The South African Institute for Computer Scientists and Information Technologists

ANNUAL RESEARCH AND DEVELOPMENT SYMPOSIUM

23-24 NOVEMBER 1998
CAPE TOWN
Van Riebeeck hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,
Potchefstroom University for CHE and
The University of Natal

PROCEEDINGS

EDITED BY
D. PETKOV AND L. VENTER

SPONSORED BY

ABSA Group

The South African Institute for Computer Scientists and Information Technologists

ANNUAL RESEARCH AND DEVELOPMENT SYMPOSIUM

23-24 NOVEMBER 1998
CAPE TOWN
Van Riebeeck hotel in Gordons Bay

Hosted by the University of Cape Town in association with the CSSA,
Potchefstroom University for CHE and
The University of Natal

GENERAL CHAIR : PROF G. HATTINGH, PU CHE

PROGRAMME CO-CHAIRS:
PROF. L VENTER, PU CHE (Vaal Triangle), PROF. D. PETKOV, UN-PMB

LOCAL ORGANISING CHAIR: PROF. P. LICKER, UCT - IS

PROCEEDINGS

EDITED BY
D. PETKOV AND L. VENTER

SYMPOSIUM THEME:

Development of a quality academic CS/IS infrastrauction in South Africa

SPONSORED BY

ABSA Group

Proceedings of the 1998 Annual Research Conference of the South African Institute for Computer Scientists and Information Technologists.
Edited by Prof. D. Petkov and Prof. L. Venter
Van Reebeck Hotel, Gordons Bay, 23-24 November 1998

ISBN: 1-86840-303-3

Keywords: Computer Science, Information Systems, Software Engineering.

Office of SAICSIT: Prof. J.M.Hatting, Department of Computer Science and information Systems, Potchefstroom University for CHE, Private Bag X6001, Potchefstroom, 2520, RSA.

Produced by the Library Copy Centre, University of Natal, Pietermaritzburg.

FOREWORD

The South African Institute for Computer Scientists and Information Technologists (SAICSIT) promotes the cooperation of academics and industry in the area of research and development in Computer Science, Information Systems and Technology and Software Engineering. The culmination of its activities throughout the year is the annual research symposium. This book is a collection of papers presented at the 1998 such event taking place on the 23rd and 24th of November in Gordons Bay, Cape Town. The Conference is hosted by the Department of Information Systems, University of Cape Town in cooperation with the Department of Computer Science, Potchefstroom University for CHE and and Department of Computer Science and Information Systems of the University of Natal, Pietermaritzburg.

There are a total of 46 papers. The speakers represent practitioners and academics from all the major Universities and Technikons in the country. The number of industry based authors has increased compared to previous years.

We would like to express our gratitude to the referees and the paper contributors for their hard work on the papers included in this volume. The Organising and Programme Committees would like to thank the keynote speaker, Prof M.C.Jackson, Dean, University of Lincolshire and Humberside, United Kingdom, President of the International Federation for Systems Research as well as the Computer Society of South Africa and The University of Cape Town for the cooperation as well as the management and staff of the Potchefstroom University for CHE and the University of Natal for their support and for making this event a success.

Giel Hattingh, Paul Licker, Lucas Venter and Don Petkov

Table of Contents Page

HARDWARE SYSTEM INTERFACING WITH DELPHI 3 TO ACHIEVE ACADEMIC INTEGRATION BETWEEN THE FIELDS OF COMPUTER SYSTEMS AND SOFTWARE ENGINEERING

A. Joubert, A. Jordaan

Department of Computer Systems, Department of Software Studies

Vaal Triangle Technikon, Priv.Bag X021, Vanderbijlpark, email joeba@vtt_nt.co.za

Abstract

This paper explores the use of dive computer downloader units and Delphi 3 object-oriented structures to integrate the fields of Computer Systems and Software Engineering in the academic environment. Currently there is a minor degree of system interfacing between the fields of Computer Systems and Software Engineering in the academic environment. Feedback from industry indicates that system interfacing between these two fields need to be addressed. For this paper it was decided to use a dive computer downloader unit as a unique real life application because it allows the addressing of both hardware and software interfacing techniques. Dive data need to be captured from the dive computer and downloaded through a hardware interface device to a personal computer (PC) for permanent storage in a database. By using Delphi 3 with its powerful Windows 95 based 32-bit object-oriented structures and excellent database manipulation techniques, it will be possible to develop a software interface communicating with the hardware devices.

