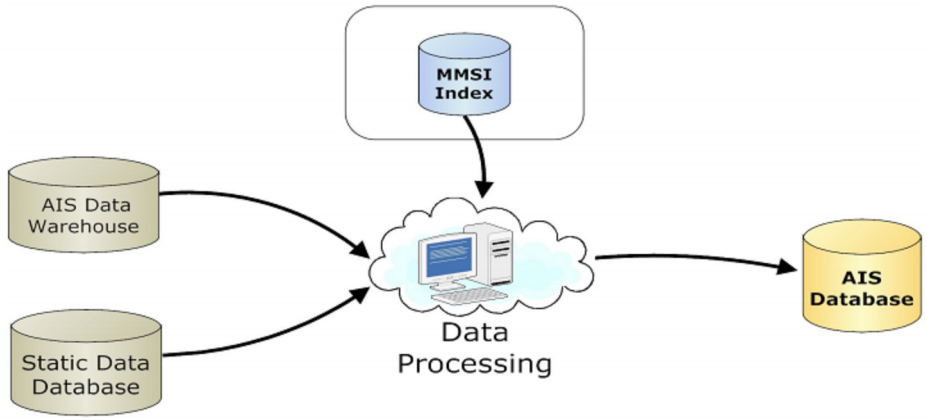


Figure 3.15. AIS DataBase Architecture

Immunity to over-deployment Over-deployment is what some proprietary database vendors regard as their No.1 licence compliance problem. With PostgreSQL, no-one can sue you for breaking licensing agreements, as there is no associated licensing cost for the software.

- More profitable business models with wide-scale deployment.
- No possibility of being audited for license compliance at any stage.
- Flexibility to do concept research and trial deployments without needing to include additional licensing costs

This has several additional advantages:

Better support than the proprietary vendors strong support offerings, have a vibrant community of PostgreSQL professionals and enthusiasts that any professional can draw upon and contribute to

Extensible The source code is available to all at no charge. If youc need to customise or extend PostgreSQL in any way then they are able to do so with a minimum of effort, and with no attached costs.

Cross platform Itis available for almost every brand of Unix (34 platforms with the latest stable release), and Windows compatibility is available via the Cygwin framework. Native Windows compatibility is also available with version 8.0 and above.

Designed for high volume environments use a multiple row data storage strategy called MVCC to make PostgreSQL extremely responsive in high volume environments.

Technical Features To mention a few: Fully ACID compliant,ANSI SQL compliant,Referential Integrity,Replication (non-commercial and commercial solutions) allowing the duplication of the master database to multiple slave machines,Rules,Views,Triggers,Unicode,Sequences,Inheritance,OuterJoins, Subselects,An open API,Native interfaces for ODBC, JDBC, .Net, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby. etc.

Chapter 4

Experimental results

This chapter describes the experiments conducted, specifying the goal of each test, the obtained results, and trying to find a valid motivation to justify the outcomes. Each experiment represents a test to verify the combined effect of selected algorithms made during the implementation. Several tests have been executed in order to evaluate accuracy of the developed implementation for the proposed system and algorithms
The WordMatchEditDistance algorithm described above was evaluated to determine its accuracy as a function of the parameter k (threshold of similarity) and to obtain an adequate value for k. The test procedure was divided in three parts:

- 1. Preparation of the data;
- 2. Execution of the test algorithm;
- 3. Analysis of the results.

4.1 Preparation of the Data

The test used a simplified data set containing only the attributes Port name, port code and country code . A test set of approximately 65455 records was considered, with the purpose of assessing the effects of different user option there is also a customize user data base.

