Review of A Procedure for Designing Abstract Interfaces for Device Interface Modules by D. L. Parnas et al.

Jianjun Hu
Yakub Sailanawala
CSE870
MSU 2002

A Procedure for Designing Abstract Interfaces for Device Interface Modules
Heninger, K. Parker, R. Parnas, D.L.
IEEE Fifth International Conference on Software Engineering - 1981

• PROBLEM
  • Changes in hardware device interfaces often lead to widespread changes in the whole system
  • The device interface module [DIM] containing the device dependent code

• SOLUTION
  • The remaining software components that should be independent of device specific details.

• GOAL
  • design good abstract interfaces for DIM

Naval Research Lab's redesign of the flight SW for A-7 aircraft

Designing Abstract Interfaces
… is to obtain its dual description

• Assumption List: List of all the assumptions about the characteristics of the hardware devices the user programs are allowed to make

• Functional Description: API & Events Specification

Assumption List explicitly states the assumptions about the characteristics that are implicit in the Functional description

These descriptions must be reviewed by the users, hardware engineers and experienced programmers
Some Problems and Tradeoffs'

- Characteristics that change independently should be contained in sub-modules
- Certain variable parameters can be specified either at system generation time or at runtime based on the cost of change and probability of change
- Predictable changes or enhancements in the hardware device should be mentioned in assumptions even if not implemented

Commonality Analysis: A Systematic Process for Defining Families

David M. Weiss

Analytical technique for defining families
- Structured format for all the elements that go into designing and defining families
- Process Description
- Commonality Analysis refers to both, the end product of this analysis in the form of a document as well as the process

Contents of a Commonality Analysis document
- Introduction
- Overview
- Dictionary of Terms - Standard set of terms used for the description of the family
- Commonality – Parnas et al. referred to this section as the Assumption List
- Variability – Documents the possible variations
- Parameters of variation – Quantifies the variabilities
- Issues
- Appendices
Stages in the Analysis process

**PLAN**
Define purpose and scope of the family

**ANALYZE**
- Define Terms
- Identify Commonalities
- Identify Variabilities

**EXTERNAL REVIEW**

**QUANTIFY**
Define parameters of variation

---

**Structuring formal control systems specifications for reuse: Surviving hardware changes**

*J. Thompson, M. Heimdahl and D. Erickson*
*In Proc. 5th NASA Langley Formal Methods Workshop - 2000*

**Problem**
To design software control system for members of a family that displays the same high level behavior and survives changes in replaceable hardware devices

**Proposed Solution**
Obtain a high-level software specification of the requirements

---

**General View of Software Controlled Systems and 4 Variable Model**

- **MON**
- **REQ**
- **CON**
- **IN**
- **OUT**
- **SOFTWARE**
- **INPUT**
- **OUTPUT**
- **Sensors**
- **Actuators**
- **Controller**
- **Process**
- **Monitored Variables**
- **Controlled Variables**
4 Variables Model was proposed by D. Parnas and J. Madey

**Conclusion**

- The proposed decomposition of the software system results in a module (SOFT\_REQ) that represents the system requirements reusable over control systems exhibiting same high level behavior.
- The sensors related information is confined to the module representing IN\(^{-1}\).
- The actuators related information is confined to the module representing OUT\(^{-1}\).

**Device Independent Navigation and Interaction in Virtual Environments**

- **Objective:**
  - Achieving independence of virtual environment applications from particular available input devices
- **Mapper vs. DIM of Parnas’ paper**
- **Major extensions**
  - Integrate Different set of input devices
  - Layered interfaces
  - Automatic configuration (selection of input devices)
Why it should cite Parna’s paper?
- Mapper is similar to DIM for separating device independent components from device dependent components
- Also discussed the issues of data transformation and emulation
- Parna’s Procedure and principles can be suitably applied here
- Natural extension of DIM

An Abstract-Device Interface for Implementing Portable Parallel – I/O Interfaces

- Objective:
  - Portable and efficient Parallel-I/O Interface
  - Facilitate implementation of any parallel I/O API on any file systems
  - Many File systems: Unix, PFS, PIOFS, NFS, ...
  - Many Parallel I/O API systems: IBM PIOFS, IntelI/O, RIO…
- Approach:
  - ADIO: Abstract -device interface for parallel I/O
  - Similar to DIM, Mapper
- Difference
  - Not used directly by application programs but by higher layer API
An Abstract-Device Interface for Implementing Portable Parallel –I/O Interfaces (Cont’)

Limitations
- this paper doesn’t provide some guidelines and procedure for how to design interfaces.
- doesn’t discuss the trade-off of the levels of exposing the details of underlying file systems.

Why should cite Parnas’ paper?
- ADIO similar to DIM
- Similar issues concerned
- Parnas’ procedure can be applied to support the design of ADIO

Hints for computer system design

Cited
Paper 3


• Problem: Computer system design
  – Functionality
  – Speed
  – Faulty tolerance

• Functionality design & Interface Design
  Interfaces are agreements on the functionalities that system should provide.

Hints for computer system design (Cont’)

• Interface Design in Computer System
  – Simplicity, not generality
    Predictable cost.
  – Completeness
  – Don’t hide power
    Sacrifice power to generality
  Concept of Transparency of Hierarchical System [Parnas. 1975]
  – Flexibility by procedure parameter