International Conference on
"Innovative Research in Agriculture, Engineering, Technology, Applied Sciences,
Humanities and Business Management for Sustainable Development"
(ETAHBS-2019)

Object Oriented Programming based Finite Element Module for STATIC ANALYSIS of Various 2D Structural Elements

Arfat Ahmed

Assistant Professor, Chandigarh College of Engg & Tech. (Degree Wing),
Chandigarh Administration, Chandigarh
E-mail:er_arfat@yahoo.com

Abstract—*In recent decades the three words object-oriented analysis (OOA), object-oriented design (OOD) and object-oriented programming (OOP) have received attention in the design of finite element tools. The essence of object-oriented analysis and design is to emphasize considering a problem domain and logical solution from the perspective of objects. During object-oriented programming the designed components are implemented into computer language. Because the finite element analysis (FEA) consists of parts which can be treated as objects, e.g. matrixes, vectors, meshes and elements, the object-oriented concepts are good candidates for designing finite element tools.*

The Finite Element Method is a widely accepted general purpose numerical modeling tool. Typical FE programs consist of several hundred thousand lines of procedural code and many complex data structures. Approaches, based on object oriented programming (OOP) concepts, are becoming popular as evidenced by the exhaustive bibliography of the use of OOP

Techniques in FEM.OOP are based on the idea of "objects" that encapsulate both the data and the operations on the data. The implementation details are hidden and every object defines itself clear interfaces for communication. This makes code very simple to maintain and modify, and hence attractive for use on research FE codes.

In this paperi am describing aspects of the design and implementation of an OOP based FE code primarily for use in structural frames using 2 node beam element as well as planer and ax-symmetric problems using constant strain triangles. The modular structure in the object oriented code clarifies the communication patterns hence make data decomposition and load balancing easier. This significantly improves scalability of the code.

.Thepresentpaperfocusesonthree main objectives i.e. creating different object oriented framework with respect to available/existing OOP frameworks, developing algorithm for that and finally taking a case study to demonstrate the application of above framework.

Keywords *:OOP,FEM, 2 node beam element,3-node constant strain triangles, polymorphism.*

1. INTRODUCTION

OOP is an art of programming that uses "objects" to design applications and computer programs. It utilizes several techniques that are inheritance, modularity, polymorphism and encapsulation. Even though it was originated in 1960 but became popular in 1990 and today nearly all popular programming languages supports OOP concept.

Object-oriented programming may be seen as a collection of cooperating objects, as opposed to a traditional view in which a program may be seen as a list of instructions to the computer. In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent little machine with a distinct role or responsibility.

Object-oriented programming came into existence because human consciousness, understanding and logic are highly object-oriented. By way of "objectifying" software modules, it is intended to promote greater flexibility and maintainability in programming, and is widely popular in large-scale software engineering. By virtue of strong emphasis on modularity, object oriented code is intended to be simpler to develop and easier to understand later on, lending itself to more direct analysis, coding, and understanding of complex situations and procedures than less modular programming methods.

2. FINITE ELEMENT METHOD AND ELEMENT MATRICES FORMULATION

AnThe Finite Element Method (FEM), also called FEA (for Finite Element Analysis), is actually an approximate mathematical method for solving problems which can be determined by differential equations. Most problems in structural analysis, fluid mechanics, and heat transfer are problems of this class.

Steps to solve the problem using FEM:

- First step is to develop governing PDE and setting Boundary conditions.
- Writing Integral statement i.e. developing the weak form using test function.
- Discretization
- Finding Element arrays and vectors
- Developing Global arrays Using elemental arrays & vectors.
- Applying essential boundary conditions
- Now find unknown vector using master equation :

Kd = F

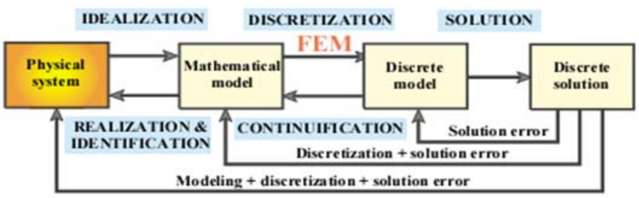
Where K, d, F are matrices.

- In general if N are the DOF that are not restrained then finally :

K is a N*N square matrix

d is a N*1 Vector

F is a N*1 Vector



A simplified view of the physical simulation process, reproduced to illustrate modeling terminology.

