In this chapter,

4.1 Preamble
4.2 System Knowledge
   4.2.1 System Knowledge Representation Techniques
4.3 Data Flow Diagram
   4.3.1 Data Flow Diagram as System Knowledge Map
4.4 Organizational Processes
   4.4.1 Process Knowledge Map
4.5 Structure Chart
4.6 DFD and Structure Chart as Process Knowledge Map
4.7 Summary
Understanding system is always been prerequisite for any development work. Without understanding domain completely, it is not possible for developers to prepare the system. Requirement gathering always starts with capturing domain knowledge. Systems analyst, adapting Structured Systems Analysis & Designing (SSAD) approach, uses various standard tools like Data Flow Diagram (DFD) for capturing domain knowledge. DFDs are significant tool for representing data sources and their flow from environment to system and within the system. Also, they represent the processes converting data to useful information. Apart from using for software development, DFDs can be used easily for representing business system/business domain knowledge.

Presently, DFDs play major role in communication process between organizations and developing team, however their utilization in communication process for with in a system has not been explored. In this chapter, an effort has been made to establish the use of DFD as a system knowledge map. This chapter establishes how DFDs can be used for representing system knowledge and can be considered as useful knowledge map for sharing the system knowledge. They can be proved as simple but powerful tool for facilitating sustainable knowledge about the system.

Organization performs through various processes. We have already discussed about procedural knowledge in process layer in Chapter 2. Main feature of DFDs is to represent the process; however, limitations of DFDs in showing level of details of process can be overcome by using Structure Chart. Structure Chart is another tool used in SSAD for designing the modules. In this chapter, we show that Structure Chart can be used as very comprehensive process (procedural) knowledge map.
4.2 System Knowledge

Knowing the system is the first step to understanding the system. Without experience, it is very difficult to understand the system completely. One has to involve himself to understand the system.

System is normally considered as collection of sub-systems and all these sub-systems work together collectively to achieve specific objective of the system. Government system, education system, police system, a University, a human body and a computing machine are some examples of such a systems.

All organizations are good example of system. Organizations can be considered as system, effectively utilizing men, material, machines and other resources to achieve pre-determined goals and objectives. Sub-systems in the organization carry out the specific function and called as Department.

Understanding the system and its functionality, thoroughly, is a challenging task for the new employees in the organization. Concept of treating employees as resource is now very well established. Hence, training and development of employees is very crucial element. In fact, we have experienced that, acquiring sophisticated and latest equipments, machines and devices are much easier than to have skilled employees to work on them. This problem increases manifold, when an experienced employee leaves the organization and replacement is not found easily.

Today’s competitive environment, forces organizations, to increase their learning capacity [Kabene, 2006]. With time and due to ICTs, learning processes have improved significantly, however, organizations lacks in formal mechanism for learning from past experience and preservation of current knowledge [Kabene, 2006].

Process of knowledge input and output, knowledge sharing and transfer, within the organization is considered as organizational learning. Investment on organizational learning is strategic investment process, that continues throughout the life of organization. Yunhe [Yunhe, 2008] analyzed cost
components of learning investment. We argue that cost can be reduced substantially by incorporating formal KM system, as shown in Fig.4.1(a) & Fig.4.1(b), below. However, percentage of cost reduction owing to use of KM system is a matter of investigation and is presently not in the scope of discussion.

Fig.4.1(a): Cost Components of Learning Investment [Yunhe, 2008]

Fig.4.1(b): Possible Reduction in Cost Components of Learning Investment after Deploying KM System

K-A Means Knowledge Artifacts that include Knowledge Maps, Knowledge Bases, Repositories etc.
System knowledge or business domain knowledge are pre-requisite to start the work, to understand the processes. Considering about example of system knowledge, system knowledge can be knowledge of the organization as whole (business domain knowledge), can be of a particular department, about any manufacturing facility or about a particular machine. This depends on the boundary of the system, i.e. level of abstraction, which is decided by the knowledge need and knowledge gap analysis. There is a great need to have simple, standardize, reusable, reproducible and effective knowledge map for representing system knowledge.