Introduction

Two distinct directions are followed in the Computer Science academic environment namely Software Engineering and Computer Systems. In Software Engineering the emphasis is on developing the student's software analysis, design and programming skills and in the field of Computer Systems the emphasis is placed mainly on the hardware aspects.

This is fine up to a certain point of the student's training program but then the need arises to integrate these two study fields. Industry requires students capable of using both the fields of Computer Systems and Software Engineering to solve unique problems within the computer environment. It is not easy for the student to combine these two fields because they are addressed separately during their academic training. With new technological development in the computer industry, adequate knowledge of both fields is essential. This paper will address the possibility of combining these two fields within the academic environment. Refer to figure 1 for the layout of the discussed scenario.

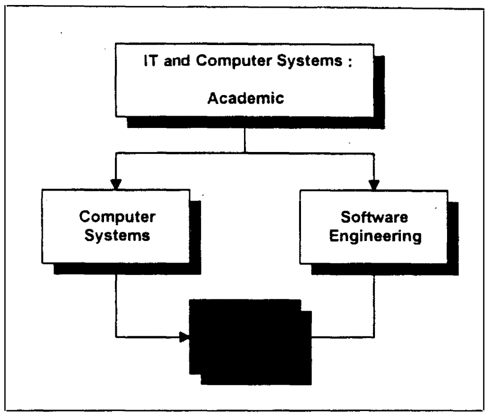


Figure 1. Layout of academic scenario

Computer Systems

In this field, electronics is a key aspect and students are shown the different techniques to build interface with logic circuits and to use these circuits to interface to other circuits. These circuit use electronics and logic devices such as gates, memory and certain micro controller devices. Students are also introduced to various hardware interfacing techniques with the emphasis on hardware manipulation.

Software Engineering

On the software side the student is introduced to Windows 95 object-oriented programming (OOP) techniques, object-oriented design (OOD), rapid application design (RAD), graphical user interfacing (GUI), prototyping and database management (DBM). The emphasis is placed on software development and the student perceives the study fields of Computer Systems and Software Engineering as separate, unrelated entities.

Interfacing

To establish an interface between Computer Systems and Software Engineering, a real world application addressing both the hardware and software fields, is needed. An application facilitating this interfacing is the use of a dive computer, a downloader unit and (OOP) language such as Delphi 3.

The purpose of the project is to develop an electronic downloader unit using hardware interfacing techniques. These techniques will incorporate electronic and logic circuits. The aim is further to investigate the use of an optic down loading unit that can be interfaced to a standard RS232-port on a personal computer. With this type of technology it will be possible to address interfacing between the hardware and software study fields.

Hardware Interface

Dive Computers

Dive computers are used by SCUBA divers and the main function is to measure the absolute pressure in the vicinity of the diver. This pressure is converted into a reading that is represented as a specific depth. Using the *time versus nitrogen tissue loading algorithm* an approximation of the diver's nitrogen levels per tissue type is indicated [7]. These algorithms used in dive computers are based on the 8 or 12-level tissue type models [2]. The decompression algorithm models used with dive computers are based on typical US navy limits. These models calculate the remaining no-decompression time that a diver is left with depending on the depths dived [2].

Dive computers are also used with certain gas mixes where the PO_2 (partial oxygen levels) are calculated. Features also supported on the computers are typical dive time remaining, total dive time, tissue loading, maximum and average depths dived, water temperature and a dive profile generated for every single dive attempted. The dive profile is a graphical representation of the actual dive displayed over time and depth. From this profile certain information can be obtained to improve diving techniques. Refer to figure 2 for a typical display of a dive profile.

