Trees and Forests: A Framework for Parallel Processing and Management of Networks

C. P. E. Agbachi

Department of Mathematical Sciences, Kogi State University
Anyigba, Kogi State, Nigeria

Abstract—*It is a cardinal principle in program engineering to match data structures to tasks in order to lay the ground for efficient and concise algorithms. One of such data structures are trees. They are very versatile and in forests provide a model interface for parallel processing of survey networks. This paper evaluates the modality and applications in Geomatic Engineering.*

Keywords—*Datum, Origin, Digraph, Tasks, Multiprogramming, Multithreading, Concurrency.*

I. Introduction

Fig.1

Trees and forests abound in nature, Fig 1. There are depending on location, palm trees, pine trees etc. A forest is group of trees with much in common, such as locality and plantation. Thus there are pine forests, rain forests etc. In computer science, they are a source ofinspiration and provide modelsfor data organization.

A. Data Structures

A tree is an abstract data type [1, 2]. As with natural trees, it has a root and grows likewise, characterised by branches and terminating at the leaves. There are two principal classifications:

i. Binary Tree
ii. Multi-way Tree

Whereas the binary tree is restricted to only two branches, a multi-way tree is not limited in the number of siblings. Often the latter is grouped in form of blocks as B-Trees, as in disk storage systems or transformed to binary model for easier processing.

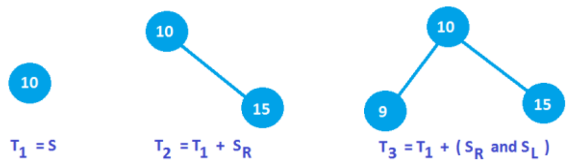

Fig. 2

Essentially, the definition of a tree is of recursive nature and applicable to the binary tree. As described in [3], $T = S/T + (S_L|S_R)$ where T is a tree and S is a subtree. A leftward addition is represented by S_L, likewise adding node to the right is signified by S_R. With slash standing for *or* and the vertical for *and/or*, examples are $T_1 = S$, $T_2 = T_1 + S_R$ and $T_3 = T_1 + (S_R \text{ and } S_L)$ as illustrated in Fig 2.

Depending on applications, three key traversal routines are available in Pre-Order, In-Order and Post-Order transformations. However, the main appeal in this research is representation of networks and for which further evolution of trees is of interest.

1. ***Graph:***

Graphs and trees are complementary in their definitions.A graph is a tree without root, while a tree can be described as a graph without cycles. In developing applications, data structures evolve from generic definitions. Other variants of graph include directed and weighted options.

A digraph is a directed graph with the complements of a tree, Fig 3. It is perfectfor modelling patterns and relationships. As such it is an ideal representation for a network.

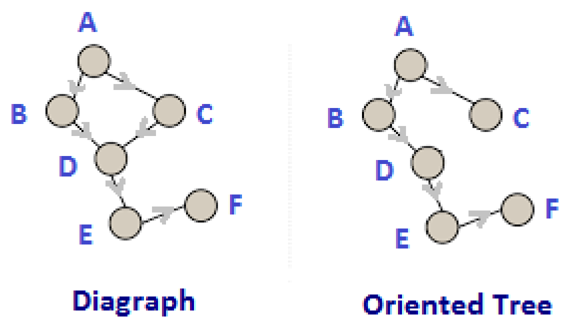

Fig. 3

II. SURVEY REVIEW

Surveying is a measurement science providing requisite information for mapping, engineering and application in diverse disciplines. It involves acquisition of large amounts of data for computation and processing. In general, management falls into two categories:

- Field Survey
- Office Administration

A. *Field Survey*

Field survey undertakes execution of a project to design specifications. And in this consideration and requisite computations the datum and origin of the survey must be established. There are two main types of survey:

- Control Survey
- Detail Survey

A control survey is used to establish known positions in the area of interest as framework for reference. Detail surveys rely on these fixed positions to create or update map information.

1. *Control Survey:*

Control surveys are either for vertical, horizontal control or both in 3 dimensions. The methodology though is the same, Fig 4, as in transfer of coordinate information from the root or origin of the survey [4].