4.2 Evaluation of the Algorithm

The evaluation of the Word match edit distance algorithm encompasses two main issues: the accuracy and the behavior of the similarity threshold, the distances between all possible pairs of records (ri .. rj) are computed, records with lowest value of distance is suggested as first while also the rest of the records are evaluated using the Java pattern library to populate the proposed solutions. Furthermore, the user has an option to manually add, insert, manually choose records for subsequent searching.(an example is given in figure 4.1)

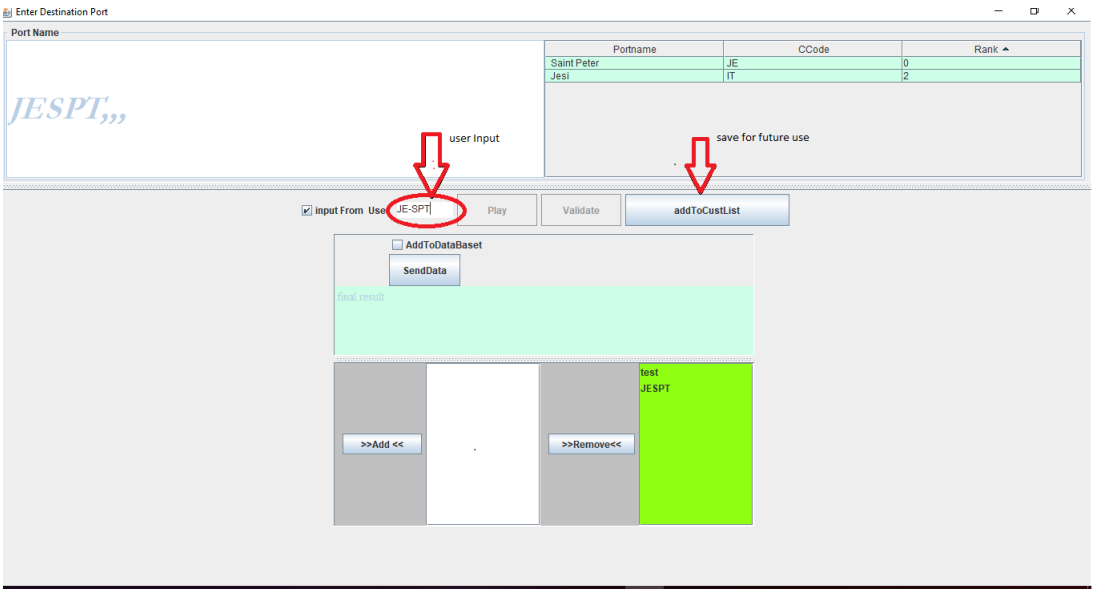


Figure 4.1. manually add records

4.3 Analysis of the Results

As expected there is a trade-off between Precision and Recall since the list of result is expected to be all possible records with max of distance three between two records and possible customized user data list.

4.3.1 Port Name as an Input

A preliminary set of experiments was conducted by receiving as input a port name (longer than 3 characters) but there are ports with three letters and this scenarios can be handle by considering java pattern to find a match beside the distance
The best example is GEX it is port name , a port code with three letter. Geelong(AU), Geisenfeld(DE) and Morgex(IT) has port code GEX and Gex is also a port name in france the final result should include all those list (figure 4.2)