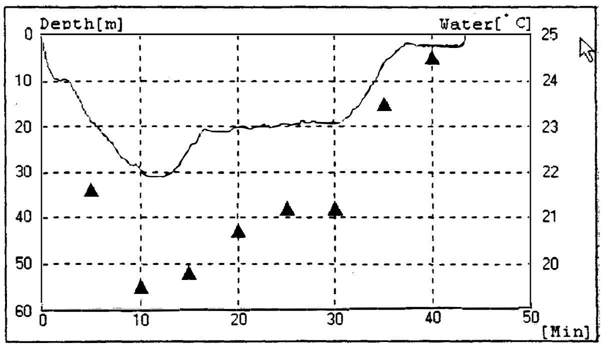


Figure 2. Dive profile

The ascend rate of the diver to prevent the diver from becoming a victim of decompression sickness (DCS) is also monitored. If a diver is ascending too fast he/she is warned by the computer. All this dive data is captured on the dive computer and needs to be retrieved and downloaded for permanent storage on a PC or as a hard copy used for logbooks and certification purposes.

Downloader

Dive computers use either touch technology or optical methods of transferring the data. A special down loader unit is used to retrieve the data from the dive computer through an optical receiving circuit [4].

The circuit converts the optical data into a binary bit stream and then transmits this data from the interface to the RS232-port on the PC. The down- loader unit is connected with a serial cable to the PC's RS232-port where it draws power from the port and transfer the data through this port [4]. The idea is that the unit will be connected to the dive computer without external power sources.

Downloader units are very expensive and not readily available on the market. When dive data need to be downloaded, the dive computer has to be taken to a dive shop. This means the computer can not be used further for approximately two weeks. To perform such a download is also very expensive. It is this scenario that lead to the decision of investigating the possibility of designing a downloader unit that will address both the hardware and software issues as discussed in the previous section.

Downloader program

To accomplish the downloading of dive data from a dive computer into a database, a downloader program is needed. This program requires the following specifications :

- selection of COM ports for connection with the downloader unit
- printer setup for hard copies
- different dive sites
- type of dive
- special equipment used during the dive

- weather conditions
- dive buddies
- activate the actual download
- facility to choose from different dive computers to download from
- filing system for data storage and retrieval
- view the dive profile for each dive
- print the logged data and dive profile

The software interface must be part of this program to enable access of the RS232-port and the data capture through this port. Refer to figure 3.

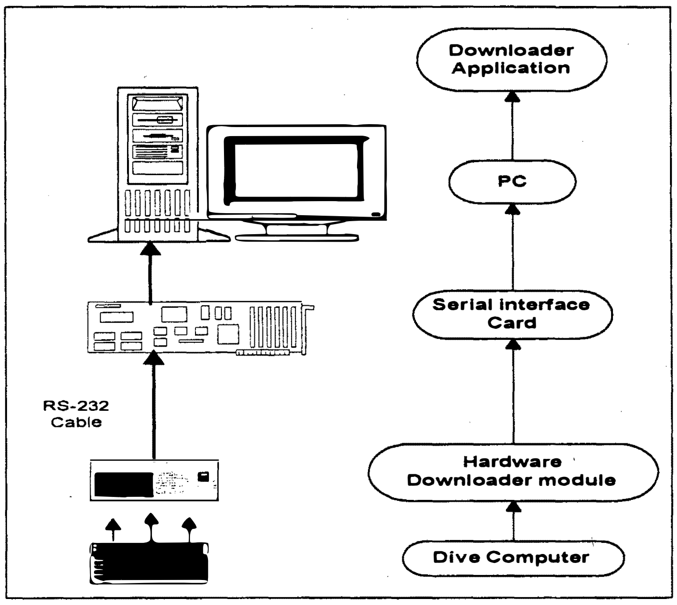


Figure 3. Hardware layout

Software interface

The Delphi 3 programming language is a fourth-generation software tool facilitating both the development of software I/O interfaces and high-level project applications. Delphi 3 runs in a 32-bit Windows environment that supports object-oriented programming techniques. Features unique to the Windows environment and fundamental to fourth-generation language (4GL) programming are (Refer to Figure 4):