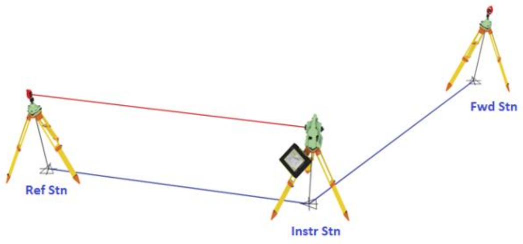

Fig. 4

In the foregoing therefore, a survey network has origin and orientation in line with design or observed topology. The form of these networks is illustrated in Fig 5.

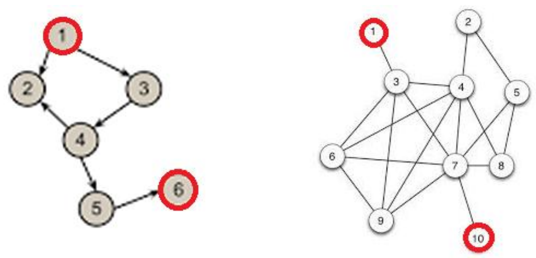

Fig. 5

2. *Detail Survey:*

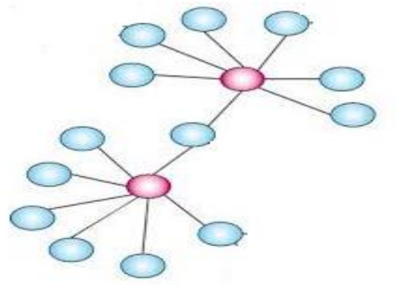

Fig. 6

Detail surveys provideinformation from attributes to detail points that are usually acquired with reference to already established control stations, Fig 6. There are of lower order of accuracy and so are normally considereddistinct from the main network during computations.

B. *Office Administration*

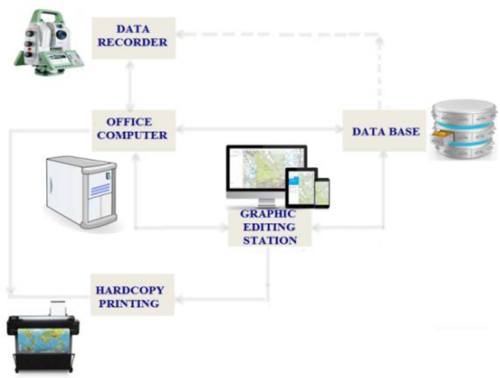

Fig. 7 e-Administration

A model of survey administration is best represented in Fig 7, for any aspiring organization [5]. This is particularly suited for a land records department with responsibility for cadastre.

Going back over the years, there has always been the need to authenticate the work of private and registered surveyors by the office of the Surveyor-General. The problem associated with this verification is long delays as the offices tend to be very overwhelmed. The answer in form of diversifying authorization only gave room to inconsistency in standard. In some localities, the vetting procedures by government agencies have been abandoned altogetherin order to eliminate bottlenecks. The result in many instances is not welcome.

However, it is possible to delegate this function to software assistant to overseeonline submissions and vetting of survey jobs, in a central office, Fig 8.

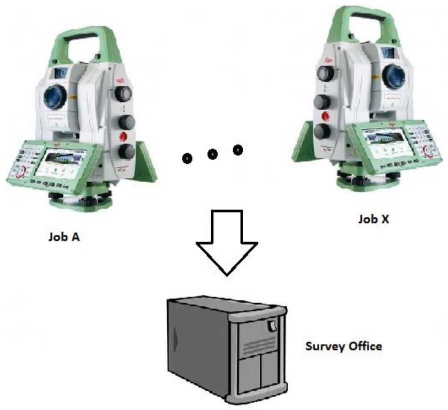

Fig. 8

III. KNOWLEDGE REPRESENTATION

In the discussions so far, it is obvious that diagrams of Fig 3 and Fig 5 are common. While the former is a representation of graph data structure, the latter is the form of a survey network. It can be deduced therefore that trees are perfect model of networks.

