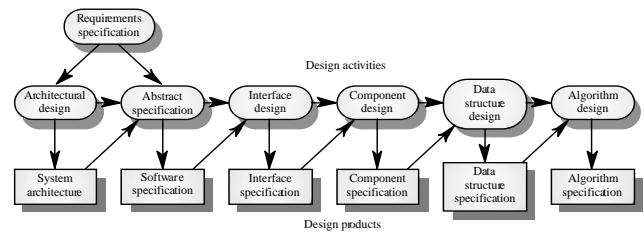


Function-oriented design

The software design process



Functional and object-oriented design

- For many types of application, object-oriented design is likely to lead to a more reliable and maintainable system
- Some applications maintain little state - function-oriented design is appropriate
- Standards, methods and CASE tools for functional design are well-established
- Existing systems must be maintained - function-oriented design will be practiced well in the 21st century

Functional design process

- Data-flow design
 - Model the data processing in the system using data-flow diagrams
- Structural decomposition
 - Model how functions are decomposed to sub-functions using graphical structure charts
- Detailed design
 - The entities in the design and their interfaces are described in detail. These may be recorded in a data dictionary and the design expressed using a PDL

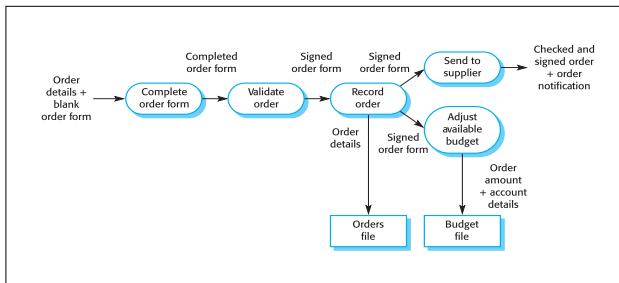
Data flow diagrams

- Show how an input data item is functionally transformed by a system into an output data item
- Are an integral part of many design methods and are supported by many CASE systems
- May be translated into either a sequential or parallel design. In a sequential design, processing elements are functions or procedures; in a parallel design, processing elements are tasks or processes

DFD notation

- Rounded rectangle - function or transform
- Rectangle - data store
- Circles - user interactions with the system
- Arrows - show direction of data flow
- keywords and/ or. Used to link data flows

Order processing DFD



©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 7

Structural decomposition

- Structural decomposition is concerned with developing a model of the design which shows the dynamic structure i.e. function calls
- This is not the same as the static composition structure
- The aim of the designer should be to derive design units which are highly cohesive and loosely coupled
- In essence, a data flow diagram is converted to a structure chart

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 8

Decomposition guidelines

- For business applications, the top-level structure chart may have four functions namely input, process, master-file-update and output
- Data validation functions should be subordinate to an input function
- Coordination and control should be the responsibility of functions near the top of the hierarchy

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 9

Decomposition guidelines

- The aim of the design process is to identify loosely coupled, highly cohesive functions. Each function should therefore do one thing and one thing only
- Each node in the structure chart should have between two and seven subordinates

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 10

Process steps

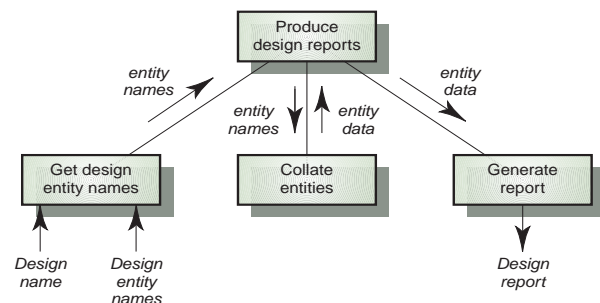
- Identify system processing transformations
 - Transformations in the DFD which are concerned with processing rather than input/output activities. Group under a single function in the structure chart
- Identify input transformations
 - Transformations concerned with reading, validating and formatting inputs. Group under the input function
- Identify output transformations
 - Transformations concerned with formatting and writing output. Group under the output function

©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 11

Initial structure chart

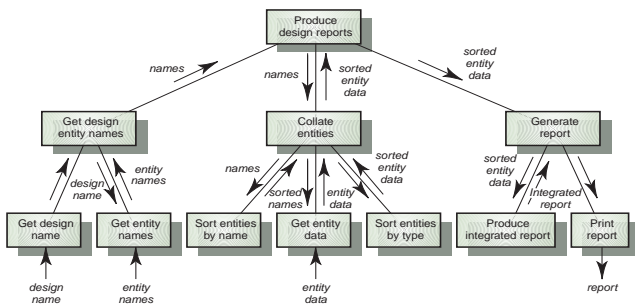


©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 12

Expanded structure chart

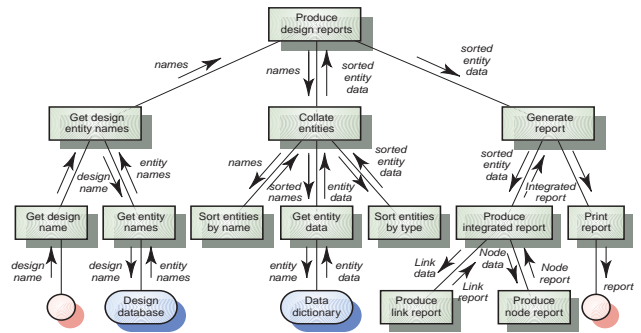


©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 13

Final structure chart



©Ian Sommerville 2004

Software Engineering, 7th edition, Chapter 1

Slide 14