J. Marine Sci. Appl. (2012) 11: 504-511
DOI: 10.1007/s11804-012-1161-y

Development of Re-usable C++ classes for Location Aware Management and Control Systems for Shipping Applications

Dhiren Dave*, Sanjay Nalbalwar and Ashok Ghatol

1. Department of Electronics & Telecommunication Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere 402103, India

Abstract: Location aware management and location based automation are fast upcoming technologies which are facilitated by availability of reliable, accurate, and cheap global positioning systems and location based services. This paper discusses the development of C++ classes, which may be used for development of software for location aware management and control, specifically in merchant shipping, so as to provide improved safety, increased automation, prevention of pollution and reducing the work burden of the crew. The GPS is used in conjunction with the regionally accessible nested global shorelines (RANGS) dataset for this purpose. The design of two main C++ classes named CShorelines and CGPSInput have been discussed. A control system mandatory on all oil-tankers above 150 GRT (and few other vessels) called ODMCS has been made location aware using the novel software developed and the results obtained have been critically examined and presented.
Keywords: GPS; Location Aware Control; ODMCS; RANGS
Article ID: 1671-9433(2012)04-0504-08

Symbols and Abbreviations

DOP – Dilution of Precision (of GPS data)
GPS – Global Positioning System
GRT – Gross Tonnage
GSHHS – Global Self-consistent Hierarchical High-resolution Shorelines
LBS – Location Based Services
MARPOL – Marine Pollution (International Convention)
MFC – Microsoft Foundation Classes
NIMA – National Imagery and Mapping Agency (USA)
NMEA – National Marine Electronics Association (USA)
ODMCS – Oil Discharge Monitoring and Control System
RANGS – Regionally Accessible Nested Global Shorelines
UTC – Coordinated Universal Time
WDB – World Data Bank
WVS – World Vector Shorelines (dataset)

1 Introduction

In last decade, the availability of low cost, reliable and accurate GPS (Global Positioning System，2011) and concurrent growth of mobile telephone based LBS has paved the way for serge in location aware management and control systems. Such systems are finding acceptance in mobile applications like cellular telephony, rail/road based transportation, aerospace, navy, merchant shipping etc (Dhiren and Ashok, 2009; Nord *et al*., 2002). This paper discusses the development of reusable C++ class templates, developed around MFC, which may be used for development of software for location aware management and automatic control, specifically in merchant shipping applications. It can be shown that such systems provide improved safety and reliability, increased automation, prevention of pollution and reducing the work burden of the crew (Dhiren and Ashok, 2007; Dhiren, 2010a; Dhiren, 2010b; Dhiren and Ashok, 2009).

A location aware system should be able to receive location inputs, maintain location database, and take location based actions after processing the inputs and location database. The information about the current location may be received using various technologies like GPS, LBS, Blue-tooth etc. The location database may include important land-marks, geographical information, map etc. For marine applications, this may include shorelines, locations of ports, important routes, special areas, environmentally sensitive areas etc. This information may be processed and used for various purposes like navigation, route planning, collision avoidance, ship-board related waste management, environment protection, anti-piracy measures etc.

Section 2 briefly describes NMEA protocol, a popular data communication standard used for communication of data between GPS and other ship-board devices and the state-machine implementation of the same. Section 3 gives a brief description of RANGS database. Section 4 presents software implementation issues like features requirements, software architecture, simple flowcharts and templates for the two important classes namely CShorelines and CGPSInput, which act as the main data retrieval and processing mechanism. Section 5 describes a few of the possible applications of such a system. Section 6 presents the test results of a location aware oil discharge monitoring and

Received date: 2011-12-09.
***Corresponding author Email:** dhirend@tmi.tolani.edu

control system developed based on these classes.

2 NMEA 0183 Standard

The GPS is a worldwide radio-navigation system formed from a constellation of 24 satellites and their ground stations. Most GPS devices onboard the ships support NMEA protocol for data communication between GPS device and the computer. The present work is based on the NMEA 0183 specification (National Marine Electronics Association, 2002). The NMEA 0813 standard for interfacing marine electronics devices specifies the NMEA data sentence structure as well as general definitions of approved sentences. However, the specification does not cover implementation and design. The author has presented the software design tasks needed to parse through NMEA sentences robustly. The technique for parsing and data integrity checking is proposed.