Depending upon the nature of problem the dependent variable u is defined as follows:

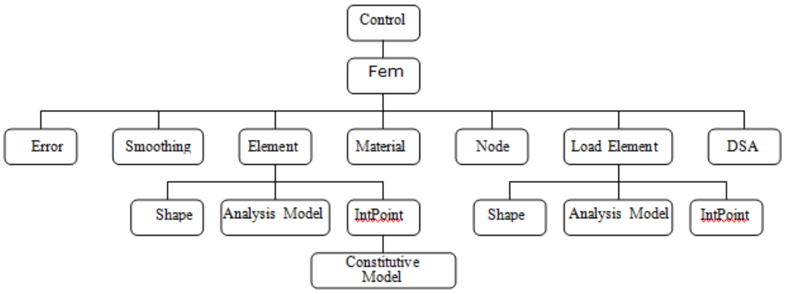
Application Problem	*State (DOF) vector* **u** *represents*	*Conjugate vector* **f** *represents*
Structures and solid mechanics	Displacement	Mechanical force
Heat conduction	Temperature	Heat flux
Acoustic fluid	Displacement potential	Particle velocity
Potential flows	Pressure	Particle velocity
General flows	Velocity	Fluxes
Electrostatics	Electric potential	Charge density
Magnetostatics	Magnetic potential	Magnetic intensity

ISBN-978-93-85822-96-4

Existing OOP Approachand Lirature Review

➢ An Object-Oriented Framework for Finite Element Programming (*Luiz Fernando Martha and EvandroparenteJunior)*

In this approach the important feature is the capability of treating multi-dimension finite element models in the same object oriented, generic fashion. This is accomplished through the definition of two OOP classes: Analysis Model and Shape. The former is responsible for handling the specific aspects related to the differential equation that governs the element behavior, while the latter deals with the geometric and field interpolation aspects of the element. Another interesting feature of FEMOOP is the generic handling of natural boundary conditions. This is implemented through the automatic creation of fictitious elements responsible for translating these boundary conditions into nodal coefficients of the solution matrices and forcing vectors. These elements are instances (objects) of a class called Load Element. At the global level, an- other class (Control) is responsible for the implementation of the algorithm that controls the analysis of the problem, from which a derived class, called Equilibrium Path, handles the different path-following methods implemented in the program.



➢ *Object-Oriented Design of Finite Element Calculations with Respect to Coupled Problems(Wolfgang Mai and Gerhard Henneberger)*

This paper presents a new object-oriented design of software for finite element calculations. Special attention is given to coupled problems with nonlinear materials. Fundamental ideas of object-oriented design, especially high cohesion low coupling and encapsulation of classes are strictly taken into account. The concept is guided by the idea to reuse as many parts as possible. The hierarchy of classes for the elements, materials and field problems can be extended easily by specialization

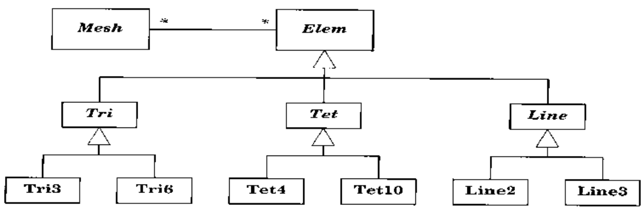


Fig:generalizationoftheelementclasses

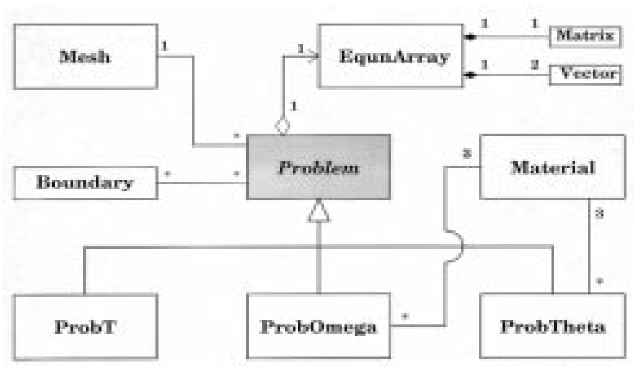


Fig: The class diagram of the problems and their associations

ISBN-978-93-85822-96-4

3. DEVELOPEDALGORITHM AND COMPUTER PROGRAM IMPLEMENTATION

The Algorithm of program is based on Object oriented approach using C++ as its primary language. A class has been chosen with name Element which is inherited number of times publicly depending upon the need; with any inherited class is specializing one basic structural element.