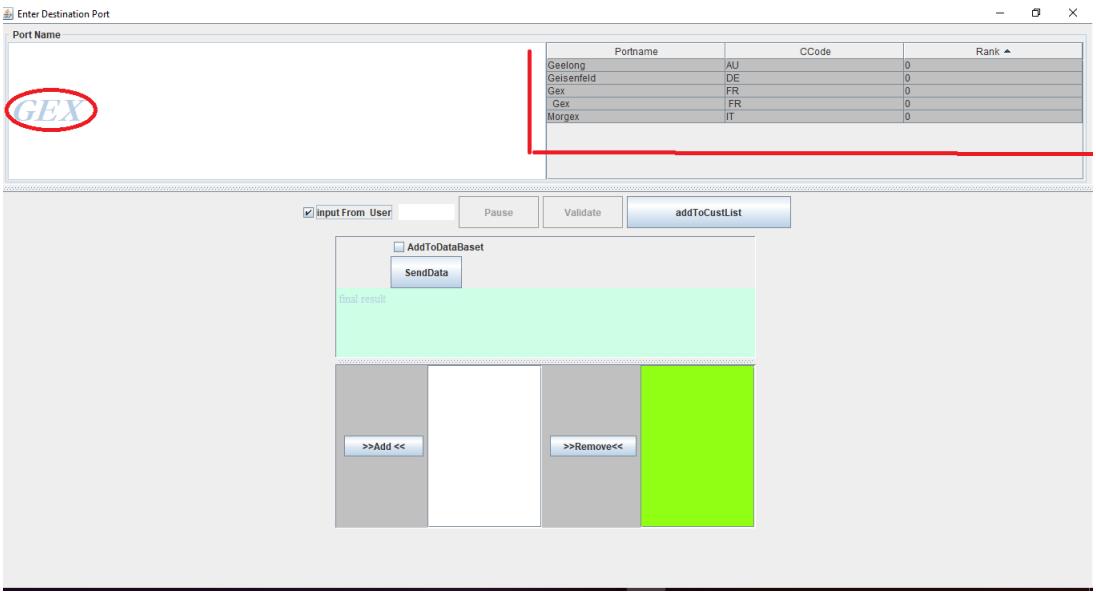


Figure 4.2. Example for GEX as a port name and code

Expected OutPut :Genova

Test 1.A INPUT : Genoa Genoa : suggested as a port name(more than 3 char) with java pattern	OUT PUT 1 Genola(1):IT 2 Genga(1) :IT 3 Genova(1):IT 4 : Genas(2):FR 5 :Genay(2) :FR 6 :Gendt(2) :NL 7 : Geneva(2):ES 8 :Genk(2) :BE 9 : Gensac(2):FR 10 :etc(3 left)

Expected OutPut :Genova

Test 1.B INPUT : GENOvA GENOvA : suggested as a port name(more than 3 char) with pattern	OUT PUT 1 Genova(0):IT 2 Geneva(1) :ES 3 Geneve(2):CH 3 Genola(2) :ES

No Expected Output

Test 1.C INPUT : berge berge : suggested as a port name(more than 3 char) with pattern	OUT PUT 1 Berge(0):ES 2 Berge(0) :DE 3 Berg(1) :CH 4 Berg(1) :DE 5 Berg(1) :NL 6 Berg(1) :NO 7 Bergen(1) :BE 8 Bergen(1) :DE 9 etc.. 10 Berat(2):AL

4.3.2 Port Code as an Input

The second set of experiments was conducted by receiving as input a port code and as we mentioned above port names might have have also equal number of letters so the result set may contain related port name based on distance or pattern of the first few letters(in our case max of four letters) To compare the result we will look the same port code with and with out country code(SPT) and result of San 'Pellegrino Terme'.

Expected OutPut :San Pellegrino Terme	
Test 2.A INPUT : IT..SPT or ITSPT where IT stands for Italy and SPT is the port code with pattern	OUT PUT 1 San Pellegrino Terme(0):IT 2 Bosisio Parini(1):IT port code is BSP 3 Ascoli Piceno(1):ASP
No Expected OutPut	
Test 2.B INPUT : SPT SPT : port code with out country code	OUT PUT 1 Sao Miguel Paulista 2 Saint-Prouant 3 Sint-Pieters-Leeuw 4 Saint-Esprit 5 Speightstown 6 Sopot 7 Spanish Town 8 Saint Peter 9 Saint Peter Port 10 San Pellegrino Terme 11 etc....

4.3.3 Country Code with Port Name as an Input

In which the first two character is country code and the rest is assumed to be Port name or code but in this section the test will consider it as a port name . If the given string(with out the country code) grater than three the first thing to do is that search as port name and next find the the distance between available port code and the given search string but some times port name will be the same as port code as a result our result consider that too.