NMEA data is sent in 8-bit ASCII where the MSB is set to zero. The specification also has a set of reserved characters. These characters assist in the formatting of the NMEA data string. The specification also states valid characters and gives a table of these characters ranging from 0x20[2] to 0x7E. As stated in the NMEA 0183 specification version 3.01, the maximum number of characters shall be 82, consisting of a maximum of 79 characters between start of message "$" or "!" and terminating delimiter <CR><LF> (0x0D and 0x0A). The minimum number of fields is one. The basic format for NMEA sentence is as below:

$aaccc,c--c*hh<CR><LF>

Where the various characters/fields represent the following,
$ - Start of sentence
aaccc - Address field/Command
"," - Field delimiter (0x2C)
c--c - Data sentence block
* - Checksum delimiter (0x2A)
hh - Checksum field (the hexadecimal value represented in ASCII)
<CR><LF> - End of sentence (0x0D 0x0A)

3 RANGS Dataset

To make a software to be used in marine applications location aware, at the very minimum, it requires the information on present location (for e.g. from GPS) as well as information on shorelines in vector format and on a global basis. The WVS provides data on shorelines in vector format on a global basis. The WVS was originally provided by the National Imagery and Mapping Agency, USA. In fact, an improved version of WVS called RANGS (Feistel, 1999) has been used for the purpose of present work. RANGS files are based on the GSHHS files. The GSHHS dataset had been derived by Wessel and Smith from the WVS dataset combined with additional WDB dataset (Wessel and Smith, 1996). Wessel and Smith also developed various low resolution versions from the main WVS dataset using the algorithms developed by Douglas and Peucker (1973). The RANGS dataset is organized into cells of 1° longitude x 1° latitude covering entire globe. This forms 64 800 cells. The cell contains an array of points (lon, lat) representing the shoreline segments, forming a simple, closed cell polygon. It also includes the inland lakes and ponds. The dataset also contains information on whether the inside of a particular polygon is land or water.

RANGS files are available in five resolution versions (full resolution = 0.1, high = 0.2, intermediate = 1.0, low = 5.0 and crude = 25 km). The corresponding files are named as rangs(0), rangs(1), rangs(2), rangs(3) and rangs(4) respectively. The dataset for each resolution level n ($0 \le n \le 4$), is organised in group of three files:

(i) Cell Address Table file, Rangs(n).cat
(ii) Cell Extraction List file, Rangs(n).cel
(iii) Shoreline data file, gshhs(n).rim

The details of the structure of these files and how they are organised to access RANGS information may be found in Feistel (1999). The flowcharts for reading and interpretation of RANGS data is given in section 5.

4 Software Design

Typical location aware software for application in merchant shipping is expected to cater for any one or more of the location based features like alarm, monitoring and control, management etc. Accordingly, the specific features to be taken into consideration while developing such software are as follows.

4.1 Features Requirements

(i) Ability to receive location information from various location providers (GPS, Blue tooth devices, Mobile Networks etc.), in real-time, using commonly used communication protocols like RS232C, USB etc.
(ii) Ability to interpret/process location data format for extraction of information like position, speed, heading etc.
(iii) Ability to import shoreline data in vector format.
(iv) Ability to integrate into other applications easily.
(v) To calculate distance between any two points on surface of earth along the great circle.

To calculate the distance between any two points, represented by say, $P_1(\theta_1,\delta_1)$ and P_2 (θ_2,δ_2), where θ_1 and θ_2 represent longitudes and δ_1 and δ_2 represent latitudes, various methods for e.g. Haversine's Formula and Spherical Law of Cosines (Sinnott, 1984; Gellert *et al*., 1989) etc. are available. This paper proposes using spherical law of cosines because of its

[2] A prefix 0x indicates that the number following it is hexadecimal.

simplicity and accuracy, which is good enough for many applications in shipping. The spherical low of cosines is as follows.

$$D = a\cos\left(\sin(\delta_1)\cdot\sin(\delta_2)+\cos(\delta_1)\cdot\cos(\delta_2)\cdot\cos(\theta_2-\theta_1)\right)R$$

where, D is the distance (in Knots) and R is the mean radius of earth (km). This formula gives the results to the accuracy of about one meter (available online: http://www.movable-type.co.uk/scripts/latlong.html, accessed on 17th Sept, 2011).