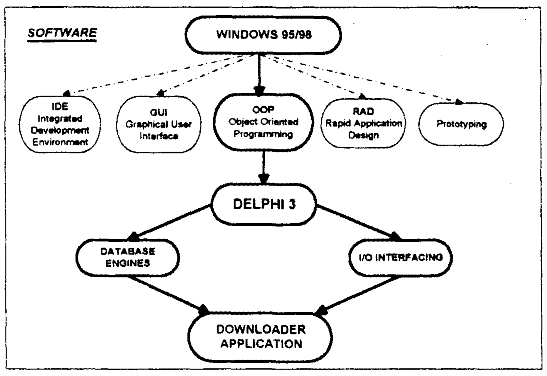

SOFTWARE
WINDOWS 95/98
IDE Integrated Development Environment
GUI Graphical User Interface
OOP Object Oriented Programming
RAD Rapid Application Design
Prototyping
DELPHI 3
DATABASE ENGINES
I/O INTERFACING
DOWNLOADER APPLICATION

Figure 4. Software layout

Integrated Development Environment (IDE)

The Delphi environment has been laid out in a very usable and organized fashion. The IDE is an environment in which all tools necessary to design, run and test an application are present and well connected to ease program development. The tools complement one another and the result is faster and error-free development of complex applications [8].

Graphical User Interface (GUI)

GUI is a type of display format that enables a user to choose commands, start programs and view lists of files and other options by pointing to pictorial representations (icons) and lists of menu items on the screen. Choices can generically be activated either with a keyboard or a mouse [8].

Rapid Prototyping

With rapid prototyping, working models of an application's GUI can be created early in the design process. This interface can then be presented to customers for feedback about the object placement and other features important to the users [8].

Rapid Application Development (RAD)

RAD implies using a set of the latest generation tools to develop applications quickly with very little turn-around time. This is accomplished by tools that allow the programmer to perform large leaps in functionality with a minimum of coding. An example would be using the Windows-95 **Open** and **Save** dialog boxes, instead of writing code for them from scratch [8].

Object-Oriented Design (OOD)

Most programming languages have a large variety of operations which can be performed, but only a small set of nouns to describe the objects. Even those languages that have object capabilities are usually flat (they cannot inherit attributes from their parents).This is not a reliable model of the three-dimensional real world. Delphi provides a realistic three-dimensional set of nouns that allow programmers to successfully describe objects in a 3D way [8].

The goal of OOD is for each module in the system to represent an object or a class of objects in the real world that interact with one another [8][5]. An example of an object is **Image**. This image object has properties (attributes) and events. Properties the image may have are **Height**, **Width**, and **Name**. Possible values for each of these properties are **20cm**, **35cm** and **Undersea**. These properties are manipulated by changing their values [9][5]. Events indicate the actions that can be performed on the image eg. **Enlarge**, **Hide** and **Minimize**. Events are manipulated by writing code for it [9][5].

Form

The heart of every Delphi Windows application is the form (or window). The window is the base object upon which the rest of the environment is built [8]. Every object needed for a specific application is placed on a form and from there these objects are manipulated and programmed.

Database Management

Delphi provides a development environment with tools and components that are especially useful to database application developers. The database features and tools in Delphi are tightly-integrated to ease the design, implementation, deployment and maintenance of database applications in a variety of computing environments [1].

Delphi uses a new, open database architecture allowing the programmer to create database applications using any database engine available [6]. Possible database engines that can be used are Paradox, dBASE, Access and FoxPro. Database Desktop (DBD) can be used to browse and modify existing Paradox and dBASE tables or create and populate new ones, create indexes, define referential integrity, and create database-level validation and business rules. BDE aliases that set the absolute path for applications, can also be created and browsed. The DBD is a stand alone utility that runs outside the Delphi IDE. [1].

Delphi provides a series of data-aware controls to provide a user interface (UI) to data in all types of Delphi database applications. Data-aware controls display data from fields in a database record and can permit users to edit that data and post changes back to the database. Data-aware controls are placed on Delphi forms making them visible and accessible to users [1].