4.2.1 System Knowledge Representation Techniques

In this section, we have discussed, in short, some techniques being used for communicating system knowledge. Techniques discussed here are: Textual, Pictorial and Diagrammatic:

- Textual Technique- This is simple and widely used technique for communicating ideas, hence, system can also be described through text. Correct interpretation, reproduction, standardization and sharing are the critical issues associated with textual representation. Size depends on context and limited by expression capability. Normally textual representation does not considered as knowledge map.

- Pictorial Technique- Representing system through pictures is very easy and communicating technique. Interpretation is not the issue with this technique. However, standardization and sharing through automated system are bottleneck for their effectiveness. Pictorial techniques can be considered as knowledge map.

- Diagrammatic Technique- System representation by collection of various shapes, lines and text. This is semi-formal technique. Some approaches of this category are formalized and widely used like UML, ERD, DFDs etc. However, they are not being used extensively for system knowledge representation. These techniques are good example of knowledge map.
We have identified some characteristics that a knowledge map must have to represent knowledge, whether system, process or any other knowledge. The knowledge map:

- Must be abstract i.e. it must be precise, crisp and clear
- Must be shared and understood easily i.e. it must have commonly understood symbols and shapes.
- Must be repeatable i.e. it must have some systematic approach to develop.
- Must be easy to modify and maintainable i.e. it must have simple and robust approach to develop, modify and update.
- Must have semantics i.e. it must be able to reflect proper meaning.
- Must have ease of access i.e. it must be available through systematic approach.
- Must be validated i.e. its correctness must be endorsed.
- Can be produced through reverse engineering i.e. it can be automated and be produced through software system.

In the next section, we have established DFD as a knowledge map for system knowledge.

### 4.3 Data Flow Diagram

Data Flow Diagram (DFD) is a very useful system modeling tool, which pictures a system as network of functional processes [Yun, 2009]. As the name suggests, DFD is a graphical representation of the flow of data between business functions or processes and how the processes transforms such data [Ferri, 2006].

DFD is function oriented and mainly used for visualization of processes of the organization in SSAD for understating system requirements. With the use of symbols, which are having definite meaning/understanding, these diagrams are capable to communicate the requirements of the desired software system. DFD is a standard tool in SSAD for requirement gathering.
DFD has mainly four graphical components [Ferri, 2006]:

- Processes
- Data Flow
- Control Flow
- External Entities
- Data Stores

DFDs are very well understood and established concept in SSAD, hence emphasis in this chapter is on the use of DFDs as knowledge map.

### 4.3.1 Data Flow Diagram as System Knowledge Map

We interpret DFD as knowledge map due to several reasons. First of all, they fit into criteria for knowledge map as discussed in the previous section. DFDs are standardized and hence numerous tools available to produce them. They can be shared and modified easily along with management of older versions. The most important point is that they can be produced through reverse engineering also [Butler, 1995] so as to map organizational knowledge stored in knowledge base. Our perception of DFDs from KM perspective and one-to-one mapping form software engineering concepts is described in Table 4.1.

<table>
<thead>
<tr>
<th>Software Engineering Perspective</th>
<th>KM Perspective</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Analysis Tool</td>
<td>Knowledge Map</td>
<td>Our perception</td>
</tr>
<tr>
<td>Context Diagram (Level 0 DFD)</td>
<td>System Knowledge Map</td>
<td>Use of Context Diagram for representing system knowledge</td>
</tr>
<tr>
<td>Data Flow Diagram</td>
<td>Process Knowledge Map</td>
<td>DFD can be used to codify process knowledge</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>DFD Components</td>
<td>Knowledge Artifacts</td>
<td>What we will call here</td>
</tr>
<tr>
<td>Data Flow</td>
<td>Input/ output</td>
<td>Data Flow are seen as input to and output from any process</td>
</tr>
<tr>
<td>Control Flow</td>
<td>Stimuli to trigger process</td>
<td>The signal that will trigger the execution of process</td>
</tr>
<tr>
<td>Data Store</td>
<td>File Cabinets, Physical Storage of Files, Documents, Rule Books, Reports</td>
<td>Organizational Data, Information &amp; Knowledge stored in various formats</td>
</tr>
<tr>
<td>Entities</td>
<td>Stake Holders, Departments</td>
<td>Participants to the process</td>
</tr>
<tr>
<td>Processes</td>
<td>Set of activities triggered by input and then gives output, functions of department</td>
<td>What organization performs through entities</td>
</tr>
</tbody>
</table>
We have shown two cases here, representing use of Context Diagram (Level 0 DFD) as system knowledge map. The next section presents organizational processes.

