



Research of System Software Tool with Innovative Technological Trends of Detailed Design in Advanced Computer Software Engineering & Quality Assurance for 2012 -2013 Applications

Bedre Heeramani¹, B.Nagaraj²

Lecturer, Department of Computer Science ,
Sahyadri Science College(auto), Shivamoga
Shivamoga , Karnataka, INDIA

Abstract: Whole universe is composed of five elements-the earth, water, fire, air & space which frames the boundaries for the challenging extension of computer science & engineering with applications. The rapid growth powered by computer technology arising query-can intelligent design plays a vital role in saving planet. Historically, design seems to be everywhere omnipresent right across the world. In today's technological era of computer science & engineering it is quite interesting to analyse, design, develop & maintain the software –application software as well as application. Certainly the designers are in a key position to influence wide spread usage of quality solutions with trends of software engineering advancements. Enormous extension of the computers powered by software and hardware emerging technologies with the views of solving challenging problems made it possible for widespread acceptance in most of the fascinating fields including engineering, research, medicine, pharmaceutical, science & technology, arts, commerce, management, economics, politics, industries, universities & schools, government departments, private sectors, business & e-trading, e-banking, e-commerce, reservations, tourism etc computers reputation skills have varieties of fastest speeds & high accuracy with the promise of providing the quality solutions in finite number of the precise calculations unless there is proper software will be created & used appropriately. System software plays significant role in the execution of the application software utilizing the hardware efficiently. This paper focus on challenges & research issues of the a system software loaders & also on software engineering trends for motivation of optimism of algorithm space and time leading enormous quality growth of software & global strategy of applications for the image processing, pattern recognition, robotics, expert systems, artificial neural networks, multimedia applications & cloud computing with quality assurances for the effective usage of software.

Keywords: Memory, Compiler, Loader, Multimedia, Algorithm, Operating systems, Software, Hardware.

1. Introduction

The power of design lies in creating any fascinating aspects of life including fine art design, game, graphic & web design, industrial design, product design, architect design, interior design & fashion design using digital computer more accurately & also effectively provided the software detailed design is of the highest top ranking quality assurances. Algorithm is a finite sequence of steps that leads to the solution of problem [37]. Software is the collection of the programs [17]. Software(system software[40]& application software[12])together with the hardware constitutes a computer system. Application software is for the particular application System software is for the proper working of system & to provide facilities to user to get the software to be executed by corresponding hardware. Most of the popular system softwares includes assembler, macroprocessor[], compiler[40], linker, loader, editors. Loader is a system program that performs the tasks of allocation, linking, relocation & loading[1]. Software engineering is the branch of computer science and engineering that deals with the various aspects of software including the requirements analysis, design, coding, testing & maintenance. The globe has witnessed the use of cycles of the generations. The globe has witnessed the use of the thousands of human languages. Exciting & thrilling true is w.r.t. computer there are only three languages and they are machine level language, assembly level language & high level language. The first programming language for the computer system uses only zeros and ones, called as machine level language. It is difficult to program, understand, modify & enhance. But its merit is to understand by ALU directly. Some of the demerits are removed by assembly level language which uses mnemonics requiring system software assembler. High level languages (procedure oriented, object oriented, function oriented, logic oriented) including Basic, cobol, Fortran, pascal, Lisp, algol, C, C++, visual basic, visual c++, Java etc uses either compiler or assembler to convert the high level language statements in the form of the machine level language.

Among data, information, knowledge & intelligent, it is natural that intelligent is at the highest level [41]. An operating system is a program that acts as intermediary between a user of computer & hardware. The entire users of the globe are presenting various fascinating & motivating problems in front of the computer science engineer groups & there by in this race programmers are trying to make computer system capable of making best tremendous problem solver. Multimedia is the combination of audio, graphics, animation, video, text and image[20].

