IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Challenges In Computer Science-A Review

S. Divya Meena
School of Information Technology and Engineering, India
VIT University, Vellore., India
divyameena.s2010@vit.ac.in

Abstract

This paper presents the connection between Computer Science and other fields of Computer Science and it provides some of the challenges in these fields. Here theory plays an Impact in the connection between these fields.

Keywords: Computer Science, Networking, Computer Architecture, Operating System, Data Base System, Software Engineering, Object Oriented Programming Language.

Introduction

Problems in computer science are considered unsolved when a particular expert or several experts in the field disagree about a solution to a problem.
Theoretical Computer Science (TCS) is the "science" underlying the field of computing, i.e. it is the scientific foundation of Computing. Theory is an essential part of almost everything we do today in the field of computing. It is a guiding light for almost all models. Using theory, we can better understand the model and by that we can provide solution to Challenges.
In this paper, we shall discuss few challenges in Computer Field. Though the solutions are not provided here, I hope that these challenges can be faced easily once the theory behind it is understood.

Principal behind Theory

Using the following basic principle, Theory can be better understood.

i). *Find the Abstract:*

By avoiding the irrelevant details, we get a precise idea about the content. This can be then analyzed and modified based on the requirement.

ii). *Derive a Link:*

Discover a link between the problem at hand and a similar previous problem. Once a link is established, the similarities between them can be found. The ideas can then be transferred from the old model thus paving a way for better understanding.
With these two basic principles, theory behind any model can be well studied.

Challenges In Networking

i) Communication Protocol Automata:

Networking and Software Engineering face same problems in terms of Communication. But Networking being relatively simple in terms of Communication Protocols have some better means to overcome the problem. One such possibility is using I/O Automata. Specifying, constructing and verifying communication protocols are mostly done by hand.
The main challenge here is to Automate the process of Constructing, verifying the communication Protocols rather than doing manually.

ii) Routing Protocols:

OSPF and MPLS are the two routing protocols used on Internet for Congestion control and node failure recovery. However, as both the protocols requires more control, the communication between routers and the computation increases. This introduces scalability problems, a problem currently addressed by splitting the network into sub-areas and restricting the routes between the sub-areas.
So, the challenge here is to develop better routing mechanism that is real-time adaptive. Indeed, network experts and operators are quite doubtful of real-time adaptive routing protocols, based on experiences of major "thrashing" when this was tried in the early ARPANET.

Challenges In Database

i). Data Storage:

As the database grows larger, the memory storage grows larger. Hence, memory hierarchy comes into picture. Here we need to more accurately model the differing costs of accessing data and storing intermediate results in cache, RAM, disk, tape, etc. when attempting to reduce the cost of executing database queries.

Here the Challenge is to invent new ways of executing queries on large-scale data, using secondary-storage models of cost, and possibly develop parallel and distributed processing when available.

ii). Query Optimization :

Query Plan is the answer to a query. This is a detailed algorithm that is constructed from basic database operation such as Projection, Join, etc. Query can have any number of Query Plan based on SQL language. Query optimization is the process of finding the most efficient plan.

The challenge is to deal with complex operations and to optimize the queries when sub-queries are from variety of sources.

Challenges In Operating

i) File System Protection:

A set of Programs can be used as a source to gain access to confidential files. This keeps security at stake. For example, both the owner of a mail and his opponent can access the mail sent by owner. A mail program can be used by an opponent to gain access to the mail sent by owner. Here, both have the same access pattern. This problem will become even worse as we increase our use and dependence on "mobile code".

The challenge is to model the secure execution of un-trusted code. Browsers currently attempt to do this, but the frequent reports of holes in their security schemes suggest that better modeling and analysis are needed.

ii) Data Access Optimization:

For tasks that are performed by operating systems, the main bottleneck is the Response Time (times spend obtaining data from storage devices). To improve the Response time efficiency, the layout of the Data has to be designed based on the often encountered access method.

The challenges are to model Automatic techniques for "optimal" dynamic rearrangement of data in response to access patterns and fast online reordering techniques for access requests so as to minimize overall latency, given a fixed data layout.

Challenges In Computer Architecture

i) Exploiting Instruction Level Parallelism:

Recently developed processors exploit more and more hardware for improving performance by a variety of ways, including instruction-level parallelism.

The real challenge here is to develop a processor that does not exploit the computer resources and to develop an Algorithm in which the responsibility of programmer could be stated clearly.