4.2 Software Architecture

The software architecture for the developed system is shown in Fig. 1.

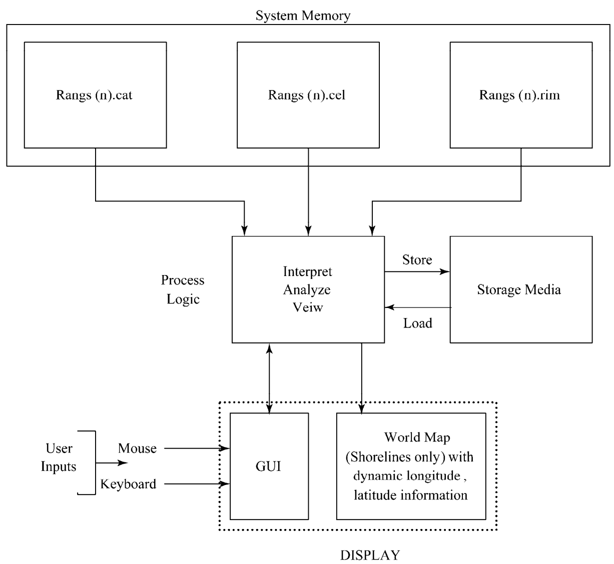

Fig. 1 Software Architecture of the proposed location aware system

It consists of the following modules:

(i) *Data Files*: Rangs(n).cat, Rangs(n).cel and Rangs(n).rim.

(ii) *Process Logic*: This module consists of routines for performing various tasks like
- Receive user inputs through GUI
- Access database (load/store)
- Interpretation of database format, data analysis, rendering and viewing
- Calculation of distance between any two points on map along the great circle
- Zoom-in, zoom-out etc.
- Receive and process location data

(iii) *Storage Media*: The software related to storage media performs only functions of serializing (Kruglinski *et al.*, 1998) the data to and from the hard drive or any other media accessible through standard Microsoft™ Windows© interface.

(iv) *Display Module*: This module is mainly responsible for receiving user input commands form keyboard and mouse via GUI and displaying the world shorelines on the graphic display along with dynamic display of longitude and latitude at mouse pointer in status-bar of the window. This module also displays current position of ship, based on data received from GPS.

5 Implementation

Two of the most important classes which encapsulate most of the data and functionality of a location aware system, for the proposed shipping applications, are implemented in native C++ using Microsoft® Visual C++ 6.0 integrated development platform.

CShorelines class

The CShorelines is designed to encapsulate data and functions to meet the following main application requirements.

(i) Ability to import (read, interpret and display) shoreline data in vector format.
(ii) Save/Load the display image to/from storage media (hard-drive).
(iii) Dynamically display longitude and latitude at any user specified point anywhere on map.
(iv) To measure distance between any two points on the map in nautical miles.
(v) Zoom in/Zoom out whole map.
(vi) Zoom-in a selected portion of the map for closer inspection.
(vii) Determination of point background (ground or ocean) programmatically.
(viii) Determine if a given position specified by (longitude, latitude) lies within a *special area* (Dhiren, 2010a; Dhiren, 2010b; Dhiren and Ashok, 2009) or not.
(ix) Calculate the distance of a point from the nearest shoreline.
(x) Configurable with different display resolutions and pallets.

CGPSInput class

The CGPSInput class is designed to encapsulate data and function to meet the following application requirements.

(i) Ability to receive location data (from GPS) using COM/USB port or blue tooth device into the application.
(ii) Ability to read, interpret and process NIMA sentences received from GPS device.
(iii) Ability to handle timeouts and other data transmission errors (related to GPS).
(iv) Display the important information received from GPS device on the monitor of computer.

The header and implementation files for CShorelines and CGPSInput classes are not presented here to conserve the space. However, the same may be obtained from the author on request.