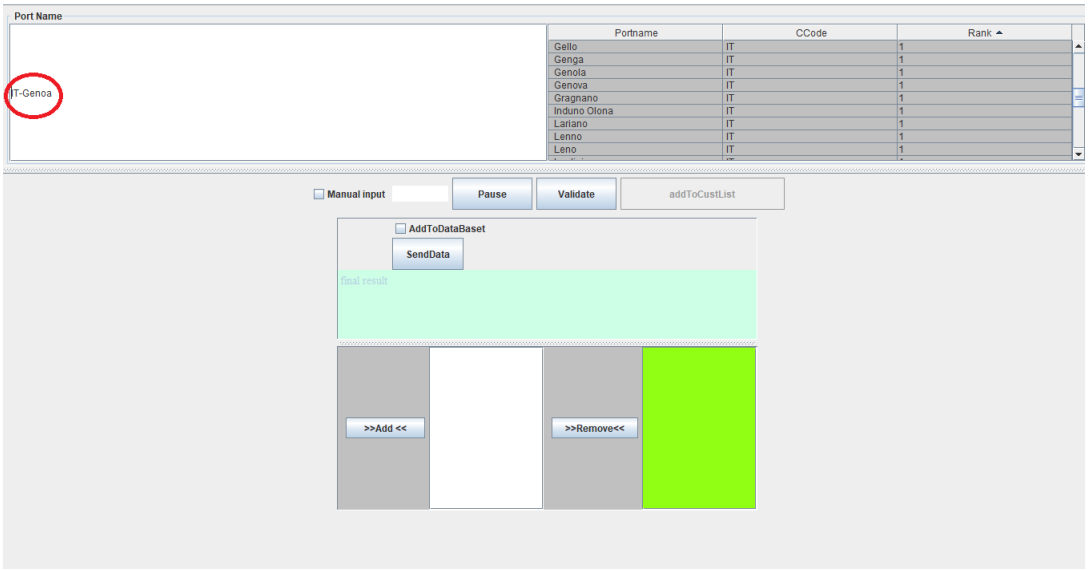


Figure 4.3. Example for country code and port name

Expected OutPut : Genova	
Test 3.A INPUT : ITGENOA or IT-Genoa IT : Italy and	
GENOA : misspelled port name with pattern	OUT PUT 1 Gello(1):IT 2 Genova(1):IT 3 Genga(1):IT 4 Genola(1):IT 5 Gragnano(1):IT with port code GNO 6 etc..
Expected OutPut : Barge or Erbe	
Test 3.B INPUT : ITBERGE or IT–berge IT : Italy and	
BERGE : misspelled port name with pattern	OUT PUT 1 Barge(1):IT 2 Erbe(1):IT 3 Bernareggio(2):IT port code is BEG 4 etc
Expected OutPut :Genova	
Test 3.C INPUT : IT-GENOvA or IT-genova IT : Italy and GENOvA : port name	OUTPUT(4) 1 Genova (2) 2 Nola(3) 3 Nove(3) 4 Genola(3)
Expected OutPut :Acate	

4.3.4 Country code with Port Code as an Input

In this set of experiments, the first two characters represent the country code and the rest of the string represents the port code but since some port names have equal number of character with port code, like GEX is a port code for Geelong ,Geisenfeld and Morgex the length is three but it is also port name 'GEX' in France so the algorithm consider those possibilities too.(see figure 4.4)