Now base class has two types of functions mainly:

- **Overloaded**

These are basic mathematical functions +,-,* and so on mostly overloaded to provide facility of matrix operations. One more group of function is read file functions that are overloaded and surely constructors depending upon nature of problem.

- **Virtual**

These are algorithm based functions for all sub classes depending upon nature of Element solved using them that is why these are defined virtual in base class.

Now derived class:

- **Constructors**

Constructors of derived classes are used to assign value to derive class specific variables and also if some data is needed to pass to base class as well.

- **Virtual Functions**

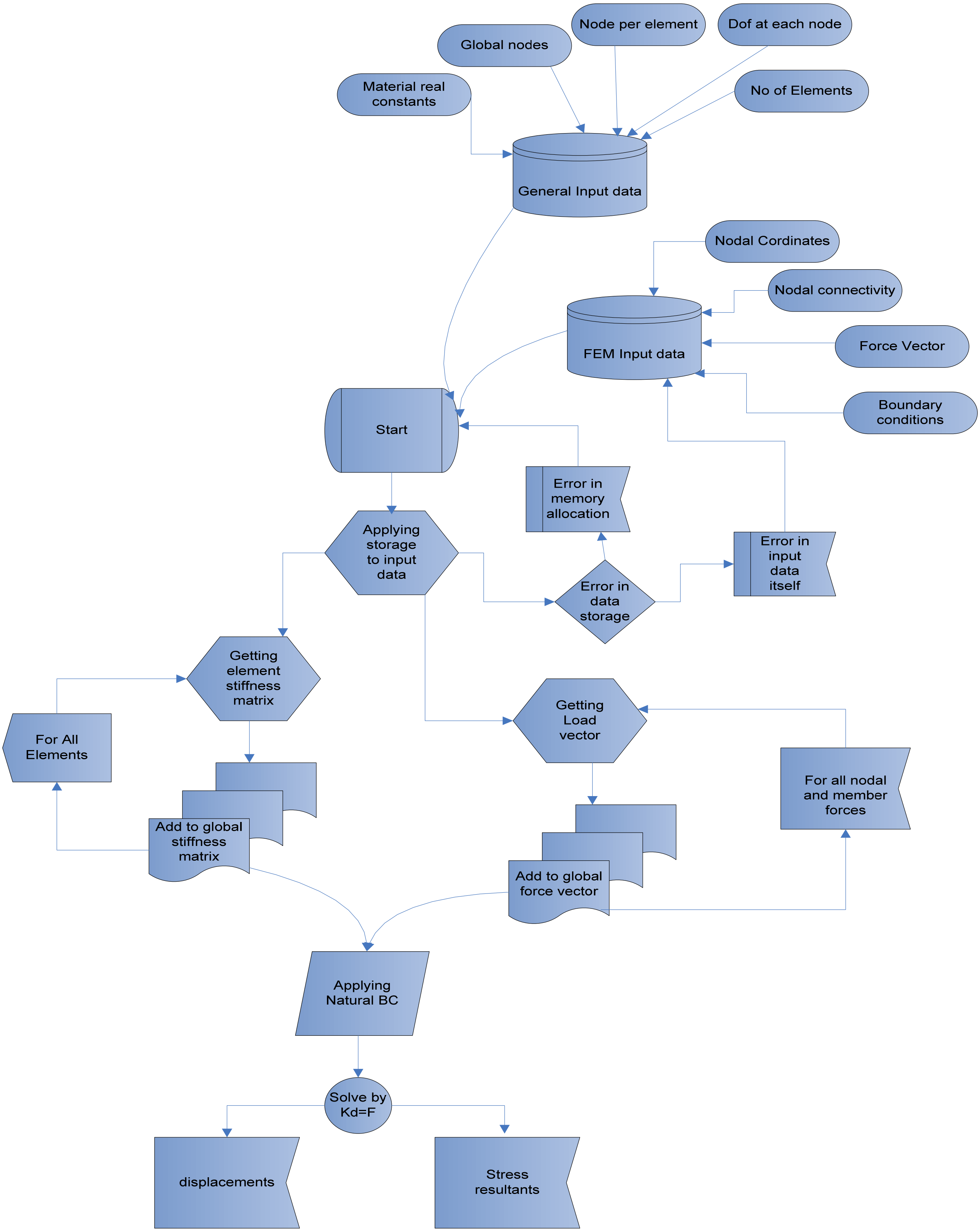
These functions are basic logic of whole program dealing with Nodal coordinates, connectivity and Force and boundary vector to find out displacements and Stresses using all available data specific to a particular element.

