
Abstract
XH: Experience and/or Heuristic
Studies reported here of <rearchitecting a jet’s onboard flight control system> supported by <an information hiding design heuristic> generate a number of findings concerning <the most efficient way to modularize software and documentation>, including <the importance of creating a cross-referenced module guide and focusing on what portions of the requirements are most likely to change>. They indicate that <requirement> is met by <decomposition based on information hiding>.

Why XH?
The paper is largely a confirmation that modularization by information hiding is a viable way of designing and documenting software. Their experience also generated a list of ideas that complement the main information hiding design heuristic.

Question - [Method/means of development]
<i>issues</i>
What is the most efficient way to modularize software and documentation?
Main criteria: understandability by people joining the team, and maintainability.

Results - [Procedure / technique]
<design-heuristic>
The authors favor a decomposition of software and documentation based on information hiding.

<i>list-of-findings</i>
Designers should first create a cross-referenced module guide to organize the design of individual interfaces. They should take account of what portions of the requirements will likely change, and then encapsulate the corresponding functionality as a “secret” of a module.

Validation – [Experience]
<i>application</i>
The authors rearchitected a jet’s onboard flight control system.

<i>supporting technology</i>
The authors put their information hiding design heuristic into action.