2. Methodology-Detailed design & loader schemes

2.1 Detailed design

Earlier only multidimensional companies and large scale industries opted for design concepts. But now even small & medium scale industries have become design conscious, which has made the field more lucrative & demanding. Sustainability designs create the possibility for every demands on planet to have their basic rights, needs & motivations for the overall integration of the growth. The solution of the problem is obtained with the software engineering by performing its stages like requirement analysis [17], system design[17], detailed design[18], coding[17], testing[17] & maintenance[18]. The design is broadly classified into system design and detailed design. The detailed design is carried out only after system design i.e, high level design. The low level design-detailed design emphasizes on algorithm or the logic design. User requirements are acquired & analysed by requirement analysis & transferred to modules by system design. Next step is detailed design which strictly follows the the module boundaries laid out by system design. The logic plays important role for the overall performance of the software. A software may take years, months or days or hours or minutes or second or fraction of milliseconds depending only upon the strength of the algorithm!. The output of the detailed design is passed to coding & then testing. It is clear and surprising that the cost of the hardware is decreasing but cost of the software is increasing due to accumulation of the expenditures spent over all the phases of the software development. As per the survey it has found that error discovered later stages will increase the overall cost of software. In order to reduce the errors & to have fastest, accurate, definite solution detailed design has key role by using the design notation called as Process Design Language(PDL).

Verification shows that detailed design meets the specifications laid down in system design. Most commonly used verification methods are:

- i) Design Walk throughs
- ii) Critical Design reviews
- iii) Consistency checkers

Design walkthroughs is done in an informal meeting called by the designer or by the leader of the designer group. A small group is created by having designer, colleagues, group leader. The designer explains the logic step by step & members of group ask questions. As design is explained this method has merit of uncover some errors by designer.

Critical design review slightly differs by having focus on search of errors & using check list during study of design as well as review meeting. The contents of check list includes questions covering areas of the data formats, statement of data structures, exceptional condition handling, loop conditions, module logic complexity etc.

“Fig 1”, Consistency checkers detects the design defects, checks for the interfaces and the existence of all modules as specified in the system design & they are actually compilers[jalot]

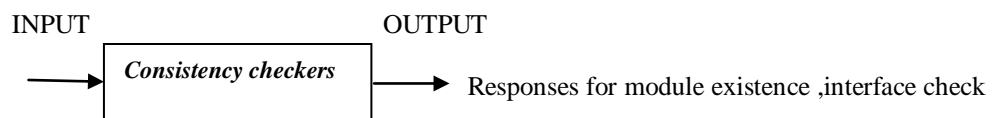


Fig 1: Working of Consistency checkers

2.2 Existing Loader schemes

Some of the popular existing loader schemes includes

- i) Compile & go loader / Assemble and go loader
- ii) General loader scheme / Absolute loader
- iii) Relocating loader(ex: Direct linking loader)
- iv) Binder & module loader
- v) Dynamic loading

Easy compile & go loader runs in one part of the memory and places directly the assembled machine instructions and data as they are assembled into their assigned memory locations. In general loader scheme, assembled program is saved in object decks as they are assembled and loaded whenever the code was executed. Absolute loader is a type of general loader scheme with allocation linking done by the programmer. Relocation and loading performed by assembler and loader respectively.

Relocating loader, example-BSS loader (Binary symbolic subroutine). The output of relocating assembler using BSS scheme is the object program and information about all the other programs it references. It gives the information about locations

which are dependent on the core allocation. BSS loader scheme is used on computers with a fixed length direct address instruction format. Binder performs allocation, relocation & linking. Two types of binders are core image binder and linkage editor. Core image binder is simple & fast with specific core allocation of program done at time the subroutines are bound together. Linkage editor allows flexible allocation and loading scheme by keeping track of relocation information so that the resulting load module can be further relocated and there by loaded anywhere in the core. Dynamic loading (load on call) has mutually exclusive subroutines with the overlay structure and overlay supervisor. Dynamic linking is mechanism by which loading and linking of external references are postponed until execution time in order to avoid the linking of these subroutines that are referenced but never executed.

2.3 Proposed loader schemes & PDL schemes

A new type of loader has to be developed so that it is faster and consumes less memory space. It should perform relocations encountered while loading modules for multiple subroutines thus freeing programmer to remember address locations. Loading and linking are postponed with usage of temporary data structures such as linked list [] to minimize the complexities and this proposed loader inherits the profits of the some loading schemes. If many facts considered for build of system software loader ultimately it consumes more space there by increasing speed loading challenges. The new schemes for the loader & PDL are discussed in section 6

3. Merits

- Improved speed of execution of programs
- Advanced complex problems design can be simplified and verified easily
- Reduces the wastage of time in testing
- Increase accuracy
- Reliability and quality performances & fault tolerance can be increased
- Overall cost of software can be reduced

