Towards 100 Percent Correct Software

Felix C. Gärtner

1 Introduction

It is a well known fact and an everyday experience, that programs of sufficient size cannot be implemented free of errors. This is not only due to the inherent complexity of todays software systems, but also partly the result of inadequate software design and implementation methodologies. Inspired by previous work [2] we introduce the concept of a "Do what I mean button" (DWIMB) as an abstraction of programmers' intentions. We show, that writing correct software is exactly as difficult as implementing such a DWIMB. By doing this, we identify a key component of any large software system that is supposed to be correct.

2 Main Result

Definition 1 (Correct Software) *A correct software system (CSS) is a system that behaves exactly as wished by the user.*

Definition 2 (Do what I mean button) *A "do what I mean" button (DWIMB) is a module of a software system, that – if started – forces the system to do the thing that the user wanted it to do.*

Theorem 1 $CSS \Leftrightarrow DWIMB$

Proof. $\Rightarrow$: Let s be a CSS. Now "disguise" s by hiding the system behind the interface of a DWIMB b and let s only run, when b is pressed. If pressed and by Definition 1, s subsequently behaves exactly as wished by the user. If the rest of the surrounding system is inhibited from taking any steps, the system therefore does what the user wanted it to do. By Definition 2, b is a DWIMB.

$\Leftarrow$: Let b be a DWIMB and s be a (possibly incorrect) software system. By Definition 2, b forces the system to do what the user wishes, if it is started. Construct the user interface of s in such a way, that it contains only one button, and invoke b whenever the user asks for action. Thus, the system always behaves as expected and by Definition 1 s is a DSS. This concludes the proof.

3 Conclusion

We have shown that writing correct software is exactly as difficult as implementing a construct called the "Do what I mean button." The main contribution of this result resides in the potential of great simplification of today's software engineering disciplin: The problem of writing correct software is reduced to the implementation of a single button. Such a button can be placed easily on today's keyboards, on mobile phones or devices as small as banking cards. Because of these prospects, we think that researchers in the area of computer science should drop everything they are doing and concentrate on this key problem, which offers a huge potential in user friendliness, correctness and research funds.

References

[1] Frederick P. Brooks. No silver bullet. *IEEE Computer*, 20(4):10–19, April 1987.

[2] Volker Trumpf. Codename 'Gizeh' - Erste Vorabversion von Windows 97. *c't*, (4):58–59, 1996.