5.1 NMEA interpreter state machine implementation

Fig. 2 shows the state machine implementation of a typical NMEA sentence interpreter. The state machine tracks the protocol state and any errors that may occur during the data transfer. This approach allows to track the state of the system (within the protocol) and also to recover from errors like timeouts, checksum errors etc. The various states of the state machine are described below:

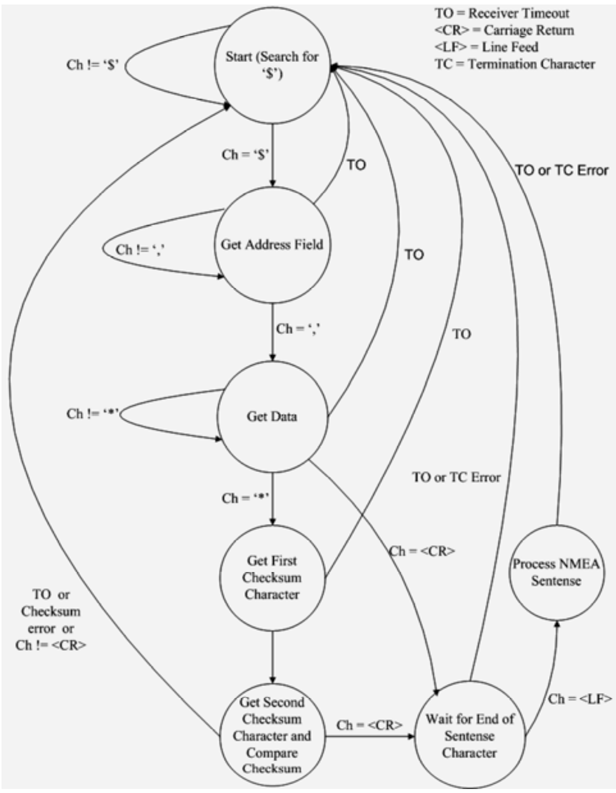

Fig. 2 The state machine implementation for receiving GPS data.

The timeout transition in the majority of the states is necessary so as to synchronize the state machine when no data is received for a period of time.

(i) State SOS (Start of Sentence): In this state the state machine looks for the '$' (0x24) that is start of the sentence character.
(ii) Receive address/command: In this state, the state machine collects characters until it receives a ',' (0x2C) character. The variable length address field allows parsing of any undefined or proprietary sentences.
(iii) Get sentence data: In this state, the state machine continues to collect data and also performs a checksum until it receives a checksum delimiter "*" (0x2A) or sentence terminator <CR><LF> (0x0D 0x0A).
(iv) Get checksum character (First): In this state, the state machine simply waits for the arrival of first checksum character.
(v) Get checksum character (Second): In this state, the machine waits for the arrival of second checksum character. After receiving the second checksum character, the received checksum is verified against the calculated checksum.
(vi) Get sentence terminator (ST), first character: In this state, the state machine simply waits for the first sentence terminator character (0x0D).
(vii) Get sentence terminator (ST), second character: In this state the state machine waits for the arrival of 0x0A character. When this character is received, sentence is complete and may be processed by NMEA interpreter.

The state machine makes a transition to initial state, when no data has been received for a period of time (timeout). The timeout duration is application dependent. A 4800 baud device typically sends data every 1 to 2 seconds and timeout of 3 to 4 seconds is sufficient.

Fig. 3 shows the data flow diagram for the NMEA processor. The NMEA processor receives NMEA sentences from the state machine and processes this sentence for extracting information. The state machine receives the data through the RS-232C serial port of the computer.