A number of analogies are apparent, Fig 9. While a tree is rooted, a network or job has an origin. Similarly, a forest is a group of trees in a locality, and in the same vein networks or jobs located in an area, have datum in common.

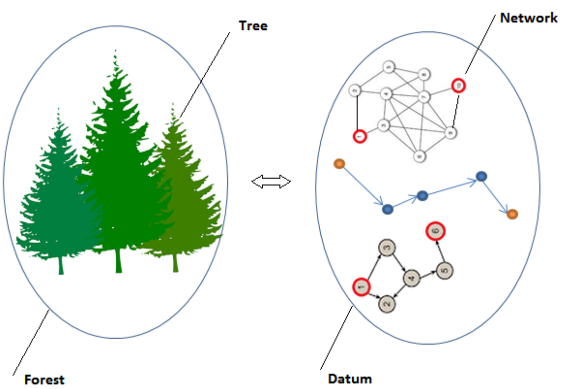

Fig. 9

A. *Objects*

```
PNetWork = ^TNetWork;
TNetWork = object(TObject)
             :
   NodeList, Fixed :PCollection;
   EdgeList: PSortedEdges;
   StnNode:PNode;
             :
   procedure FormNodes;
   procedure LinkNodes;
   procedure Add_ProvCoords;
              :
   procedure SaveData;
end;
```

Fig. 10

The model and implementation of a network object, Fig 10, as discussed in [6] forms the basis of Survey Management System. A follow up in the present discussion is a forest representation, a datum object. This form is illustrated in Fig 11.

```
PDatum = ^TDatum;
TDatum = object(TObject)
                :
        NetWork: PNetWork;
        NetWorks: PCollection;
                :
        constructor Init;
        procedure Process;
                :
        destructor Done;
end;
```

Fig 11

A survey datum is a reference frame on which jobs may be located. There are topographic, cadastral and geodetic datums etc. And so thedatum object, as would be expected, is a container for all the networks in a given datum. While the variable NetWork is an instance of PNetwork type, NetWorks is a list of all such instances in the object.

There are a number of methods for which Init and Process provide vital functions. Init, installs all the variables so defined, fetching information required for processing. The procedure Process handles parallel computations and management.

IV. PARALLEL PROCESSING

Multi-tasking has been standard with desk top operating systems for over a decade. Whats more, the trend in multi core processors meant that multi-programming is now the norm. The benefits are there in online registration and booking activities etc.

It is in this respect that the model can be adopted in this application. There are four key approaches in methodology:

- Multi-threading
- Concurrency
- Memory Efficiency
- Program Architecture

A. *Threads*

A program that is running is known as a process, where it is assigned to a processor within the mechanism of transition state involving scheduling, wait and ready status. A process then can be divided into executing units operating concurrently. Such a unit is a thread and every running program is comprised of at least one, the main thread.

In conclusion, a computer solution can be obtained through combination of multiprocessors,multiprogramming and multithreading as in Fig 12.

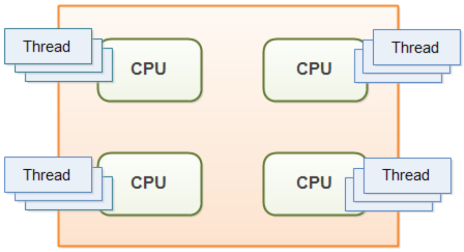

Fig 12

B. Concurrency

One of the common problems in parallel computing is conflict among threads. For instance, in Fig 13 is a situation where two jobs arrive at the server, virtually at the same time, for submission.

Fig. 13

It can also arise when threads write to shared memory, leading to undesirable results. However, it can be avoided if only one job is registered at a time. This can be achieved through a form of synchronisation where entry to registration process, Critical Section, is mutually exclusive.