**Case Study– 01**

In this case, we have shown that how Context Diagram can be used as system knowledge map. Instead of describing whole system in this case we have shown a sub-system of the organization i.e. a particular department.

**Textual Description of the system** - A manufacturing company has a purchase department. This department procures items for manufacturing, maintenance, office and other departments of the organization. The department that needs item prepares the list and sends requests to purchase department. After getting request for items, purchase department checks for the availability of items in stores. If an item is not available in stores, and the department requesting for item has sufficient budget, then only purchase department process the request for item. The finance department fixes the budget for every department at starting of every financial year.

Suppose an item has to purchase, then purchase department sends request for quotation to registered vendors. Once all quotations received, analyzed and comparative statement prepared, selecting a vendor, a purchase proposal is prepared. This purchase proposal is sent to finance department for financial concurrence. After approval from finance department, purchase order is sent to vendor. Vendor delivers items to stores department, and sends an invoice to purchase department. After verification and confirmation, stores department sends acceptance report to purchase department. After getting acceptance report from stores department, purchase department, passes the invoice and forward to finance department for payment. In case, if material is not found satisfactory, purchase department rejects the invoice.

Textual description has always some issues like interpretation, reproduction and sharing capabilities. However, this is not the case with diagrammatic form
of knowledge representation. Below here the Context Diagram (Fig.4.2), that simply describes what has been described above in textual form.

It is worth mentioning that it is assumed that the user of this system knowledge map understands basic functioning of departments. However, relationship between departments is being represented here.

Fig.4.2: Context Diagram Representing System Knowledge (A Department)
Case Study – 02

This case shows, how the industrial systems can be represented through context diagram, which again turns out to be simple and easy to understand and sharable. In this case, we have taken Vapour Compression Refrigeration System [Nag, 2008] as an example.

Textual Description of the System - The Vapour Compression Refrigeration System (VCR System), widely used for commercial and domestic refrigeration purpose, use a circulating liquid as a refrigerant.

A VCR system has four components: a compressor, a condenser, an expansion valve and an evaporator. Circulating refrigerant enters the compressor as a vapor and is compressed to a higher pressure, resulting in a higher temperature as well. The hot, compressed vapor is then routed through a condenser where it is cooled and condensed into a liquid by some heat removal medium, may be water or the air.

The condensed liquid refrigerant is next routed through an expansion valve where it undergoes an abrupt reduction in pressure. That pressure reduction results in adiabatic flash evaporation that lowers the temperature of the liquid and vapor refrigerant mixture.

The cold mixture is then routed through the evaporator. The evaporator is where the circulating refrigerant absorbs and removes heat which is subsequently rejected in the condenser and transferred elsewhere by the water or air used in the condenser.

VCR system operates on the inputs from temperature sensor, which actually results in electrical signals to compressor motor to start or off.

Figure 4.3 shows the context diagram of VCR System. The context diagram represents VCR system at highest level of abstraction. All components and user of the system are represented as entities.
Organizational Processes

Organizations as a system, perform through processes. Interaction within components of system i.e. sub-systems, called departments and interaction of system with environment i.e. with customers, suppliers, service providers, government agencies etc called transaction. Organizational processes define this transaction. Outcome of processes produces a specific service, product or serves a specific objective.
Hammer & Champy [Hammer, 1993] defined process as – “a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer”. However, several processes are there in an organization that directly do not interact with customer, but are necessary for making business smooth.

American Productivity & Quality Center (APQC) has given Process Classification Framework (PCF) [APQC, 2009], which categorizes Business Processes in two categories– Operating Processes and Management & Support Processes. Another categorization for processes is possible i.e. Departmental Process (within a department) and Cross-functional Process (across two or more departments). Figure 4.4 shows process types based on process span.

