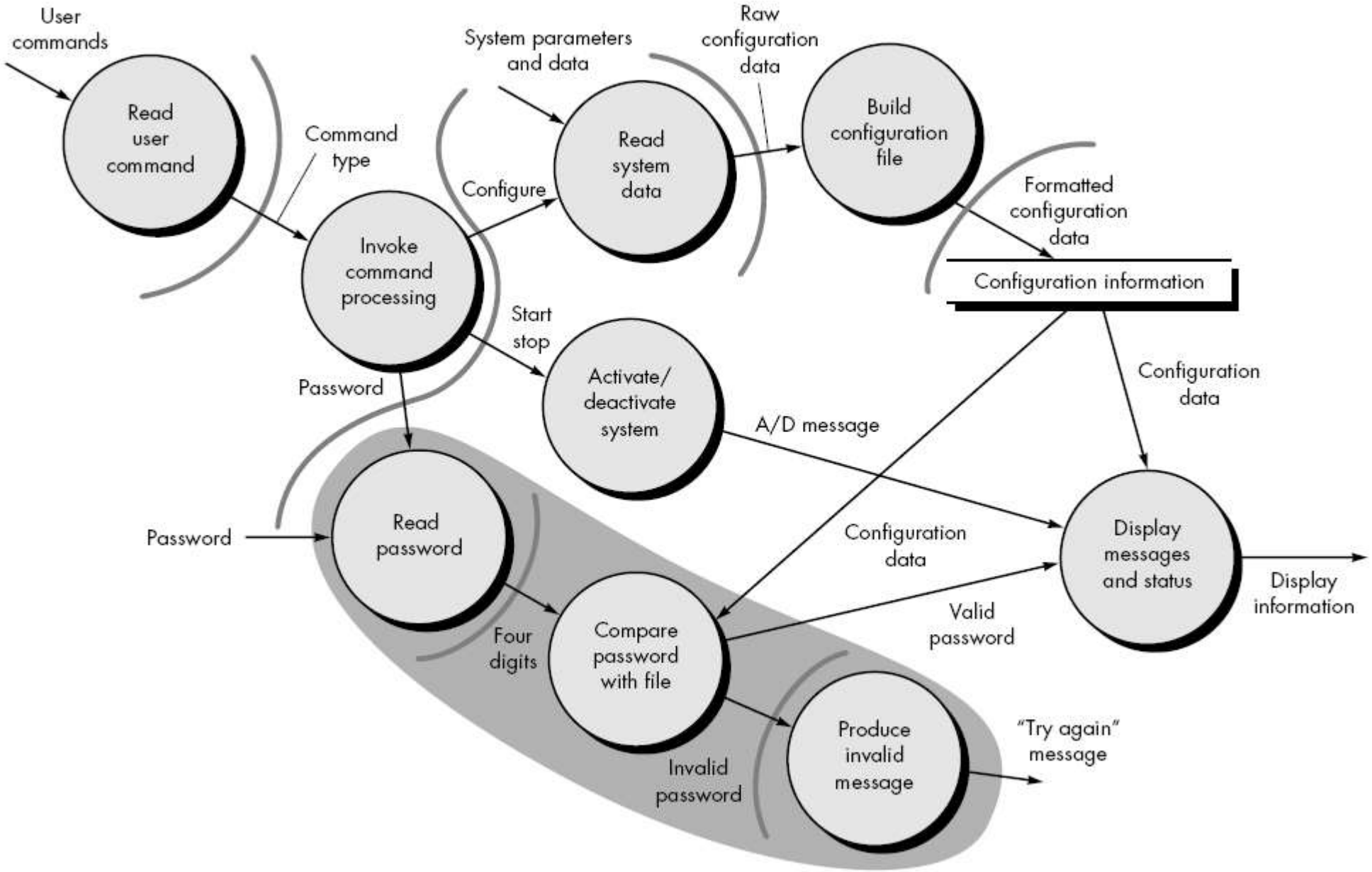
Transaction mapping

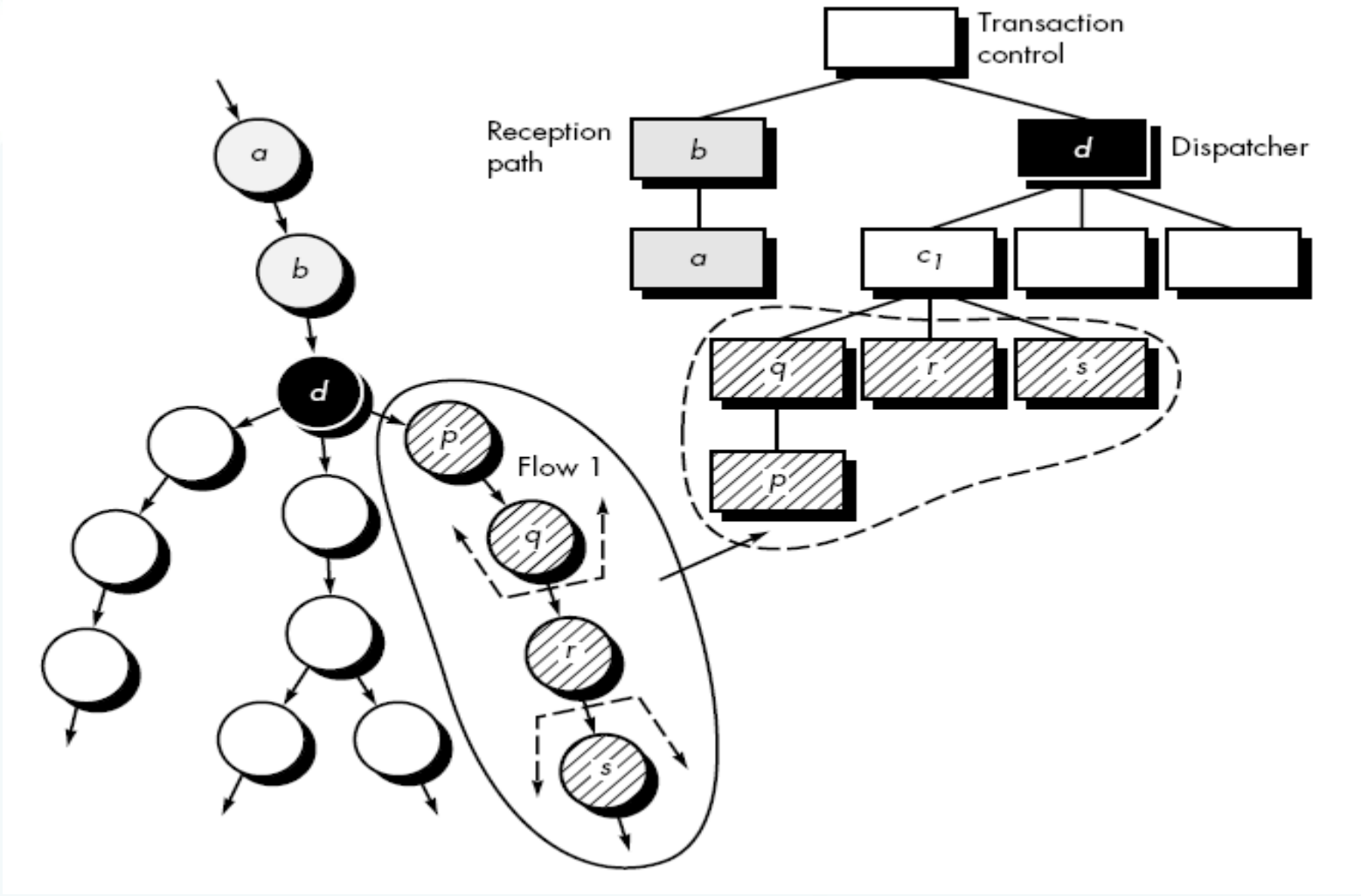
- In many software applications, a single data item triggers one or a number of information flows that effect a function implied by the triggering data item. The data item, called a transaction, and its corresponding flow characteristics . In this section we consider design steps used to treat transaction flow.
- An Example
- Transaction mapping will be illustrated by considering the user interaction subsystem of the SafeHome software.
- As shown in the figure in next slide, user commands flows into the system and results in additional information flow along one of three action paths. A single data item, command type, causes the data flow to fan outward from a hub. Therefore, the overall data flow characteristic is transaction oriented.
- It should be noted that information flow along two of the three action paths accommodate additional incoming flow (e.g.,


User commands
Read user command
Command type
Invoke command processing
Configure
System parameters and data
Read system data
Raw configuration data
Build configuration file
Formatted configuration data
Configuration information
Start stop
Activate/ deactivate system
A/D message
Configuration data
Password
Password
Read password
Four digits
Compare password with file
Configuration data
Valid password
Display messages and status
Display information
Invalid password
Produce invalid message
"Try again" message