Challenges In Software Engineering

i) Software Specification:

Requirements make sure how well the software system meets the user requirements. But they are ill-specified in most of the cases because they are written in informal natural language that is too long and full of redundant and inconsistent standards.

MSC (Message Sequence Chart) is a Precise formal methods for specifying desired behavior, but this do not have sufficient expressive power to completely specify what we might want from a system. LSC (Live Sequence Chart) is a recently introduced mechanism for adding expressivity, but it is not yet is clear that they are exactly what are needed, nor is there currently any automatic way for going from an informal specification to a formal one.

The Challenge is to devise formal specification techniques for software systems that can adequately express the functionality and behavior desired.

ii) Automatic Transformation of Requirements to System Design:

The requirements talks about overall system behavior, whereas the system design specify how they work and interact. They are usually represented using State-Chart or other Automata-theoretic model. It is a long step from the first to the second.

The Challenge is to design algorithms and tools for converting requirements into system designs and provide restrictions on what sorts of specifications are allowed and what types of systems designs are allowed. By this we can save our time and work.

iii) Understanding Complex System:

Software systems are usually built by designing system architecture that consists of interdependent parts. On ignoring some important design, the software tends to fail or run partially.

The challenge here is to develop a theory that can understand the incomplete programs. This theory should provide the basis for testing and evaluating designs that may be partially built.

iv)Transformation of System Design to Code:

Transforming a system design into code requires some programming language and this is simply viewed as High-level compilation task. This involves the use of its own algorithm which is not clearly studied.

The challenge here is to develop an Automatic Compilation technique that describes the algorithm clearly and transform the design into code automatically. It is an area with many remaining opportunities for theoreticians.

v) Software System Query Language:

Modern software systems are difficult to understand because of the complex entities. Existing software system can be viewed as a Database in order to understand the Properties of system.

The challenge is to develop a Query Language like SQL that inquires about the existing software. Along with, it should also develop a system that could devise algorithm for a given problem. This algorithm can be used to answer the queries put forth by the Query language developed for this.

Challenges In Object Oriented Programming

i) Simplification of Programming:

In Object programming, control flow is secondary and data are primary. Its features are extremely useful in providing encapsulation and protection mechanisms and promoting modularity and code reuse. The current popular programming languages are Java and C++.But, Programming language theorists are ever exploring the structure and meaning of such languages in the desire to simplify the programming process.

The challenge is develop means to understand the semantics of objects, especially their interactions with types.

ii) Way to Express Programming Language:

The set of definable functions on the integers is the set of partial recursive functions in all programming language. However, most programmers make distinctions between languages. Some languages are useful for quick programming of simple tasks while others may be better for producing large-scale maintainable software systems.

The challenge is to devise a way to express the programming language differences. One possible way is by comparing the ability to localize program design decisions. Another possible way is compare languages according to the ability to express operations on program modules.

Central Issue Of Computer

Protocol security:

For protocol security, simple strong formalisms can be developed. Present formalism can just find bugs in the protocols. There is no formalism that gives protocol security.

The challenge is to develop automated tool for verifying correctness of protocol. On developing such formalism, we can avoid the problem of underlying protocol's security risks; avoid bugs during implementation and hence resulting in software security. In most cases, the current security notions do not guarantee that security is preserved when many instances of various protocols are executed concurrently, a very real threat.

Conclusion

In this paper, we discussed about the major challenges in different fields of computer Science. Perhaps, I shall write the possible solutions in my next paperwork. These fundamental issues are although at the core of the field; provide an opportunity for the future generation. Finding a solution for all these not only increases the understanding ability of subject at hand but also eases our work. Every Business Corporation, NGO, and government organization would like to create innovative solutions to real world problems. In the modern world, knowledge creation would indeed become the most important means for success and survival not only for an individual but also for a company.

References

[1] Challenges in Theoretical Computer Science –Paperwork by Ed Clarke, David Harel, Anna Karlin, Gerard Holzmann, Daphne Koller, Butler Lampson, Jeff Ullman, and Uzi Vishkin.

[2] Database System Implementation, Prentice Hall, 1999,Garcia-Molina, Ullman, and Widom.

[3] Challenges at the Interface of Data Analysis, Computer Science and Optimization by Gaul.W.A; Geyer Schulz.A Human Language Technology-Challenges for Computer Science and Linguistics by Vetulani.