Every process has some parameters associated like –

- When process shall be executed (stimulus)? Like, *deposit yearly premium when due date comes, calculate quarterly interests* etc.
- What will be the inputs? Mainly in the form of Data, like *account number, amount* and *due date*.
- Who will initiate the process? *Name of the department and responsible person* (by position or in certain cases by name) like *Purchase Manager in Purchase Department.*
• What steps are required to complete the process (atomic activities)? This is detailing of process, which mainly include workflow.

• When the process shall be end? When all steps are carried out as defined in the process or when some unexpected happens like insufficient balance, students is not registered etc.

• What will be the outputs? Outcome of the process that is mainly in data form, like Name of Students to be detained etc.

All these parameters are actually knowledge artifacts needed to execute a particular process.

Operational processes, which are carried out in the organizations on day-to-day basis, needs relevant information and knowledge, which can be easily, mapped through process knowledge maps, which are discussed next.

4.4.1 Process Knowledge Map

Organizational processes are one of the three important components of an organization from KM perspective; other two are people and technology. To know - how a particular process has to carry out, when and who will initiate the process, all needs experience and previous knowledge. As people in the organization execute processes, a formal mechanism to understand the mechanism of process becomes more important.

A survey conducted by The Economist Intelligence Unit in 2005 [The Economist, 2005] shows main obstacles in achieving an efficient flow and use of knowledge within organization (Table 4.2). Internal barriers to sharing, inadequate understanding of existing knowledge pieces and inadequate training are the some issues [The Economist, 2005], which are prominent, and process knowledge maps may help in resolving these issues.

Process knowledge maps are required to codify know-how to do/ what to do of processes. Developing and using process knowledge maps have several advantages apart from inherited advantages of KM. When process knowledge maps based on existing processes used, they may give an opportunity to
improve or optimize the existing process. Also, they can be very useful tool for Business Process Re-engineering (BPR).

It has been practice to represent processes through flow charts and Event Process Chain (EPC) Diagram [Ferdian, 2001]. We have illustrated, through example, matrix type knowledge map in the previous chapter for describing knowledge map development model. In this chapter, we are focusing on using DFD and Structure Chart as process knowledge map.

It is worth mentioning that flow charts are basically sequence of activities and they focus on what activities must be completed before start of current activity [Xu, 2009]. However, DFD and Structure Chart focus on what data/information is needed from other activities before completing the existing one [Xu, 2009]. This changed perspective makes DFD and Structure Chart useful knowledge map.

**Table 4.2: A survey from The Economist Intelligence Unit 2005**

[The Economist, 2005]

<table>
<thead>
<tr>
<th>What are the main obstacles to achieving an efficient flow and use of knowledge within your organization? (Select up to two options)</th>
<th>% Respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal barriers to the cross-departmental sharing of information and knowledge</strong></td>
<td>50</td>
</tr>
<tr>
<td><strong>Inadequate understanding of the information and knowledge that already exist in the organization</strong></td>
<td>41</td>
</tr>
<tr>
<td>Inadequacy of IT system</td>
<td>38</td>
</tr>
<tr>
<td>Inadequate understanding of the types of information and knowledge that IT is capable of generating</td>
<td>33</td>
</tr>
<tr>
<td><strong>Inadequate training of staff in how to use information</strong></td>
<td>15</td>
</tr>
<tr>
<td>Customer resistance of staff in how to use information</td>
<td>07</td>
</tr>
<tr>
<td>Supplier / Partner resistance to greater information sharing</td>
<td>04</td>
</tr>
<tr>
<td>Other</td>
<td>06</td>
</tr>
</tbody>
</table>
4.5  **Structure Chart**

Structure Chart is an important tool for showing relationship between the modules of a computer program in SSAD [Page-Jones, 1988]. Each module that is the basic component of Structure Chart represents a function. Being hierarchal in nature, higher level modules are control module and lower level modules are executive modules [Page-Jones, 1988]. Lower level modules are called by control module as per the situation may be. This representation is done by arrow connecting the modules. Also, data that is being passed between modules, showing input and output, is represented by little arrows with open circle called data couple [Page-Jones, 1988]. Similarly, control flow between modules is represented by filled circle arrow [Page-Jones, 1988].