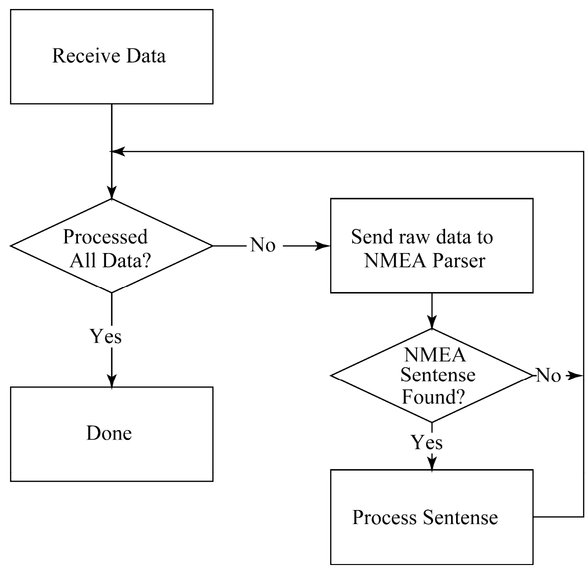

Fig. 3 Data flow diagram for NMEA sentence interpreter

The author has used the following commands in the application developed [8].
GPGGA - Global Positioning System Fix Data
GPGSA - GPS DOP and Active Satellites
GPGSV - GPS Satellites in View
GPRMB - Recommended Minimum Navigation Information
GPRMC - Recommended Minimum Specific GPS/TRANSIT Data
GPZDA - UTC Date / Time and Local Time Zone Offset

5.2 Flow chart for reading and interpretation of RANGS data

The software, for reading, interpretation, and rendering of RANGS data on screen, is developed using native C++ code. Figures 4(a) to 4(c) depict the flowchart for reading and interpretation of the RANGS data.