4. Demerits

- Time consuming for the designs
- Difficult to extend applicability for multimedia programs []

5. Applications of System software-loader & software engineering consistency checkers

Loader finds primary importance for the execution of the any program and finds uses in almost all areas. Consistency checkers play a key role for the design sections of software engineering. Some applications of System software-loader & software Engineering consistency checkers for quality software are as follows:

- Fast response Data Flow Super Computer Applications
- Expert systems
- Artificial neural networks
- Pattern recognition
- Image processing
- Finger print identification
- Animation and games
- Online services and e-commerce
- Medical Engineering
- Compiler constructions
- Editors
- Remote sensing
- Robotics
- Video on demand
- Cloud computing
- Telecommunications
- Reservations
- E-billing
- Internet
- Intranet and extranets
- Interactive multimedia applications
- Mobile computing applications
- Modern embedded computer challenging applications of distributed computing

6. Discussion: Research Dimensions & challenges

Actually loader prepares the program for the execution and initiates the execution. Several blends of the loaders area available as in section 2. Compile and go loader wastes the time for reassembling program and also portion of the memory is wasted. If source programs of multiple segments are in different languages, it is difficult to handle them. In general loader schemes programmer must specify the start of address of program and take care not to assign the two subroutines same locations with tediousness of remembering address of each subroutines if multiple subroutines are present. BSS loader increase the size of the object code due to use of the transfer vector. Direct linking loader wastes the considerable amount of the space. If more subroutines are present the four functions of loader will be extremely time consuming. Linkage editor binder is complex. maintenance of overlays is overhead for dynamic loading. Dynamic linking postpones the binding process until execution time by increasing the complexity. With the scenario it is directly evident that memory wastage, overhead of operations and complexity with time consumptions are overlooked and requires the resurgence of the new loader incorporating the speed, simplicity and flexible operations which is challenging task. Another issue is in case of variable partition scheme of memory with the usage of first fit, next fit, best fit and worst fit algorithms. Using even the first fit or best fit it leads to internal fragmentation i.e., wastage of memory inside allocated block. If process requires 100Kbytes and hole of size is 125Kbytes is selected it leads to wastage of 25Kbytes. Here once after selecting the hole, the loader has to divide into two partitions 100Kbytes and 25Kbytes. It should use the partition created with 100Kbytes and provide chance to utilize the 25Kbytes of other partition to some other process. This eliminates the internal fragmentation. Main focus is on error handling. All errors should be handled internally without any user intervention. Computer science engineers and software designers are challenged to create innovative designs for every softwares with priority strategy of preventing, reducing, sharing, changing and restoring. Designers make decisions that result in millions of items manufactured including role of software for product design, transportation, interior space and equipment design, graphic design I-pad, mobile, pc, remotes etc. Where the quality of software rely on detailed design algorithm which in turn depends on verification method used during detailed design. Designing print of building, creating new products by combining art, science and technology, product makeover for the industries, games, animations, graphical user interface activate demands full of challenges to verifications of section 2. Various challenges and research issues are as follows:

Design walk through is manual method and it can not be effective as design reviews. It has to be resurged with the emphasis on verifying each of steps but time taken is more and hence it is inevitable to equip this method with consistency checker playing some role in assisting to go through the design logic as well as group can have discussion. New method of verification is to be created in collaboration of design walkthroughs and consistency checkers. The performance is the main criteria with which system will operate under all conditions is a challenging task and resurgence of fault tolerance with repeated error checking and to correct them immediately and that is too fast with automatically, the check list is generated for the given problem. But always the problem is not same and creating a check list by software is again the challenge. None of the verification method ensure sent percent accuracy of high quality with less time and less cost. All the three methods are integrated and software support for it is enclosed to enhance the verification. But integrating them is a big task.

There is no verification method which takes care regarding all facts like of time and space consumed by algorithm, users requirement incorporation in design, quality, options for slight corrections during the testing, user satisfaction, environmental conditions, applicability area, option for expansions, misuses of software and also plentiful supports to its users from diverse fields. A verification method has to be developed which works like human brain in computer style with the care of all above factors that is too safely, fastly, accurately, effeciently, effectively with high performance by decreasing cost. A verification method can be developed as the part of the system software which can verify for any design and judge it is correct or wrong by adjusting itself for any mentioned problem by user. For a single problem by user it is possible but same software for handling multiple tasks of advanced nature is difficult. Further it should support automatic creation of detailed design algorithm also for any given complex problems.