Structure Chart differ flowchart in a way that flow charts does not represent data interchange among different modules. In contrast to flow chart, Structure Charts does not represent sequence of execution. However, this problem can be solved by following the rule of “*reading from left to right*” [Page-Jones, 1988].

In SSAD, Structure Charts are prepared by converting them from DFD. Transaction Analysis and Transform Analysis [Page-Jones, 1988] are the two strategies to converting DFDs in Structure Charts.

Basic symbols used in Structure Charts are [Page-Jones, 1988]:

- **Module**
- **Module Invocation Arrow**
- **Data Flow Arrow**
- **Control Flow Arrow**
Figure 4.5 shows an example of Structure Chart showing modules of a banking software system. Some of the operations performed by a bank are like Saving Account Operation, Current Account Operation, and Approve Loan & Maintaining Loan Account etc. In Fig.4.5, we show modules for performing Saving Operations. Main module chooses the operation, which is to perform on the basis of input parameters from the user. This shows flow of control and act as a trigger to particular operation.

Fig.4.5: An example of Structure Chart showing Modules of a Software System
4.6 DFD and Structure Chart as Process Knowledge Map

We have shown the use of DFD (Context Diagram) as a System Knowledge Map in one of the previous section. In this section we have shown the use of DFD and Structure Chart as a Process Knowledge Map.

We argue that search of a comprehensive but simple, easily shareable and above all easy to automate process knowledge map, can be end with the combined use of DFD and Structure Chart.

In Table 4.1, we have shown the view of DFDs from knowledge map perspective. In Table 4.3, we have done similar exercise with Structure Charts.

It is worth-mentioning here that though Structure Charts are being used by the designers for designing the software module, we argue that they represent “how to do?” of process knowledge. DFDs map the process structure by representing main and sub processes. However, Structure Charts help in showing how to perform the specific task or activity. Hence combined use of DFD and Structure Chart becomes a useful process knowledge map. We have shown in the examples below, that Structure Chart can be used for capturing best practices in the organization.

We have explained the use of DFD and Structure Chart as process knowledge map with the help of example in coming pages.

Table 4.3: Structure Chart as Process Knowledge Map

<table>
<thead>
<tr>
<th>Software Engineering Perspective</th>
<th>KM Perspective</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designing Tool</td>
<td>Knowledge Map</td>
<td>Shows our perception</td>
</tr>
<tr>
<td>Structure Chart</td>
<td>Process Knowledge Map</td>
<td>Used in SSAD for representing modules, however, we shall be using to codify Process</td>
</tr>
<tr>
<td>Structure Chart Components</td>
<td>Knowledge Artifacts</td>
<td>Shows our perception</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Module</td>
<td>Processes</td>
<td>Each module shall be considered as Process or Sub Process</td>
</tr>
<tr>
<td>Boss Module (Calling)</td>
<td>Main Process</td>
<td></td>
</tr>
<tr>
<td>Executive Module (Called)</td>
<td>Sub Process</td>
<td></td>
</tr>
<tr>
<td>Module Connecting Arrow</td>
<td>Connectivity of Main and its Sub Processes</td>
<td>Show hierarchical structure and indicates which process has sub processes</td>
</tr>
<tr>
<td>Data Flow Arrow</td>
<td>Stimulus to Activity, Input to Activity, Output from Activity, Process Parameter</td>
<td>Along with direction of flow it shows input to and output from the process</td>
</tr>
<tr>
<td>Control Flow Arrow</td>
<td>Flow of Control between the Modules, Process Parameter</td>
<td>Triggering of process or choice parameter, like Valid or Invalid, Yes or No, Option one etc.</td>
</tr>
<tr>
<td>Library Module</td>
<td>File Cabinets, Physical Storage of Files, Documents, Rule Books, Reports</td>
<td>Organizational Data, Information &amp; Knowledge stored in various formats</td>
</tr>
<tr>
<td>Selection</td>
<td>Decision Making Conditions of Activity/Process, Process Parameter</td>
<td>Representing possible options for executing next process</td>
</tr>
<tr>
<td>Repetition</td>
<td>Process Parameter</td>
<td>To show Same process shall be executed for every transaction</td>
</tr>
</tbody>
</table>
Case Study – 01

Domain- Similar to domain taken in previous chapter, we have taken a case of a department of a University in India. This University happens to be oldest University of Central Region, having around 35 integral departments called University Teaching Departments (UTDs). In addition to these more than 150 colleges, imparting education in area of Arts, Science, and Management etc. are affiliated to this University. Some UTDs do not get any grant/ aid from the government and hence totally depends on own sources of income. We have selected one of the self-financed UTD for our study, which is imparting higher education in engineering & technology.