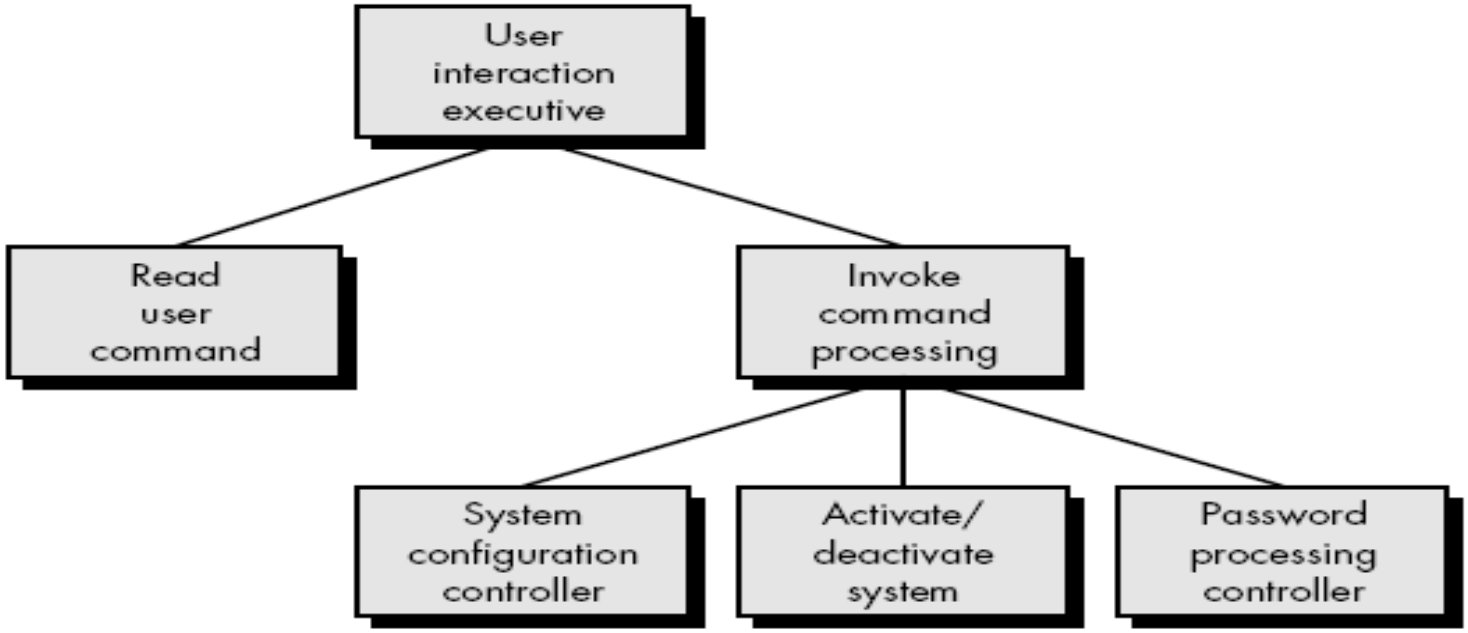
- Design Steps
- The design steps for transaction mapping are similar and in some cases identical to steps for transform mapping . A major difference lies in the mapping of DFD to software structure.
- **Step 1. Review the fundamental system model.**
- **Step 2. Review and refine data flow diagrams for the software.**
- **Step 3. Determine whether the DFD has transform or transaction flow**

- **Step 4. Identify the transaction center and the flow characteristics along each of the action paths.** The location of the transaction center can be immediately discerned from the DFD. The transaction center lies at the origin of a number of actions paths that flow radially from it. For the flow shown in figure , the invoke command processing bubble is the transaction center.
- The incoming path (i.e., the flow path along which a transaction is received) and all action paths must also be isolated. Boundaries that

- **Step 5. Map the DFD in a program structure amenable to transaction processing.**
  Transaction flow is mapped into an architecture that contains an incoming branch and a dispatch branch. The structure of the incoming branch is developed in much the same way as transform mapping. Starting at the transaction center, bubbles along the incoming path are mapped into modules. The structure of the dispatch branch contains a dispatcher module that controls all subordinate action modules. Each action flow path of the DFD is mapped to a structure that corresponds to its specific flow


Transaction control
Reception path
b
d
Dispatcher
a
c_1
q
r
s
p
a
b
d
p
Flow 1
q
r
s

