

CMSI 370: The Longevity of the Hewlett-Packard 12c Calculator

Chris Whiting

Abstract

The success and popularity of a user interface can be attributed to its correct alignment of the user and developer mental models. The developer mental model is the designer's conceptual model. The user model is the mental model created from the user's interaction with the system. The system image is the physical result of the developer's mental model. The developer expects the user's mental model to be the same as the designer's mental model. When a user interacts with a system, they communicate with the system image. However, if the system image does not accurately reflect the developer's mental model, then the user will end up with the wrong mental model for the system. The correct balance of the user's mental model, the developers mental model, and the system image is present in the Hewlett-Packard 12c Calculator.

Contents

1	Introduction	5
2	Basic Financial Functions	5
3	Error Preventing/Fixing Keys	5
3.1	The Exchange Key	5
3.2	The Last Key	6
4	Writing Custom Programs	6
5	Conclusion	6

List of Figures

List of Tables

1 Introduction

The popularity and longevity of the HP 12c calculator can be attributed to the following factors: , specific functions for financial applications, the exchange and last keys help correct user errors, and the ability to write programs. When using the hp12c, the user's mental model changes according to the specific operation being performed. When a calculation involving a specific financial function is being performed, the user's mental model consists of the givens, the unknowns, and the method of solving. Error correction helps the user pick up from their current mental model from the last point that was correct. Writing programs to fit any financial problem, lets the user create their own developer model, which therefore must equal the user's mental model. This is the most important factor of the hp12c, because it allows the user to create their own developer model and system image.

<http://h10032.www1.hp.com/ctg/Manual/c00363319.pdf>

2 Basic Financial Functions

The basic financial functions of the hp 12c helps to simplify, ordinarily complex financial calculations. To accomplish this, the calculator utilizes special financial registers in memory. The memory of the hp12c also helps improve calculation efficiency along with the financial functions. The financial registers are designated by n, i , PV, PMT, and FV as shown in Figure 1. The n financial register denotes the number of days, i represents the annual interest rate, PV denotes the principal amount or present value, PMT is the register standing for the period payment, and the FV register represents the future value during calculations. Solving simple financial calculations is only a matter of keying the correct quantities in the cooresponding financial registers with the register buttons shown in Figure 1.

These registers represent the system image that is based on the developer's model. When the user attempts to solve a financial problem, they create a mental model based on the specific calculation required. The financial functions registers create a clearer system image from the developer's mental model, by allowing the user to systematically enter their calculation. Each step in solving a financial calcaultion will be envisioned by the user and thus the user's mental model is developed. The system image for the operation of the financial functions and registers align with the user's mental model, in terms of steps to reach the answer. That is, the steps the user envisions for any financial calculation can almost directly be keyed on the hp 12c with the financial functions and registers.

3 Error Preventing/Fixing Keys

The outline after the introductory and background, preliminary, and related work sections is more dependent on the specific subject of your research. Remember to cite references where appropriate, organize the material so that it flows well and is clear to the reader.

3.1 The Exchange Key

L^AT_EX has support for up to three outline levels (`\section`, `\subsection`, and `\subsubsection`). It also recognizes `\paragraph` and `\subparagraph` directives, though those don't show up in the table of contents. All of these directives expect a title.

Note also the use of the `\verb` directive for inserting code-like labels or symbols. It was particularly needed here so that we can include the backslash character in the text.

3.2 The Last Key

4 Writing Custom Programs

We’re adding another section just so you can see how that looks. Plus there are a few more \LaTeX features to illustrate.

5 Conclusion

Wrap up your paper with an “executive summary” of the paper itself, reiterating its subject and its major points. If you want examples, just look at the conclusions from the literature.

References

- [1] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, and Steven Jacobs. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publishing Company, USA, 5th edition, 2009.
- [2] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. John Wiley & Sons, 7th edition, 2005.
- [3] Apple Computer, Inc. *I/O Kit Fundamentals*. Apple Computer, Inc., 2005. <http://developer.apple.com/documentation/DeviceDrivers/Conceptual/IOKitFundamentals>.
- [4] Greg Wilson. *Exploring Palm OS: Memory, Databases, and Files*. PalmSource, Inc., 2004.
- [5] Steve Best. *User-Mode Linux*, chapter 11. Prentice Hall, 2005.
- [6] Jakob Nielsen. Book review: Designing the user interface: Strategies for effective human-computer interaction by ben shneiderman (addison-wesley, 1987). *SIGCHI Bull.*, 18(3):85–86, January 1987.
- [7] Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis. *Commun. ACM*, 55(4):45–54, April 2012.
- [8] Keith Loepere. Stackable thread mechanisms. *SIGOPS Oper. Syst. Rev.*, 39(4):4–17, 2005.
- [9] Mark Russinovich and David Solomon. Windows XP: Kernel improvements create a more robust, powerful, and scalable OS. *MSDN Magazine*, 16(12), December 2001.
- [10] Jonas Schild, Joseph LaViola, and Maic Masuch. Understanding user experience in stereoscopic 3d games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’12, pages 89–98, New York, NY, USA, 2012. ACM.
- [11] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. iring: intelligent ring using infrared reflection. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST ’12, pages 131–136, New York, NY, USA, 2012. ACM.
- [12] Craig A. N. Soules and Gregory R. Ganger. Connections: using context to enhance file search. In *SOSP ’05: Proceedings of the twentieth ACM symposium on Operating systems principles*, pages 119–132, New York, NY, USA, 2005. ACM Press.
- [13] Thomas Hallgren, Mark P. Jones, Rebekah Leslie, and Andrew Tolmach. A principled approach to operating system construction in Haskell. In *ICFP ’05: Proceedings of the tenth ACM SIGPLAN international conference on Functional programming*, pages 116–128, New York, NY, USA, 2005. ACM Press.

- [14] Qingbo Zhu, Zhifeng Chen, Lin Tan, Yuanyuan Zhou, Kimberly Keeton, and John Wilkes. Hibernator: helping disk arrays sleep through the winter. In *SOSP '05: Proceedings of the twentieth ACM symposium on Operating systems principles*, pages 177–190, New York, NY, USA, 2005. ACM Press.