Process- Education system is typical in a sense that there is no physical outcome as a product from the system. Though Certificate/Degree can be treated as outcome from the system, but actual outcome of this system is value addition and gain of knowledge. Candidates earns knowledge about the subject, by attending lectures, performing experiments, reading books, solving assignments, which all are supported by administrative processes in the framework of system.

In Universities, academic processes like delivering lecture or conduction of experiments normally highly unstructured and actually governed by individual knowledge. Administrative processes are sometimes properly defined and repeatable in same manner, and sometimes some processes may evolve out of the wisdom of people, but executed with in the framework of university and government rules. In this case, we have used DFD and Structure Chart for capturing one of the administrative process -

“How to handle situation, when student(s) have short of attendance in a particular theory or practical subject?”

As per University rule, if a student is having attendance less than 75% in any of the theory or practical subject, he or she shall not be allowed to appear in the exam. According to this rule detention process is carried out.
Now, this process does not handle the cases wisdom fully, so that students can be warned or their parents are informed before end of semester, so as to avoid situation of detention. As this process is not defined in rulebook of University, hence need to capture & codified in some way. We have used DFD and Structure Chart to represent this process knowledge, as shown in Fig.4.6(a) & Fig.4.6(b) respectively.

Fig.4.6(a): DFD as a Process Knowledge Map for Short of Attendance Process

On this basis of domain knowledge, process details and DFD prepared as above we have prepared Structure Chart as Fig.4.6(b). It is worth mentioning that being hierarchical in nature, the same as organizations have, Structure Chart also be used to represent level of execution of the process i.e. showing Performed By. We have used off-page connectors due to space constraint for the last activity.
Fig. 4.6(b): Structure Chart as a Process Knowledge Map for Short of Attendance Process
Case Study– 02

Domain- Same as that of previous example (Case Study- 01).

Process- In previous chapter, we have discussed knowledge map development phases with an example of one small intra-department process, i.e. “Preparing the Panel of Practical Examiners”, carried out by examination department (Intra-departmental). Here, we have prepared Structure Chart based process map for the same process (Fig.4.7).

![Diagram of Preparing the Panel of Examiner Process]

**Fig.4.7: Structure Chart as a Process Knowledge Map for Preparing Panel of Examiner Process**
4.7 Summary

We have developed a model for knowledge map development, B-C Model, and discussed in detail in previous chapter. The Phase-IV of model suggests selection of suitable knowledge map type and Phase-V shows development of map. In this chapter, we have shown the use of DFD and Structure Chart as system and process knowledge maps.

A formal semantic can be developed through Calculus of Communicating Systems [Butler, 1995] for DFD and Structure Chart. With the use of automated tools [Butler, 1995] verification of DFD and Structure Chart can be done, as they can be converted to a piece of code through reverse engineering. This concept can be used for making a product for automation of B-C model, which suggest development of knowledge maps.

DFD and Structure Chart have the following benefits, inherited from the characteristic as SSAD tools, as a knowledge map:

- Level of abstraction can be defined easily
- Sequence of process execution can be represented easily
- Hierarchical nature of Structure Chart indicates level of activity execution
- Data Flow represents input and output to the activity centre
- Control Flow is the stimulus signals to the activity
- Decision making knowledge structure can be shown easily
- Knowledge bases in the form of organizational documents, files etc are the part of knowledge map
- Easy to be developed and verified with reverse engineering tools

In this chapter, major contributories are:

- Process knowledge maps are important artifacts for KM.
- Context Diagram is a simple and semantic-full type of system knowledge map.
- Structure Chart is a useful process knowledge map, which codifies organizational process knowledge in hierarchical manner.
• An automated system can be developed on the concept of reverse engineering for validation and comparison of various Structure Charts.

In the next chapter, a new knowledge map for codifying dynamic knowledge has been devised.