

人に優しいソフトウェアを設計するためには

Key Concepts to Design the Human-Oriented Software

北 英彦

Hidehiko Kita

(電気電子工学科 Department of Electrical and Electronic Engineering)

(Received September 13, 1996)

Abstract

Computers such as personal computers and machine-embedded computers come into wide use in our daily life. Accordingly, Software, especially, human-oriented software executing on these computers become important. In this paper, I describe some key concepts to design the human-oriented software.

Key Words: software quality, software bugs, software reliability, groupware, computer-supported cooperative work

1. はじめに

仕事場や趣味などで使用するパーソナルコンピュータ、炊飯器、掃除機などの家庭電化製品の中のマイクロコンピュータなど我々の身の回りには計算機が満ちあふれている。そのなかで、「ソフトウェアのない計算機はただの箱」といわれるように、ソフトウェア、特に、人間を中心として考えたソフトウェアの重要性が増している。

本論文では、著者がソフトウェアに関する研究を行う上で常に考えている人間を中心として考えたソフトウェア、すなわち、「人に優しいソフトウェアを設計するためには」どうしたらよいかということについて述べる。「人に優しいソフトウェア」という言葉は、ソフトウェアを使う人、その人が生きている時代や文化などによって、また、使われている技術により異なるものなので、ここでは「人に優しいソフトウェア」という言葉が何であるかという定義を与えるのではなく、「人に優しいソフトウェア」を考える上で著者が重要と考えているキーワードをいくつか紹介する。

2. 計算機の使われ方

「人に優しいソフトウェア」とは何かを考える前に、計算機がどのように使われているかを考える。

計算機は、最初はその名前の通り計算をする機械として作られた。よく知られているように、世界で一番最初といわれるデジタル計算機の目的は、大砲の弾丸の飛び方、すなわち、弾道を計算をすることであった。

その後、計算機の種類として、個人で使うためのパーソナルコンピュータが世の中に登場した。最近では、パーソナルコンピュータの低価格化とユーザインタフェース技術の向上により、文書の作成はワードプロセッサ（いやゆるワープロ）を用いるのがごく普通となった。また、パーソナルコンピュータの高速化と高機能化により、文字だけではなく絵や図も簡単に扱えるようになった。この結果、計算機は、計算する機械ではなく紙と鉛筆と同じような文房具になったと考えられる。

また、ここ数年の計算機ネットワークの普及には目を見はるものがある。計算機ネットワークを利用したソフトウェアの研究を行っている著者ですら予想もしなかった勢いで使われはじめている。パーソナルコンピュータが計算機ネットワークにつながることにより、電子メール、WWW（World Wide Web）を普通の利用者が使用するようになった。この結果、計算機は電話やファクシミリのようなコミュニケーションの手段として使われるようになった。このことは、さらに、計算機が単なる文房具ではなく、仕事場の一部になったことを示している。

バーチャルリアリティ、グループウェア、マルチメディア通信などソフトウェアに関する最近の研究・開発の成果が、一般に使用されるソフトウェアで普通に使われるようになれば、計算機はますます仕事場そのものになると考えられる。

これまでは利用者が直接使用する計算機について述べてきたが、この他にマイクロコンピュータとして他の機械に埋め込まれているものも多くある。身の回りで考えただけでも、炊飯器、洗濯機などの家庭電化製品、信号機、列車制御システムなどの交通制御機器、CTスキャン、NMRなどの医療機器がある。

以下では、これらの計算機を念頭において「人に優しいソフトウェア」について考察を進める。

3. ソフトウェアの品質

次に「人に優しいソフトウェア」もその中に含まれるであろう「よいソフトウェア」とは何かについて考える。「よいソフトウェア」とは何かを考えるためには、何に関して良いのか悪いのかというソフトウェアを評価するときの視点が重要である。評価の視点はいろいろ考えることができるが、JIS規格として定められたソフトウェアの品質特性および品質副特性のリスト [付録]、[1] が評価の視点として考えたときに比較的好くまとまっているのでそれについて考察する。

JIS規格では、品質特性としては、機能性、信頼性、使用性、効率性、保守性、移植性の6つが規格として定められている。さらに、これらは附属書の中で21の品質副特性に分類されている。

しばらく前までは、「よいソフトウェア」というのは、処理速度の速いこと、メモリを使わないことが決まり文句のようにいわれてきた。もちろん、これらの品質特性も重要ではあるが、開発あるいは導入しようとしているソフトウェアが利用者の目的にあっているかという合目的性、ソフトウェアが正しく動くかということを保証する正確性を品質副特性が利用者の要求に対して十分満たされていないからではない。