The virtual function ***Operations*** in all the derived class is using step by step method of Finite Element analysis to calculate displacements and stresses using all other virtual functions during its runtime.

Well as far as object oriented specialties are used Encapsulation is achieved in Base class using protected variables, due to which derived classes are to be made friend of their own base class.

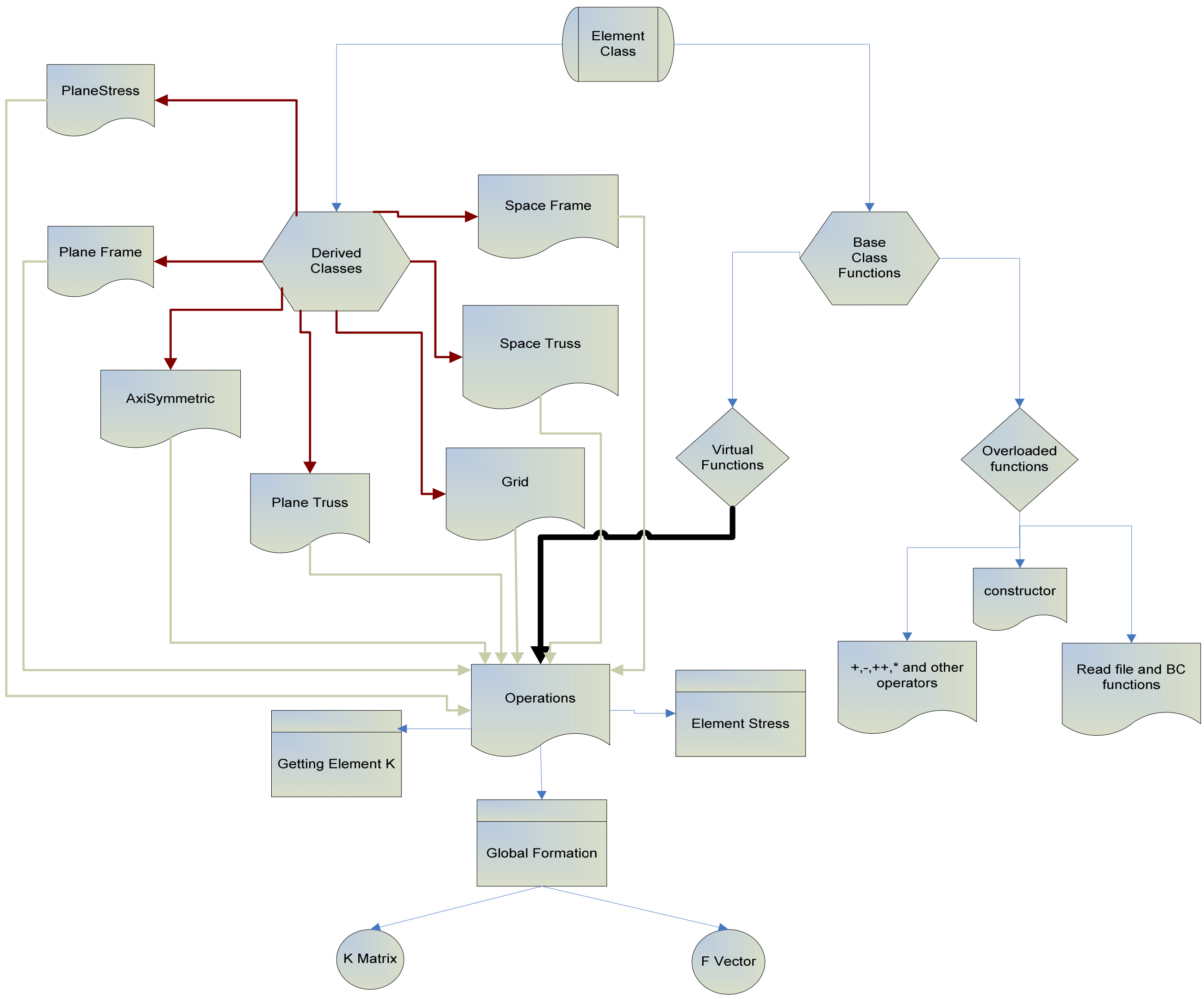
Inheritance is already used for inheriting classes which is very useful for specific element analysis. Polymorphism is used in both forms, compile time polymorphism as overloaded functions and operators as well as runtime polymorphism using base class pointer to be assigned to inherited classes which only calculates which member function to be called for analysis (during runtime only).