Downloader application

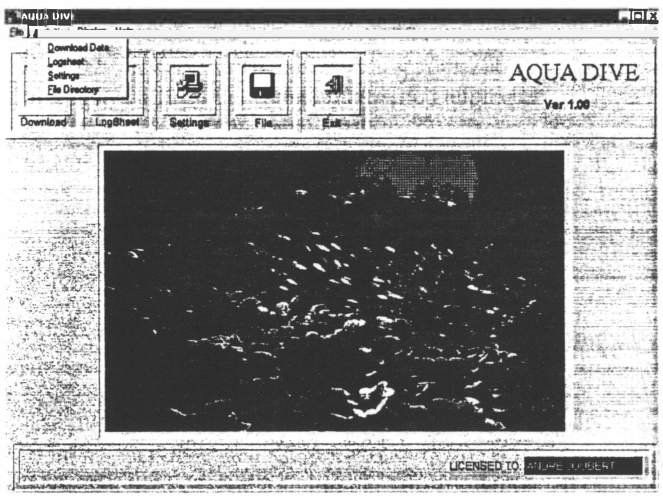
Software I/O interface

The Delphi 3 software I/O interface being developed will accept the encrypted dive data from the downloader unit, decrypt and convert it into an acceptable data set format and write it to predefined database tables. From there the downloader application will access and manipulate this downloaded data by means of data-aware controls linking the database to the application.

User interface

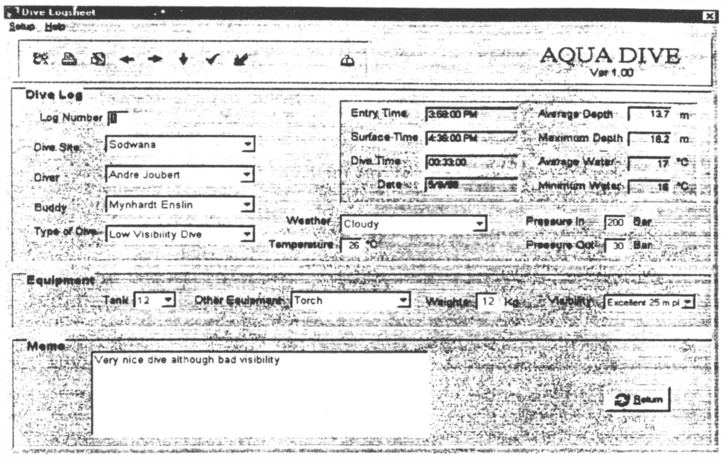
The front end visible to the user is the complete downloader application. Refer to Figure 5. With Delphi 3 a visually, inviting graphical user interface can be created without excessive effort.

Figure 5. User interface



Referring to Figure 5, the drop-down menus from the main menu become visible by clicking on the menu options. Alternatively, the user can click on the pictorial representation of the menu options. Shortcut keys are also available for each item on the main menu. By pressing the **Alt + A + D** keys on the keyboard, the action initiated will be the same as clicking on the **Download Data** option of the **Activities** drop-down menu. The complete downloader application is event-driven. This means that the current active window (form) will wait motionless for the user to take some kind of action by clicking on an object or typing data into the available space. The application will react depending on the particular object selected. The dive log sheet (Figure 6) displays a clear example of how the tables in the downloader application database can be accessed and manipulated to produce a specific graphical output. The downloaded data in the database (*Entry Time, Surface Time, Dive Time*, etc.) can be viewed but not modified because the data entries need to be kept valid for true record keeping.

The user has to provide additional information to complete each logged dive. The size of the cylinder used, the dive conditions, site and any other relevant information about the dive can be added to the database. This will enable the diver to keep a log of all the dives at a certain site or dives over a period of time.


Dive Logsheet

Setup Help

AQUA DIVE
Ver 1.00

Dive Log

Log Number: 1
Dive Site: Sodwana
Diver: Andre Joubert
Buddy: Mynhardt Enslin
Type of Dive: Low Visibility Dive
Entry Time: 3:58:00 PM
Surface Time: 4:36:00 PM
Dive Time: 00:33:00
Date: 5/9/98
Average Depth: 13.7 m
Maximum Depth: 18.2 m
Average Water: 17 °C
Minimum Water: 16 °C
Weather: Cloudy
Temperature: 26 °C
Pressure In: 200 Bar
Pressure Out: 30 Bar

Equipment

Tank: 12
Other Equipment: Torch
Weights: 12 Kg
Visibility: Excellent 25 m pl

Memo

Very nice dive although bad visibility

Return

Figure 6. Dive Log sheet

A graphical representation for each logged dive is generated as soon as all the necessary information is provided by the user. (Figure 7). These individual profiles can be saved on disk or printed out for the user to be stored as hard copy. The dive profile in Figure 7 is generated from the downloaded data as shown on the dive log sheet in Figure 6.