「人に優しいソフトウェア」という観点からは、何に使うソフトウェアであるか、どういう機能を持つかということがどのくらい分かりやすいかという理解性、そのソフトウェアを初めて利用する利用者でも苦勞せずに使えるようになるかという習得性が特に重要である。

4. ソフトウェアの安全性

ソフトウェアの品質特性については、上記にあげたJIS規格の中で述べられている品質特性よりさらに重要であると考えソフトウェアの安全性について考察する。

計算機でワードプロセッサソフトウェアなどの応用ソフトウェアを使っていて、通常「ダウンした」、「おちた」などといわれる状態になることはよくある。ひどい場合には、計算機全体が使用不能の状態になる。これらの障害の原因を通常バグと呼んでいる。

ソフトウェアを開発する人の間でよく言われるのは、「バグのないソフトウェアはない」とか、「ソフトウェアにはバグはつきもの」とかいう言葉である。ソフトウェアの開発者は、バグのないように工夫や努力を重ねてはいるが、「バグのないソフトウェアはない」というのは著者の経験からも、ソフトウェアの研究者の間でも、100%正しい事実とされている。

マイクロコンピュータのような他の機械に埋め込まれているものを含めると、私たちの身の回りは計算機であふれている。炊飯器、洗濯機などの家庭電化製品、信号機、列車制御システムなどの交通制御機器、CTスキャン、NMRなどの医療機器、アンチロックブレーキなどの自動車の制御装置、あげればきりがなし。計算機は、ソフトウェアがなければ動かないので、これらの機械の中でも当然ソフトウェアが使われている。

「バグのないソフトウェアはない」というのが事実であるとする、医療機器、交通制御機器などのように人の生命に関わる機械においても、ソフトウェアにバグが必ず存在するであろうということは、考えると恐ろしい事実である。ここで、文献からいくつか事例を引用する [2]、[3]、[4]、[5]。

- (1) 1985年米国テキサス州タイラーの病院では、放射線医療装置のソフトウェアのバグのため、致死量の放射線を患者に照射したために、死者がでた。
- (2) 1991年FAA（連邦航空管制局）は、雨や霧などの視界が悪いときでも航空機が着陸できるようにするためのシステムである、無指向性ビーコン着地アプローチシステムにバグがあり、間違ったコースを指示することがあると各航空会社へ警告した。

文献の中には、これら以外にも驚くほど多くの事例が紹介されている。前章で述べたJIS規格のソフトウェアの品質特性の中には安全性という項目がないが、著者は人の生命に関わるソフトウェアにおいては安全性が一番重要な品質であると考え。

5. システムは何故使われないか

次に、著者が研究の対象としているグループウェアについて簡単に説明し、グループウェアの研究の中で得られてきた「人に優しいソフトウェア」に関わる知見をいくつか述べる。

文書作成程度の作業ならば十分な性能をもつパーソナルコンピュータが安価に入手できるようになり、パーソナルコンピュータを各人が自分の机の上に専有して使えるようになりつつある。また、計算機ネットワーク技術が進歩し比較的高速なネットワークをこなれた費用で簡単に構築できるようになり、計算機を計算機ネットワークへつなぐのは当たり前になりつつある。

従来のパーソナルコンピュータ環境においては、各利用者は自分専有または何人かで共有するパーソナルコンピュータで、ワープロをする、表計算ソフトを使う、など、個人の仕事をするために利用していた。計算機ネットワークを使うとしても、プリンタを共有する程度であった。

計算機ネットワークをもっと活かして使えないかという発想ででてきたのが、グループウェア、あるいは、CSCW（Computer-Supported Cooperative Work）という概念である。グループウェアの定義は人によってさまざまではあるが、複数の人からなるグループによって行われる協調作業、共同作業を支援するシステム、あるいは、ソフトウェアと言われている [6]。

グループウェアの導入が成功するかどうかの要因を、グループウェア研究の先駆者のひとりであるGrudinが述べている [7]。これらの知見は、グループウェアだけでなく、ソフトウェアやシステム一般についても当てはまる。

Grudinは、グループウェアが、受けられるかどうかは技術的要因よりはむしろ社会的要因であると述べている。その中の主要なものを以下で述べる。

(1) 利益の分配が公平であること

グループウェアは、何らかのメリットがあって導入するわけであるが、システムを導入することによって、何の利益もないのに今までより多くの仕事をしなくてはいけない人が出てくるのがよくある。このようなシステムは、結局使われなくなってしまう。

システムの導入によってやり方を変えるわけなので作業が増えることがあるかもしれないが、その場合は増える作業は最小限に、または、増えた作業に見合うメリットを各利用者にもたらすようになくてはならない。