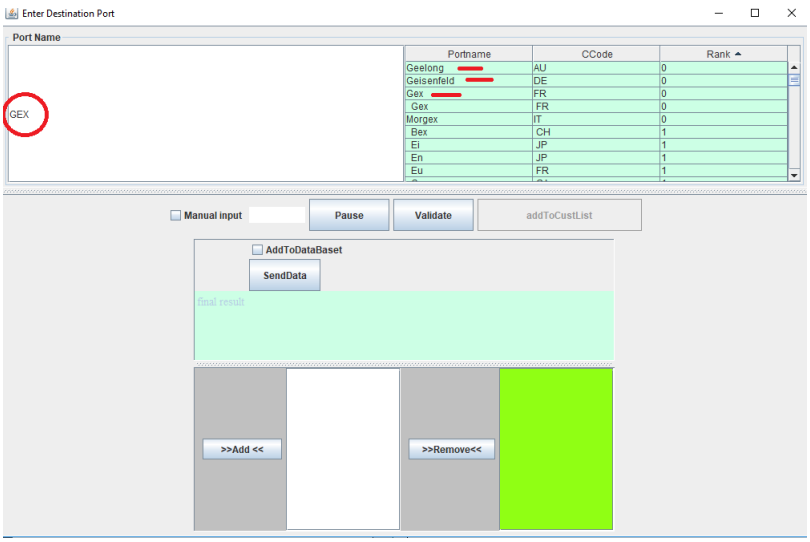


Figure 4.4. Example when with port name and port code are the same

Expected OutPut :Gello	
Test 4.A INPUT : IT..geo or ITGEO IT : Italy and geo : port code for Gello with java pattern	OUT PUT 1 Gello(0):IT 2 Vo(1):IT
Expected output: all ports with GEO in port code	
Test 4.B INPUT : geo SPT : port code with out country code	OUT PUT 1 Agrelo(0):AR 2 Georgetown(0):CA 3 Georgsdorf(0):DE Geoje 4 Genouillac (0):FR 5 Georgetown(0):GY 6 Gello(0):IT 7 etc 8 Ago (1):JP
Expected OutPut :San Pellegrino Terme	
Test 4.C INPUT : IT..SPT or ITSPT IT : Italy and SPT : port code with pattern	OUT PUT (1) 1 San Pellegrino Terme(0):IT
Expected OutPut : Saint Peter	
Test 4.c. INPUT : JE..SPT or JESPT JE : Jersey and SPT : port code	OUT PUT 1 Saint Peter(0):JE
Expected OutPut :Barge	
Test 2.D INPUT : IT..BGE or itbge IT: Italy BGE : port code	OUTPUT(1) 1 Barge(0):IT
Expected OutPut :Gello	
Test 4.E INPUT : IT..GEO or ITgeo IT : Italy and geo : port code	OUTPUT 1 Gello(0):IT 2 vo(1):IT

Chapter 5

Conclusion, Limitations and Future Work

In this work the AIS tool is adopted to improve automatic recognition of port names transmitted by vessels. The proposed approach may increase the safety of passengers, crew, cargo and ships, as well as the marine environment in such a way that the need for a system like AIS for better marine life and the need for standardization of AIS message type 5. This approach leverages on well-known algorithms to evaluate the distance between strings to univocally map an input string to a port name or code. Different algorithms have been tested to prove the effectiveness of the proposed approach for the future of AIS safety, efficiency and for drawing analysis on each ports. Based on the characteristics of type message 5 protocol, source and destination address in the vessel's track decoded from the AIS information and by combing levenshtein distance algorithms with java pattern it is possible to achieve proper and expected data based on different criteria and filters.

The preliminary results show that a matching between the incoming string and a port is possible, given some constraints on the structure of the incoming string. Tests were conducted on different types of strings to see when and how the algorithm can be considered as stable.The algorithm uses dynamic programming methods to create a matching algorithm with java pattern as a general method. The improvement comes at the expense of significant computational time in a backtracking routine. The variations were used to handle typical errors and properties of the database application's domain, like the transposition of words, the existence of unimportant

extra words that may be discarded and the multi attribute nature of the records.

The current main limitation of the system is that a pre filtering of the database is required. This would allow both to eliminate cities not on the seaside and to speed up the system. In particular, the data obtained from the AIS device are accurate and correct as they were the data entered in to the system, the logic for manually inserted port should be better explored in order to put them in the correct ranking. Furthermore, some optional fields like API Service (for further detail look [1])for port calls has an optional message fields for Port-Id with AIS processed data which could provide additional results to reduced computational complexity if they become mandatory.

AIS proves to be beneficial for general safety on the seas and its usefulness for fleet and synthetic aperture radar (SAR) management system but still there are limitation to be learned in the field of dynamic programming problems when we concerns about creating scalable, effective, parallel algorithms through out the AIS network and infrastructures. Future research includes testing to explore efficiency in different data sets and concurrent environment. In addition, analysis of dynamic programming algorithms can lead to more effective solutions, especially for problems that require ad-hoc data analysis. In this version our the system act like a black box with TCP Socket communication and we believe that in the future it can integrate with AIS system as an internal part of the system.