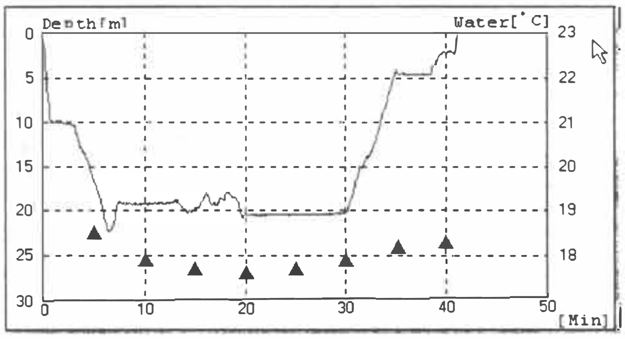


Figure 7. A dive profile

Conclusion

With this specific research project the two fields of Computer Systems and Software Engineering can be integrated by means of the software I/O interface accepting the downloaded data from the dive computer and writing it to a database. Refer to Figure 8. The Delphi 3 object-oriented programming language facilitates both the development of this interface and the downloader application.
Computer Systems students enrolled for Delphi 3 as a subject, have knowledge of both the mentioned fields of study. By introducing them to similar projects as the dive computer downloader application, they can be shown how to integrate the Computer Systems and Software Engineering fields by creating software I/O interfaces in Delphi 3 for hardware units. The result of this research project will enable academics to deliver more competent students with knowledge of how to integrate the fields of Computer Systems and Software Engineering.

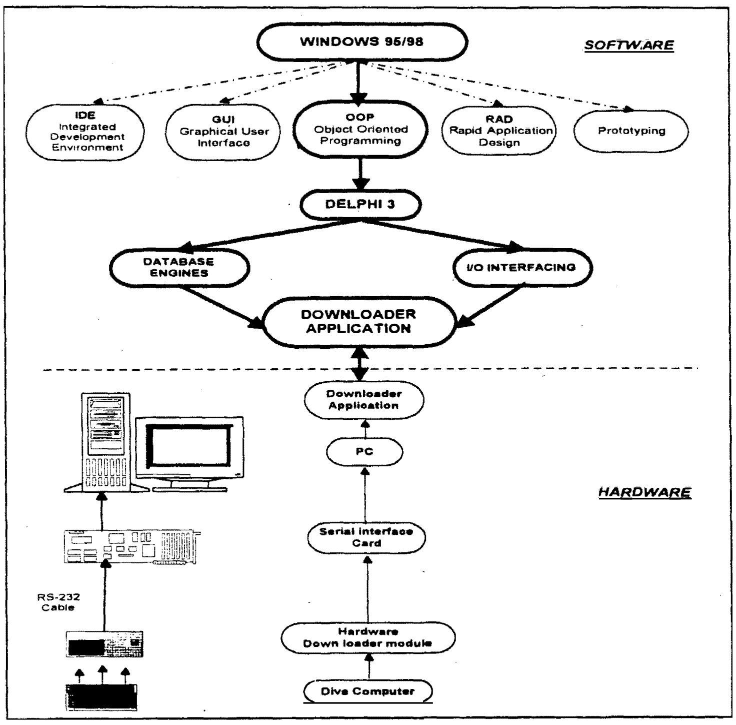


Figure 8. Complete research project

References :

1. Borland. *Delphi 3 for Windows 95 & Windows NT*, Borland International, Inc., 1997.
2. Heine, J.N. *Mastering advanced diving technology and techniques-NAUI*, Mosby Lifeline, 1995.
3. Learning the JAVA language, http://java.sun.com/docs/books/tutorial/java/
4. Minasi M. *PC Upgrade & Maintenance Guide*, Sybex, :916-930, (1997).
5. Newman Alexander, et al. *Using Java Special Edition*, Que, :18-19, (1996).
6. Optoelectronics.. Texas Instruments, 1992.
7. ORCA® Division of EIT, Inc, 1992.
8. Osier, D. Grobman, S., Batson. S. *Teach yourself Delphi 2 in 21 days*, SAMS, :14-225, (1996).
9. Rubenking, Neil. J. *Delphi 3 for Dummies. A reference for the rest of us*, IDG Books Worldwide, :1-252, (1997).