Chapter 8
DSS DEVELOPMENT TOOLS

1. Distinguish between DSSs and the tools used to build DSSs.
   Knowledge management techniques can be implemented in many tools and a tool can
   furnish multiple knowledge management techniques.
   (i.e., spreadsheet tools emphasize a spreadsheet technique, but can also use graphical techniques)
   GURU is a tool that uses a synergistic approach. It uses database techniques, spreadsheet
   techniques, word processing techniques, etc.

2. Identify important factors that deserve to be considered when selecting tools for constructing
   your DSSs.
   Technique-oriented tool categories
     - Which technique is emphasized?
     - May fit multiple categories
   Roles of tools in development
     - Assist at analysis/design stages versus implementation stage
     - Roles in implementation
       - **intrinsic tool**
         - functions as PPS
         - developer initializes KS
         - LS, PS customization possible
         - widely used
           (i.e. Excel, Lotus 1-2-3, etc.)
       - **partially intrinsic tool**
         - functions as part of PPS
         - developer designs/implements rest of PPS
           (i.e. Dbase, Access, etc.)
       - **extrinsic tool**
         - does not function as any part of PPS
         - used to produce parts of the PPS
         - may be used to produce some KS contents

3. Describe major interface styles that can be built into DSSs.
   **Interface styles**
   - Users have different preferences (friendliness is in the eye of the beholder)
   Three things that effect user friendliness:
     a. nature of the problem
     b. type of task
     c. nature of the user
   - Preference can change
     - depending on task
     - depending on experience
   - Interface refers to
     - LS
-PS
-linguistic and presentation knowledge
-portion of PPS that accepts LS requests and presents PS responses

-To a user, the interface is the system
-Should be adaptable
  -easy to learn
  -graceful shifting among tasks
  -high level of guidance/feedback based on prior interactions
  -gives user a sense of being in control
  -provides multiple interface styles

-Interface can influence the impact a DSS has on decision making

-Interfaces can be provided by or built with tools

-Language system possibilities
  -command-oriented
    -range from procedural to nonprocedural
    -user must learn LS vocabulary and grammar
  -Natural Language
    -system adjusts its LS to user’s presentations
    -problem of misinterpretation
  -Menus
    -gives guidance to users about possible requests
    -user reacts to presented options
    -LS consists of keystroke sequences (mouse manipulations, etc.) that allow users to make various requests
  -Forms
    -provides guidance to user about filling in slots in forms
    -can edit any items entered prior to completion
    -LS comprised of possible entries for forms
  -Question/Answer
    -simple case of a form
    -LS comprised of sequences of keystrokes that could make up answers to questions
  -Direct manipulation
    -requests made by manipulating system presentations of objects
    -LS comprised of user actions for manipulating PS elements
    -examples include forms interfaces, graphical user interfaces
  -Speech
  -Hybrids

-Presentation system possibilities
  -assistance messages
    -to guide user’s requests
    -help text
    -diagnostics
  -results messages
4. Explain the relationship between DSS development tools and knowledge management techniques, including ways of integrating multiple techniques in the construction of a single DSS.

Five types of software integration involving:
- Knowledge represented in a certain format. (K)
- Knowledge manipulation-depends on a certain format (P)
- Program devised to manipulate knowledge organized in a certain way (part or whole of a PPS).
- Knowledge transfer- from one repository to another (possibly reorganizing it)

1) **Integration via knowledge format conversion.**
   - conversion utility or import/export facility
   - as number of programs increase, number of utilities becomes large
   - knowledge transfer takes time and effort
   - knowledge manipulation requires program switching
   - PPS may be developed to handle transfers and switching
   - redundancy can lead to consistency problems

2.) **Integration via a clipboard**
   - common intermediate format (“clipboard”)
   - requires fewer conversion utilities
   - knowledge transfer involves “cut/copy” and “paste”
   - programs may be able to directly manipulate clipboard contents
   - knowledge manipulation queries program switching
   - redundancy can lead to consistency problems
3.) **Integration via a common format**

- all programs designed to manipulate common format
- no knowledge transfer required
- program switching still needed
- reduces redundancy/consistency problems
- difficult/inefficient to devise a single format for diverse software functionality
  *(difficult to implement-single format required for universal processing)*

4.) **Integration via nesting**

- prior approaches use knowledge as “glue” that binds diverse software functionality’s
- alternative is to incorporate functionality’s of diverse techniques into a single tool
- one way to do this is to nest capabilities within a single program
- one dominant component, plus secondary components
- do switching among programs
- no need for knowledge transfer
- must know the dominant technique
- dominant component constrains use of others
5.) **Synergistic integration**

alternative to nesting
single program without a dominant functionality
any can be used independent of others, but without switching
can use multiple techniques in a single operation
thus, traditional dividing lines among techniques blur
one program that manipulates knowledge represented in multiple formats (even for a
single request)
avoid limitations of being restricted to a single format

**6.) Combination**

select tools and integrate across them using foregoing approaches
select tool in which needed techniques are integrated