7. Conclusions

The research issues of loaders are interdependent on other system softwares and also on the memory with a challenging task. In near future new types of memories are to be developed Software design has to change in a big way that posses quality characteristics to think totally new on all aspects of design, restructuring to remain competitive. In coming days the softwares are expected to be witnesses by world, which will work on principle of human brain in computer style ther by design as well as verification is also done automatically for any problems by computer.

References

- [1] "Mobile Communications", Jochen Schiller, Addison-Wesley, 2000.
- [2] "Ad hoc Networking", Charles Perkins, Ed., Addison-Wesley, 2000.
- [3] "Mobile Ad hoc Networking", Basagni, Conti, Giordano, Eds., IEEE/Wiley Press
- [4] "Computer Networking: A Top-Down Approach Featuring the Internet", Kurose and Ross, Addison-Wesley, 2002.

- [5]. Digital design and computer architecture by Harris
- [6]. Computer organization & architecture: Designing for performance by Stallings.
- [7]. Structured Computer organization by A.S. Tanenbaum.
- [8]. Computer system architecture by Morris Mano.
- [9]. Advanced computer architecture-Hwang & Briggs
- [10]. Computer architecture – John p Hayes.
- [11]. Computer organization- Hamachar , Jaky & Varenisc.
- [12]. Computer fundamentals, Rajaraman.
- [13]. Microprocessors- Goankar
- [14]. Microprocessors- Ram
- [15]. Digital Systems-Moris Mano
- [16]. Computer networks- Tanenbaum
- [17]. Software Engineering- Pankaj Jalote
- [18]. Software engineering-Pressman
- [19]. Computer graphics-Steven Harington
- [20]. Multimedia making it work- Tay Vaughan
- [21]. Paper by Bedre Heeramani, B.Nagaraj: Volume 2, Issue 4, April 2012 ISSN: 2277 128X,International Journal of Advanced Research in Computer Science and Software Engineering Research Paper:”Research Issues of Interactive Multimedia for Advanced Computing &Communication for Challenging Applications” paper id-v2i4:0048.
- [22]. Paper by Bedre Heeramani, B. Nagaraj: Volume 3, Issue 3, April 2013 ISSN: 2277 128X,International Journal of Advanced Research in Computer Science and Software Engineering Research Paper:”Emerging technological trends of 2012-2013 data flow supercomputer design issues for challenging applications” .paper id-v3i3:0301
- [23]. Bedre Heeramani , B.Nagaraj: Volume 3, Issue 4, April 2013 ISSN: 2277 128X,International Journal of Advanced Research in Computer Science and Software Engineering Research Paper:” Research Dimensions of Advanced Mobile Computing Technology Security Issues for the Complex Applications” paper idv3i4-0499
- [24]. Computer programming in c- Dennis Ritchie
- [25]. Computer networks –Frozen
- [26]. TCP/IP programming- Douglus E Comer
- [27]. Computer Architecture-John L Hennessy & david A Patterson
- [28]. Computer organization &design, revised fourth edition by john L. Hennessy.
- [29]. Digital design and computer architecture by Harris
- [30]. Computer organization & architecture: Designing for performance by Stallings.
- [31]. Structured Computer organization by A.S. Tanenbaum
- [32]. Computer system architecture by morris mano
- [33]. Advanced computer architecture-Hwang & Briggs
- [33]. Microprocessors-Gaonkar
- [34]. Programming in C++ - Stroustrup
- [35]. Digital principles-Bartee
- [36]. Operating Systems -Galvin & Silberhatz.
- [37]. Introduction to algorithms-Coremen, leiseron, rivest , stein
- [38]. The design and analysis of algorithms,Aho, Hopcroft, ullaman.
- [39]. Algorithms and Data structures,Wirth
- [40]. Systems programming- Donovan j
- [41]. Artificial Intelligency -Ritchie E
- [42]. Unix-Sumitabh Das
- [43]. Java programming –Weber.
- [44]. paper by Bedre Heeramani, B. Nagaraj:IJEA, Volume2,issue-1,may2013,”Research of modern embedded computer system challenging program techniques for the global computer science & engineering applications of distributed computing for 2012-2013”