ISBN-978-93-85822-96-4



ISBN-978-93-85822-96-4

Developed Flow chart of Basic FEM steps



Indicating virtual functions

Indicating Derived Classes

Indicating virtual functions that are defined in Derived classes

Indicating all attached to itis part of parent one.

Class diagram with its functions

ISBN-978-93-85822-96-4

4. CASE STUDY: *PLANE FRAME*

Problem:

Determine displacements and rotations of Plane Frame.

Data:

Nodal coordinates:

nodes	coordinates
1	(0, 96)
2	(144, 96)
3	(0, 0)
4	(144, 0)

Nodal Connectivity

Member	Connectivity of nodes
1	1-2
2	3-1
3	4-2

Force vector:

3000 lb force on Node 1 and uniformly distributed
load of 500 lb/ft on member 1.

Boundary conditions:

Node 3 and 4 are fixed.

E=30e6 psi; I = 65 $in^{4;}$ and Xn Area = $6.8in^2$

Solution:

Result:

-------------FIRST ELEMENT----------------

Local Node	Corresponding global node	Nodal deflections
1	1	0.0917665
2	2	-0.00103585
3	3	-0.00138737
4	4	0.0901188
5	5	-0.00178768
6	6	-3.88301e-05

Member forces......

2334.22

2201.18

-3776.63

-2334.22

3798.82

-111254

ISBN-978-93-85822-96-4

-------------SECOND ELEMENT----------------

Local Node	Corresponding global node	Nodal deflections
1	7	0
2	8	0
3	9	0
4	1	0.0917665
5	2	-0.00103585
6	3	-0.00138737

Member forces......

2201.18

665.783

60138.5

-2201.18

-665.783

3776.63

-------------THIRD ELEMENT----------------

Local Node	Corresponding global node	Nodal deflections
1	10	0
2	11	0
3	12	0
4	4	0.0901188
5	5	-0.00178768
6	6	-3.88301e-05

Member forces......

3798.82

2334.22

112831

-3798.82

-2334.22

111254

Deflections Node by Node in Global system

0.0917665

-0.00103585

-0.00138737

0.0901188

-0.00178768

-3.88301e-05

0

0

0

ISBN-978-93-85822-96-4

0

0

0

Above results are absolutely matching the results obtained by **ANSYS**Software results.

5. CONCLUSION

FEM is the heart of any Design Software and Object oriented programming is an established norm in software development. Combining these two in an efficient manner is a boon for design software industry.

In this project we have taken a different approach to the existing general OOP approach of using combination of different classes where each separate class is responsible of dealing with general inputs starting from Nodal coordinates to material. Our approach is to define a base class consisting of general input data needed for all type of problems and then inheriting the classes depending upon the specific type of problem.

Well the benefit of this approach is apparent in smaller problems as you don't have to deal with so many different classes at a time so chances of errors are less and program flow is simple.

As far as complexity is concerned;as number of specific domain problems will increase, algorithm complexity will be a bit tricky to handle but complexity will definitely be high if different elements/sections will are of different material.

Although it looks easy to work initially but real advantages or disadvantages will come in to picture when isoperimetric elements will come in to picture and material specific algorithm will be introduced in the system.

As far as results of plane frame results are to be seen they are perfectly sameas design software ANSYS.

REFERENCES

1. *Professor M Venkateshwarlu*, IIIT Hyderabad,"FEM class Notes".

2. *G.R LIU and S.S QUEK* , 2nd Edition ,"The Finite Element Method".

3. *T. R. CHANDRUPATLA and Ashok D Belegundu, 3rd* Edition ,"Introduction to Finite Elements in Engineering", PEARSON EDUCATION.

4. *Luiz Fernando Martha and Evandroparante junior* ,Research Paper: An Object Oriented Framework for Finite Element Programming .

5. *Wolfgang Mai,GerhardHenneberger*, Research Paper: Object Oriented Design Of Finite Element Calculations with respect to Coupled problems.

6. *Bjarne Stroustrup, 3rd* Edition," The C++ Programming Language*",* PEARSON EDUCATION.

7. *Robert D Cook, David S Malkus and Michael E plesha*,4*th* Edition ,"Concepts and Applications of FEA" ,JOHN WILEY & SONS.