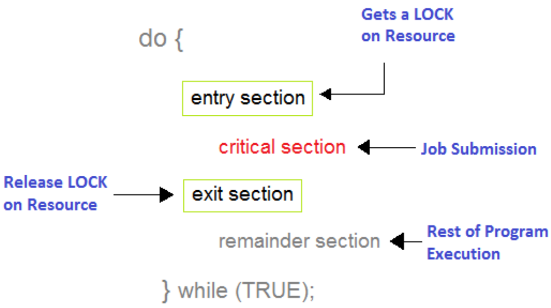

Fig. 14

There are a number of algorithms for implementing Critical Section, but essentially they follow the pattern in Fig 14, and is best achieved through the use of binary semaphores [7, 8].

C. Memory Efficiency

In any client-server environment, it is always safe to assume that processes will be greater than available resources, requiring use of most optimal techniques.This is particularly true of online services. A proven technique is virtual memory as managed by the operating system. However, the developer can further enhance these operations through program segmentation [9,10]. This allows attributes to segments such that only one copy can reside in memeory. By this arrangement thousands of jobs can be submitted and processed comfortably.

D. Program Architecture

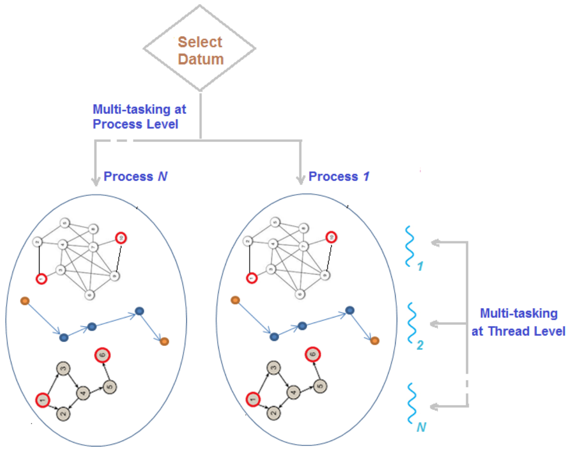

Fig. 15

The architecture for program development is illustrated in Fig 15. It starts with selecting a datum process which translates into a forest program. There may be as many processes required that can run in the mode of multiprogramming.

Then for each process chosen to run, there may be as many threads, tree based units of network computations executing concurrently, in the concept of multithreading.

Thus, Fig 15 is a realisation of the concepts envisioned in Fig 12 for optimum in multi programming.

V. CONCLUSIONS

Applications of this algorithm is invaluable in many instances of engineering involving networks, such as Control Surveys and Land Management.

In each of these cases, it is a question of providing a central quality control administration that can routinely vet vast and diverse results, within a sustainable deadline. This is possible through dynamic allocations for multi tasking, at process and thread levels.

It is accomplished by matching data structures to tasks so that the algorithms can follow naturally, leading to precise and efficient solutions.

REFERENCES

[1] Niklaus Wirth, "Algorithms and Data Structures", © N. Wirth 1985.

[2] Jean-Paul Tremblay, Paul G. Sorenson, "An Introduction to Data Structures With Applications", McGraw Hill Computer Science Series 2nd Edition.

[3] Milton Rosenstein, "Data Structuresfor Programmers" Copyright © 1988 John Wiley & Sons, Inc.

[4] C. P. E. Agbachi, "Parallel Multiple Key Sort Algorithm for Traverse Data Set", IJETT, Volume 56 Number X1 – February 2018.

[5] C. P. E. Agbachi, "A Hands Free Approach to Automated-Cartography: Case Study of Land Records", IJETT, Volume 40 Number 2 - October 2016.

[6] C. P. E. Agbachi, "Algorithm for Dynamic Formation of Condition Equations in N-Dimensional Network", IJMTT, Volume 52 Number 9 - December 2017.

[7] Doug Lea, "Concurrent Programming in Java", State University of New York at Oswego.

[8] C. K. Ayo, "Operating System (OS) Concepts," Department of Computer & Information Sciences Covenant University Ota – Nigeria (NOUN).

[9] Jerry Breecher, "Operating Systems memory Management", http://web.cs.wpi.edu/~cs3013/c07/lectures/Section08-Memory_Management.pdf

[10] Silberschatz, Galvin and Gagne ©2005,Operating System Concepts, Memory Management,http://www2.latech.edu/ ~box/os/ch08.pdf.