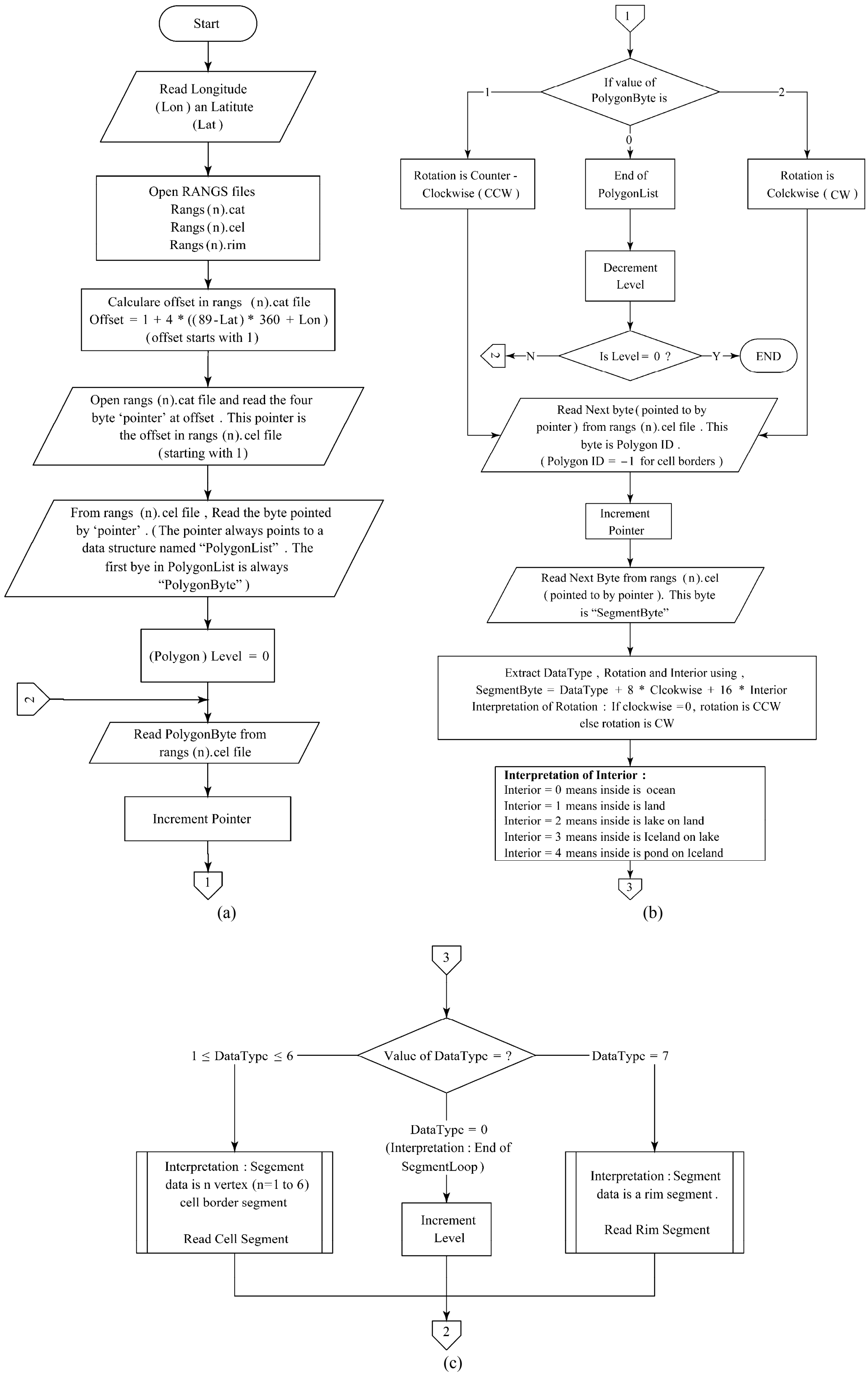

Fig. 4 Flowchart for reading RANGS data

5.3 Algorithm to detect if the ship is in special area

Following procedure is used to detect the presence of ship in any of the special areas as defined in MARPOL regulations (International Maritime Organisation, 2002).

1. Get ship position (Longitude, Latitude).
2. Start with any one of the special areas.
3. Get array of points forming a enclosing polygon (for selected special area). Each point in array represents a pair of longitude and latitude.
4. Is ship within selected special area (polygon)? If yes, go to step 7.
5. All special areas checked? If yes, go to step 8.
6. Select next special area. Go to step 3.
7. Set the ship is in restricted area alarm. Return from Procedure.
8. Clear the flag representing alarm that the ship is in restricted area. Return.

5.4 Algorithm for finding the nearest land

Please see (International Maritime Organisation, 2002) for the definition of 'nearest land'. The algorithm is as follows-

1. Current_Ship_Position = (Longitude_From_GPS, Latitude_From_GPS).
2. Cell_Longitude = Longitude of the lower left corner of the current cell.
3. Cell_Latitude = Latitude of lower left corner of current cell.
4. Longitude = (Cell_Longitude - 1). If Longitude is -181, Longitude = 179.
5. If Latitude < 89 OR Latitude > -90 Than Cell_Latitude = Latitude-1.
6. Get the cell, whose lower left corner has longitude and latitude one less than that of the current cell.
7. Get the array of points for the cell obtained in step (c) and check if any of the points in the array has a distance of 50 Nautical miles or less, from the current position of ship, using equation (4). If yes, set the flag for this alarm and terminate.
8. Otherwise, repeat step (c) and (d) for all the cells up to one longitude and one latitude greater than the current cell (total 9 cells including current cell).
9. If all the cells are checked, clear the flag for corresponding alarm and terminate.

6 Results and Conclusions

The testing of CGPSInput class was carried out using GARMIN® GPS-72™ device, giving accuracy of about 15 meters, 95 % times (GARMIN GPS-72 Specifications, 2011), which was interfaced with an IBM-PC compatible computer with Intel® Core2 Duo™ processor running Windows® XP™, using RS232C interface cable. The CShorelines class was developed and tested around Rangs(0).cat, Rangs(0).cel, and Rangs(0).rim files, giving a resolution of about 100 meters (Feistel, 1999).

A control system mandatory on all oil-tankers above 150 GRT (and few other vessels) called ODMCS (Dhiren, 2010a) has been made location aware using the novel software developed. The developed system automatically detects the presence of ship in special areas (Dhiren and Ashok, 2007; Dhiren, 2010a; International Maritime Organisation, 2002), generates alarm and takes appropriate control actions. Also, it generates alarms and initiates suitable control actions when the ship is less than 50 nautical miles from the nearest shore, in accordance with the MARPOL regulations (International Maritime Organisation, 2002). It is not possible to publish the detailed results of the developed system (ODMCS) because of limit of space. Hence, a summary of results is presented in the appendix-I. However, the complete data generated may be obtained from the author on request.

References

Calculate distance, bearing and more between Latitude/Longitude points. [2011-09-17]. http://www.movable-type.co.uk/scripts/latlong.html.