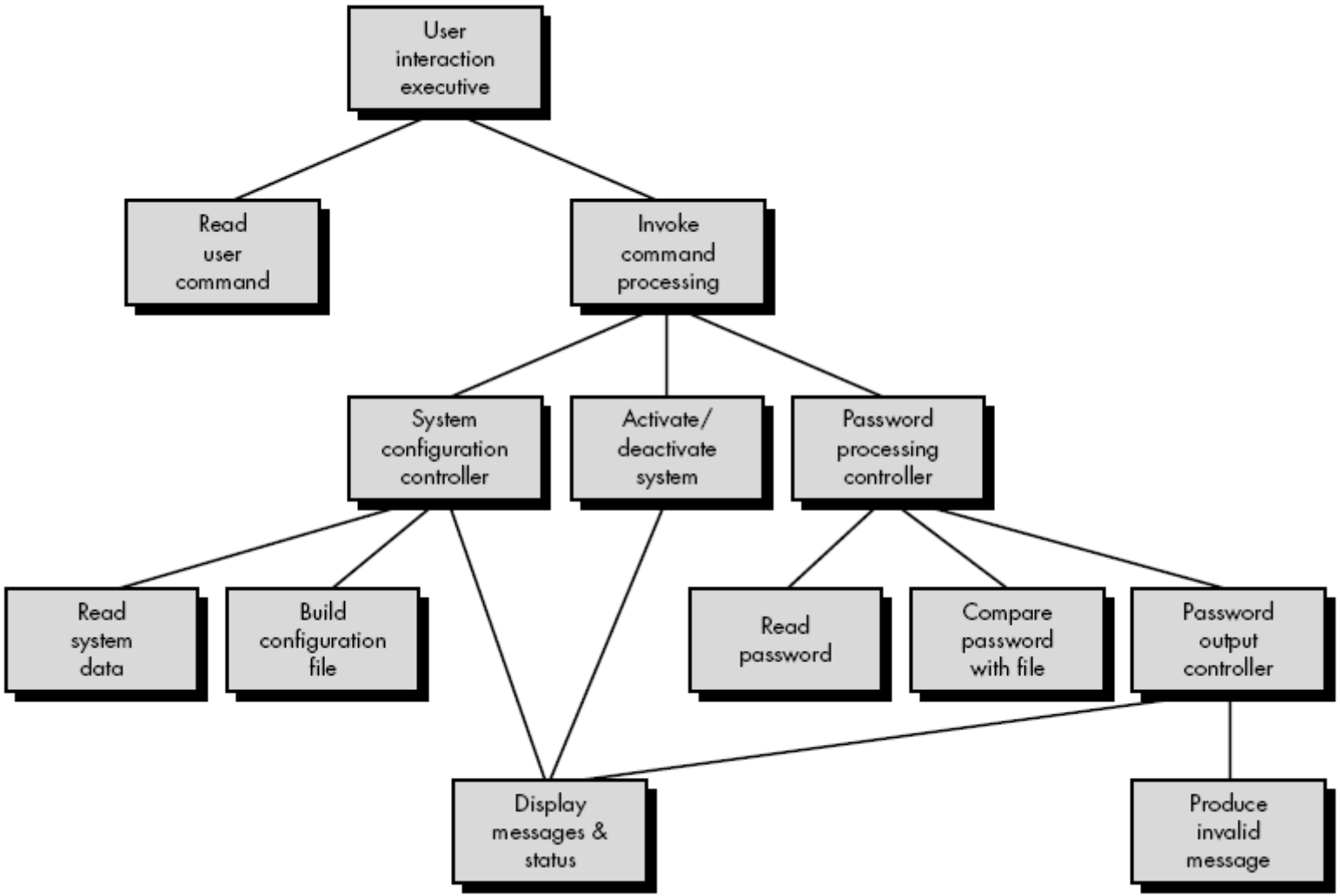
- Considering the user interaction subsystem data flow, first-level factoring for step 5 is shown in below figure.



The bubbles read user command and activate/deactivate system map directly into the architecture without the need for intermediate control modules. The transaction center, invoke command processing, maps directly into a dispatcher module of the same name. Controllers for system configuration and password processing are created as illustrated in figure.

- **Step 6. Factor and refine the transaction structure and the structure of each action path.** Each action path of the data flow diagram has its own information flow characteristics. We have already noted that transform or transaction flow may be encountered. The action path-related "substructure" is developed using the design steps discussed in this and the preceding section.

- As an example, consider the password processing information flow shown (inside shaded area) in figure The flow exhibits classic transform characteristics. A password is input


User interaction executive
Read user command
Invoke command processing
System configuration controller
Activate/ deactivate system
Password processing controller
Read system data
Build configuration file
Read password
Compare password with file
Password output controller
Display messages & status
Produce invalid message

- Step 7. Refine the first-iteration architecture using design heuristics for improved software quality. This step for transaction mapping is identical to the corresponding step for transform mapping. In both design approaches, criteria such as module independence, practicality (efficacy of implementation and test), and maintainability must be carefully considered as structural modifications are proposed.