[1] https://www.marinetraffic.com/en/ais-api-services/documentation/api-service:ps03

Bibliography

[1] Tejada, S.; Knoblock, C. A.; and Minton, S. 2001. Learning object identification rules for information integration. Information Systems 26(8):607−633

[2] IEC 62287-1 Ed.1 Maritime Navigation and Radio Communication Equipment and Systems - Class B ship borne equipment of the automatic identification system (AIS) - Part 1:Carrier-Sense Time Division Multiple Access (CSTDMA) Techniques.

[3] Cohen, W. W.; Ravikumar, P.; and Fienberg, S. E. 2003. A comparison of string distance metrics for name-matching tasks. In Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03).

[4] Farid Dowla ,Lawrence Livermore National Lab Livermore, California 94551 Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

[5] Gonzalo Navarr University of Chile, ACM Computing Surveys, Vol. 33, No. 1, March 2001. A Guided Tour to Approximate String Matching

[6] Halifax, Canada, May 3-6, 2015 AIS Data Exchange Protocol Study and Embedded Software Development for Maritime Navigation. Proceeding of the IEEE 28th Canadian Conference on Electrical and Computer Engineering

[7] Eric S. Raymond ¡esr@thyrsus.com¿ v1.32, June 2011 AIVDM/AIVDO protocol decoding = https://web.nlcindia.com/gpsd/gpsd-3.1/www/AIVDM.txt

[8] Monge, A., and Elkan, C. 1997. An efficient domain-independent algorithm for detecting approximately duplicate database records. In The proceedings of the SIGMOD 1997 workshop on data mining and knowledge discovery.

[9] Bilenko, M., and Mooney, R. 2002. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report Technical Report AI 02-296, Artificial Intelligence Lab, University of Texas at Austin. Available from http://www.cs.utexas.edu/users/ml/papers/marlin-tr-02.pdf.

[10] IALA Guidelines on the Universal Automatic Identification System Vol 1 Part I Operational issues, March 2003

[11] R. A. Wagner and M. J. Fischer, The string-to-string correction problem, J. Assoc. Comput. Machinery, vol. 21, no. 1, pp. 168-173, Jan. 1974.

[12] http://nlp.stanford.edu/IR-book/html/htmledition/edit-distance-1.html

[13] P.A.V. Hall and G.R. Dowling, Approximate string matching, ACM Comput. Surveys, vol. 12, pp. 381-402, Dec. 1980.

[14] Y. P. Yang and T. Pavlidis, Optimal correspondence of string subsequences, IEEE Trans. Patt. Anal. Machine Intell., vol. 12, no. 11, pp.1080-1087. Nov. 1990.

[15] The Role of AIS for Small Ships Montoring ,Marek Dziewicki, Maritime Office Gdynia . Department of ATON Technique and Radionavigation Systems. BalticMaster Workshop , Gdynia 11-12 May 2006

[16] http://catb.org/gpsd/AIVDM.html

[17] http://www.marinetraffic.com/en/ais-api-services/documentation/api-service:25/

[18] http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-(SOLAS),-1974.aspx.

[19] http://opencpn.org/ocpn/developers manual

[20] he International Journal on Marine Navigation and Safety of Sea Transportation http://www.transnav.eu/

[21] IALA Guidelines on the Universal Automatic Identification System Vol 1 Part I Operational issues, March 2003

[22] Decode AIS messages (2017,January 25) https://github.com/tbsalling/aismessages/wiki

[23] Objectaid-uml-explorer The ObjectAid UML Explorer for Eclipse:(2017,may 24) https://marketplace.eclipse.org/content

[24] The ObjectAid UML Explorer (2017,may 25)http://www.objectaid.com

[25] PostgreSQL object-relational database management system. (2017, may 15) https://www.postgresql.org