DOI 10.1007/s00165-008-0101-8
BCS © 2009
Formal Aspects of Computing (2009) 21: 571–588

Formal Aspects
of Computing

Model-checking user behaviour using interacting components

Thomas Anung Basuki, Antonio Cerone, Andreas Griesmayer, Rudolf Schlatte
International Institute of Software Technology, United Nations University, Macau SAR, China.
E-mail: anung@iist.unu.edu

Abstract. This article describes a framework to formally model and analyse human behaviour. This is shown by a simple case study of a chocolate vending machine, which represents many aspects of human behaviour. The case study is modelled and analysed using the Maude rewrite system. This work extends a previous work by Basuki which attempts to model interactions between human and machine and analyse the possibility of errors occurring in the interactions. By redesigning the interface, it can be shown that certain kinds of error can be avoided for some users. This article overcomes the limitation of Basuki's approach by incorporating many aspects of user behaviour into a single user model, and introduces a more natural approach to model human–computer interaction.

Keywords: Human–computer interaction; Interacting components model; Rewrite systems; Model-checking

1. Introduction

Formal methods and user-centred design are two alternative methodologies aiming at reducing the likelihood of system failure. In user-centred design analysing and foreseeing user's behaviour in interacting with the computer, as well as possible human errors, is a fundamental concern of designers, who also have to test the validity of their assumptions in the real world, with actual users. Formal methods have been originally applied to system design and analysis with the perspective that human errors are outside their scope.

However, this perspective started gradually to change during the 1980s. Firstly, more and more importance is given to the formal analysis of user interfaces. Chi [Chi85] compares four algebraic techniques in the analysis of a commercial user interface, manually proving a number of not trivial properties of the interface, which however do not involve the interaction with the user. Secondly, it is understood that a user model should be included in the design process, separately from the model of the interface. Cognitive complexity theory (CCT) [KP85] clearly separates the description of the user's goals from the description of the device with which the user interacts. Another interesting approach is the Executable Cognitive Architecture developed by Newell, Laird and Rosembloom, which was then implemented in the SOAR system [LNR87]. This cognitive architecture inspired the definition of the programmable user model (PUM) defined by Young, Green and Simon [YGS89], which envisage an executable model of the user within a psychologically constrained architecture which allows a predictive evaluation of the interface design.

In the 1990s, the catastrophic consequences of human errors experienced in safety-critical systems, including for example plant/process control, traffic control (air, road, rail, sea), medical devices and defence, gave a

Correspondence and offprint requests to: T. A. Basuki, Email: anung@iist.unu.edu