Unit: II
Lecture: 7
Mapping Requirements into Software Architecture
Transform and Transaction Mapping

There are two types of information flow which are the drivers for the mapping approach:

a) Transform Flow.
b) Transaction Flow.

Transform Flow:

- Information must enter and exit software in an "external world" form. External data (like data typed on keyboard, tones on a telephone line and video images in multimedia application) must be converted into an internal form for processing.
- Information enters the system along paths that transform external data into an internal form.
- These paths are identified as incoming flow. At the kernel of the software, a transition occurs. Incoming data are passed through a 'transform center' and begin to move along paths that now lead "out" of the software. Data moving along these paths are called outgoing flow.
- The overall flow of data occurs in a sequential manner and follows straight line paths.
- When a segment of a DFD exhibits these characteristics, 'transform flow' is present.

Transaction Flow:

- The fundamental system model implies transaction flow; therefore, it is possible to characterize all data flow by a single data item, called a transaction that triggers other data flow along one of the many paths. So, when DFD takes the form as shown below, transaction flow is present.
- Transaction flow is characterized by data moving along an incoming path that converts external world information into a transaction. The transaction is evaluated and based on its value; flow along one of many action paths is initiated. "The hub of information flow from which many actions path originate is called a transaction center"

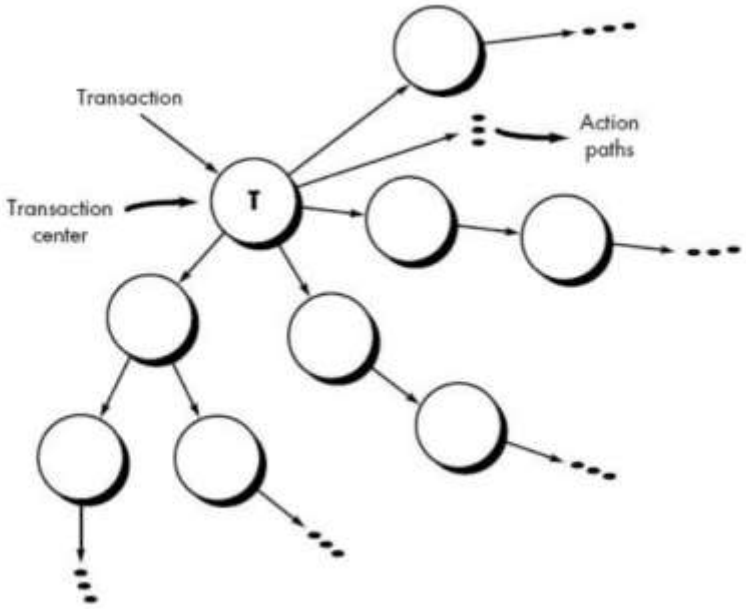

Fig Transaction Flow

Structure Chart: represents hierarchical structure of modules. It breaks down the entire system into lowest functional modules, describe functions and sub-functions of each module of a system to a greater detail. Structure Chart partitions the system into black boxes (functionality of the system is known to the users but inner details are unknown). Inputs are given to the black boxes and appropriate outputs are generated.

Transform Mapping:
Transform mapping is a set of design steps that allow a DFD with transform flow characteristics to be mapped into a specific architectural style.
Structural design provides two strategies to guide transformation of a DFD into a structure chart:

- Transform Analysis
- Transaction Analysis

Transform Analysis (Transaction Mapping): It identifies the primary functional components (modules) and the high level input and output of these components.

First step in transform analysis is to divide the DFD into three parts:

a) Input
b)Logical Processing
c) Output

- The input portion in the DFD includes processes that transform input data from physical (e.g., character from terminal) to logical form (e.g., internal tables, lists etc). Each input portion is called 'afferent branch'. There may be more than one afferent branch in a DFD.
- The output portion of a DFD transforms output data from logical to physical form. Each output portion is called on 'efferent branch'. The remaining portion of a DFD is called central transform.

In second step of transform analysis, the structure chart is derived by drawing one functional component for each central transform, for each afferent and efferent branch. The first level of structure chart is produced by representing each input and output unit as boxes and each transform as a single box.

In the third step of transform analysis, the structure chart is refined by adding sub– functions required by each of the high level functional components. Many levels of functional components may be added. This process of breaking functional components into subcomponents is called 'factoring'.

Transaction analysis (Transaction mapping): It is an alternative to transform analysis and is useful while designing transaction processing programs. A transform centered system is characterized by similar processing steps for each data item processed by input, process and output systems. In a transaction driven system, one of several possible paths through the DFD is traversed depending upon the input data item.

A transaction is any element of data that triggers an action. Every transaction carries a tag identifying its type. Transaction analysis uses this tag to divide the system into transaction modules and a transaction center module. This is an architectural design.