Dhiren D, Ashok G (2009). Patent Application: Location aware pollution control for ships. Application Number: 2019/MUM/2009, Dated 03/09/2009.

Dhiren D, Ashok G (2007). A Software paradigm for complete automation of ODMCS. *Int. Nat. Conf. on All Electric Ship*, London.

Dhiren D (2010a). Design of Completely Automatic ODMCS. *International Journal of Maritime Engineering*, **152**(A3), 147-157.

Dhiren D (2010b). Location based intelligent pollution control for ships. *Int. Nat. Maritime Tech. Conf.*, Sept 20, Mumbai, India.

Dhiren D, Ashok G (2009). Software architecture for modeling, simulation and automation of ballast water discharge from oil tankers. *International Journal of Engineering Simulation*, **10** (2), 27-35.

Douglas D, Peucker T (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, **10**(2), 112-122.

Feistel, R (1999). New shoreline map-drawing data available. *EOS Transactions. American Geophysical Union, Electronic Supplement*, **80**(22), 249, doi:10.1029/99EO00188.

GARMIN GPS-72 Specifications. [2011-11-17]. http://www8.garmin.com/products/gps72/spec.html.

Gellert W, Gottwald S, Hellwich M, Kästner H, and Küstner H (1989). *The VNR Concise Encyclopaedia of Mathematics*, 2nd ed., ch. 12, Van Nostrand Reinhold, New York.

Global Positioning System. [2011-11-17].http://en.wikipedia.org/wiki/.

International Maritime Organisation (2002). *MARPOL 73/78*, Mumbai: Bhandarkar Publications, 58-65.

Kruglinski D, Wingo S, Shepherd G (1998). Programming Microsoft Visual C++. Microsoft Press.

National Marine Electronics Association, *NMEA 0183 Standard For Interfacing Marine Electronic Devices*," Version 3.01, January 1, 2002.

Nord J, Synnes K, Parnes P (2002). An architecture for location aware applications. *Proc. 35th Hawai Int. Nat. Conf. on System Sciences*. [2011-11-17]. http://csdl2.computer.org/comp/proceedings/hicss/2002/1435/09/14350293.pdf.

Sinnott R (1984). Virtues of the Haversine. *Sky and Telescope*, **68**(2), 195.

Wessel P, Smith WHF (1996). A Global self-consistent, hierarchical, high-resolution shoreline database. *Journal of Geophysical Research*, **101**(B4), 8741-8743.

Dhiren P. Dave obtained B.E. degree in Electrical Engineering from Govt. College of Engineering, Amravati, M.S., India in 1987.and M.E. degree in Electronics Engineering from S.G.G.S. College of Engineering and Technology, Nanded, M.S., India in 1992. He has worked as a lecturer, assistant professor and associate professor at various engineering colleges for about 25 years in the area of electronics engineering. He has also worked as software development manager for two years at CMR Design Automation Pvt. Ltd, New Delhi, India. Presently, he is working as senior associate professor at Tolani Maritime Institute, Talegaon Chakan Road, Induri, Taluka – Maval, Dist – Pune, India, PIN – 410 507. His areas of interest are Modeling, Simulation and Animation of systems. He is a life member (LM-19742) of Indian society of technical education, india and a fellow (F-1149349) of the Institution of Engineers, India.

Dr. Sanjay Nalbalwar has received B.E. (Computer Science & Engineering) in 1990 and M.E. (Electronics) in 1995 from SGGS College of Engineering and Technology, Nanded. He has completed Ph.D. from Indian Institute of Technology, Delhi, in 2008. He has around 22 years of teaching experience and is working as a Professor & Head of Electronics & Telecommunication Engineering Department at Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, Maharashtra (India). His area of interest includes Multirate signal processing and Wavelet, stochastic process modeling. He has around 150 papers to his credit in national and international conferences and 45 papers in the international journals. He has guided about 50 M. Tech projects and about 200 B. Tech projects. Also, presently guiding 5 PhD s tudents.