[失敗例] 会議の時間を自動的に決めるシステム

会議の時間を自動的に決めるためには、会議の参加者が自分のスケジュールをこまめにシステムに入力しなければならないが、利益のあるのは会議の時間を参加者に問い合わせ調整しなければならない人だけなので、全体の利益は大きいですが、これまでは必ず失敗してきた。

[成功例] 電子メール

電子メールは、電子メールを送る人と受け取る人の双方に、使うための手間がかかり、また、受ける利益がある。また、全体として受けるメリットが大きいので広く普及した。

(2) 例外処理を考慮すること

グループウェアは、その時に解決すべき問題を分析して設計を行うわけであるが、これから起こる全てことを予見できるような人間は存在しないし、また、後から思うと何故気が付かなかったのだろうと思うようなことを見落としてしまうことはよくあることである。

例外が生じても何とか回避できるようにしておくのが必要である。

Grudinは、グループウェアの研究、開発において、社会的要因を検討することが重要であると述べているが、これからの応用ソフトウェアはグループウェア的な機能を普通にもつことが予想される。ソフトウェアの研究、開発をすすめる上において、上記にあげた社会的要因を含めて、人間、および、人間同士の関わりを中心とした考察を進めていくことが重要である。

6. まとめ

「人に優しいソフトウェア」を考える上で重要と思われるキーワードをいくつか述べた。はじめにも述べたように、「人に優しいソフトウェア」という言葉は、ソフトウェアを使う人とその人が生きている時代によって、また、使われている技術により異なるものである。ソフトウェアの研究を進める上で「人に優しいソフトウェア」とは何かを常に考えていく必要がある。

参考文献

- [1] ソフトウェア製品の評価－品質特性及びその利用要領－ (JIS X 0129)、および、その附属書A 品質副特性、JISハンドブック 情報処理 (ソフトウェア編)、日本規格協会、pp.82-89
- [2] 鈴木裕信、加藤光明、The BUG、オーム社
- [3] Ed Joyce, *Software Bugs: A Matter of Life and Liability*, *Datamation*, No.33, Vol.10, pp.88-92, May, 1987.
- [4] *Risks to Public in Computers and Related Systems*, *Software Engineering Notes*, Vol.11, No.3, pp.12, Jul., 1986.
- [5] *Risks to Public in Computers and Related Systems*, *Software Engineering Notes*, Vol.16, No.3, pp.24, Jan., 1991
- [6] 石井裕、CSCWとグループウェア、－協創メディアとしてのコンピューター－、オーム社
- [7] Grudin, J., *Perils and Pitfalls; To Succeed with groupware, you must first clear these hurdles*, Dec., 1988, *BYTE*, pp.261-264.

付録

ソフトウェア製品の評価—品質特性及びその利用要領— (JIS X 0129)、および、その附属書A
品質副特性 [1] よりの抜粋 (*は著者による解説)

4. ソフトウェア品質特性

4. 1 機能性

機能の集合の存在及びそれらの明示された性質の存在をもたらす属性の集合。機能は、明示的又は暗示的な必要性を満たすものとする。

*品質副特性として、合目的性 (利用者の目的にあっているか)、正確性 (正しく動くか)、セキュリティ (不当なアクセスを排除できるか) などがある。

4. 2 信頼性

明示された条件の下で、明示された期間、ソフトウェアの達成のレベルを維持するソフトウェアの能力をもたらす属性の集合。

*品質副特性として、成熟性 (故障する頻度がどれくらい少なくなったか)、障害許容性 (どのくらいの障害なら問題なく動くか)、回復性 (故障したときにどれくらい早く復旧できるか) などがある。

4. 3 使用性

明示的又は暗示的な利用者の集合が、使用するために必要とする労力及び個々の使用結果による評価に影響する属性の集合。

*使い勝手のこと。品質副特性として、理解性 (使い方がすぐ分かるか)、習得性 (初めてでもすぐに使えるようになるか)、運用性 (管理するのは楽か) などがある。

4. 4 効率性

明示的な条件の下で、ソフトウェアの達成のレベルと使用する資源の量との間の関係に影響する属性の集合。

*品質副特性として、時間的効率性 (処理は速いか)、資源効率性 (メモリなどの資源を多く必要としないか) がある。

4. 5 保守性

仕様化された改訂を行うために必要な労力に影響する属性の集合。

*品質副特性として、解析性 (プログラムが分かりやすいか)、変更性 (プログラムが変更しやすいか) などがある。

4. 6 移植性

ソフトウェアをある環境から他の環境へ移す際の能力をもたらす属性の集合。

*品質副特性として、設置性 (インストールは簡単か)、環境適応性 (いろんな環境で使えるか) などがある。