

Second-Level Decomposition: Hardware-Hiding Module

1. extended computer module
2. device interface module

Extended Computer Module

- hides that part of the HW/SW interface that is likely to change
 - when computer modified
 - when computer replaced
 - same for operating system, if used
- example A-7E computer:
 - floating point unit or software simulation?
 - single / multi-processor?
- extended computer provides a virtual machine that can be implemented efficiently on all likely platforms

- primary secrets for A-7E computer:
 - number of processors
 - instruction set of the computer
 - capacity for concurrent operations

Device Interface Module

- hides that part of peripheral devices that is likely to change
 - each device might be replaced by an improved one capable of the same tasks
- example A-7E:
 - all angle-of-attack sensors measure angle between reference line on aircraft and the velocity of the air
 - they differ in: input format, timing, amount of noise

- **module provides virtual devices**
 - sometimes one virtual device corresponds to several hardware devices
 - sometimes the capabilities of a physical unit may change independently: then hide in different modules
- **primary secrets for A-7E:**
 - those characteristics of the present devices that
 - ▷ are documented in the requirements document
 - ▷ are not likely to be shared by replacement devices

Second-Level Decomposition: Behaviour-Hiding Module

1. function driver module
2. shared services module
 - supports function driver module

Function Driver Module

- a set of individual modules (“function drivers”)
- each function driver is sole controller of a set of closely related outputs
 - outputs related closely: if it is easier to describe their values together than individually
 - example: sine of an angle, cosine of same angle
- these outputs go to the virtual devices
- primary secrets: the rules determining the values of the outputs

Shared Services Module

- some aspects are common to two or more function drivers
 - A-7E: they control the same aircraft
 - odometer example: the display mode
- a shared services module hides one such aspect

Searching for a Behaviour-Hiding Module

- documentation users:
 - will not know which aspects are shared
- documentation for the function driver modules:
 - must have a reference to the shared services modules used
- start search:
 - always with function driver

Second-Level Decomposition: Software Decision Module

1. application data type module
 - hides implementation of certain variables
2. physical model module
 - hides algorithms that simulate physical phenomena
3. data banker module
 - hides data-updating policies
4. system generation module
 - hides decisions that are postponed until system generation time
5. software utility module
 - hides algorithms used in several other modules

Application Data Type Module

- supplements data types by extended computer module
- provides data types useful for avionics that do not require a computer dependent implementation
- primary secrets: the data representation of the variables
 - variables can be used without units
 - where necessary, the modules provide unit conversion operators which deliver or accept values in specified units

Physical Model Module

- software requires estimates of quantities that cannot be measured directly,
but can be computed from other observables
- primary secrets: the physical models
- secondary secrets: the implementations of the models

Data Banker Module

- most data:
 - produced by one module and consumed by another
- usually: consumer gets value as up-to-date as practical
- data banker: middle-man, determines update policy
- if update policy changes:
 - change neither producer nor consumer
- don't use data banker if consumer requires
 - specific members of value sequence
 - values with a specific time (e.g., when an event occurs)

Some Data Update Policies

name	store	when new value produced
on demand	no	whenever a consumer requests the value
periodic	yes	periodically. consumer gets most recently stored value
event driven	yes	whenever data banker is notified by an event of a possible change
conditional	yes	whenever a consumer requests the value, provided certain conditions are true. otherwise: previously store value

Choice of Updating Policies

- consumers' accuracy requirements
 - how often consumers require the value
 - max. wait that consumers can accept
 - how often the value changes
 - cost of producing a new value
-
- the policy decision does not depend on coding details of consumer or producer
 - data banker usually not rewritten if producer or consumer change

System Generation Module

- **primary secrets:**
 - decisions that are postponed until system generation time
 - system generation parameters
 - choice among alternative implementations
- **secondary secrets:**
 - method used to generate executable code
 - representation of the postponed decisions
- **these programs do not run on on-board computer**
 - A-7E: cross-platform build

Software Utility Module

- primary secrets: the algorithms implementing common software functions and mathematical routines
 - resource monitor
 - square root, logarithm, . . .