Dr. Ashok A. Ghatol obtained B.E. degree in Electrical Engineering from Nagpur University, Nagpur, M.S., India in 1971 and M. Tech. and Ph. D. degrees in Electrical Engineering from Indian Institute of Technology, Mumbai, India, in 1973 and 1984 respectively. Over the last more than 35 years, he has been actively involved in the field of Technical Education as Academician, Researcher, Teacher, Planner and Administrator and handled various aspects of Technical Education including Management, Engineering, Technology etc. He served as vice-chancellor of Dr. Babasaheb Ambedkar Technological University, Lonere-Raigad, India. Before joining as vice-chancellor, he was principal/director at College of Engineering, Pune, India, during 2001-2005 and principal at Government College of Engineering, Amaravati, India, during 1994-2001. He is also the chairman of Western Regional Council of All India Council of Technical Education. He is also the executive member of Indian Society of Technical Education. He is Fellow of Institution of Electronics and Telecommunications Engineers, India, Fellow of Institution of Engineers, India and Chairman, Instrument Society of India. He has lectured extensively in various National and International Conferences and has earned unique honor and distinctions. Best Teacher Award, of Govt. of Maharashtra, India, was bestowed upon him during the academic year 1998 – 1999. He has also received Quality Environment Award in the year 2002.

Appendix I

Summary of test results

Test Case No	**Start and End Coordinates of the ships route (Longitude/Latitude)**	**Alarms generated (if any) in order of their occurrence**	**Action taken by ODMCS**	**Test Passed/ Failed**
1	Start: -16°39' 1"/ 19°13' 1" Dest: -44°45'32"/ -1° 7'36"	5, Slop Tank Full (STF)	For alarm 5 -ODV Closed, STV Opened For STF – ODV Closed, STV closed	PASSED
2	Start: 80°34'39"/ 5°46'35" Dest: 93°38'59"/ 5°53' 6"	5 (at start location)	ODV Closed, STV Opened	PASSED
3	Start: -8°57'25"/ 36° 4'22" Dest: -35°47'38"/ 42°20'36"	Nil	ODV Opened, STV Closed	PASSED
4	Start: 67°24'51"/ 20°53' 8" Dest: 72°39' 2"/ 19°11'52"	5 (near destination)	ODV Closed, STV Opened	PASSED
5	Start: 77°32'31"/ 5°14' 6" Dest: 94°24'54"/ 5°56'19"	5, 1, STF, 5	For alarms 5,1 before STF occurred – ODV Closed, STV Opened On STF alarm – ODV Closed, STV Closed. Outputs remained unchanged for alarm 5 after STF alarm	PASSED
6	Start: 76°39' 0"/ 6°25'57" Dest: 109°49'51"/-10°25'56"	2, STF, 5, 5	Alarm 2 – ODV Closed, STV Opened STF – ODV Closed, STV Closed For both the Alarm 5 after STF, outputs remained unchanged at ODV Closed, STV Closed	PASSED
7	Start: 81°33'34"/ 5°27' 8" Dest: 85°44'49"/ 5°27' 8"	5, 3, STF	For alarms 5,3 – ODV Closed, STV Opened On STF alarm – ODV Closed, STV Closed.	PASSED
8	Start: -9°52' 4"/ 43° 8'39" Dest: -6°35'46"/ 52° 3'17"	5, 4, (4)(5), (4)(5), STF, 4, (4)(5)	For all alarms before STF alarm – ODV Closed, STV Opened On STF alarm – ODV Closed, STV Closed. For all alarms after STF alarm, the output remained unchanged	PASSED
9	Start: -11°59'34"/ 56°45'29" Dest: 0°45'31"/ 63° 9'41"	4, (4)(5), (4)(5) STF, 4	For all alarms before STF alarm – ODV Closed, STV Opened On STF alarm – ODV Closed, STV Closed. For alarm 4 after STF alarm, the output remained unchanged	PASSED
10	Start: 56°29'17"/ 26° 6'31" Dest: 64°18'24"/ 19°27' 9"	(4)(5), 4, (4)(5), 5	For all alarms - ODV Closed, STV Opened	PASSED

Alarms:

1. Instantaneous Discharge Rate > 30 L/Nm
2. Total Overboard Discharge > (1/30000) of Total Cargo on Last Voyage
3. Ship Not En-route
4. Ship in Restricted Area
5. Ship < 50 Nm from the *Nearest Land*