

To be decided Measuring Interaction before developing prototypes Measuring Interaction Graphs

1st Author Name
Affiliation
City, Country
e-mail address

2nd Author Name
Affiliation
City, Country
e-mail address

3rd Author Name
Affiliation
City, Country
e-mail address

ABSTRACT

Early development and testing of prototypes is good practice for user interface development. However, prototypes have to cover specific usage scenarios, and because they are limited in focus, the whole picture of the user interface is easily lost. Even simple questions dealing with numerosity and length of optimal execution paths or impact of possible user errors can be answered only for the specific scenarios being analysed.

We discuss a tool that transforms models of the behaviour of a user interface into a graph. This is then used to specify usage scenarios, and to generate possible execution traces. Metrics based on number and length of different possible execution paths, with or without possible mistakes, can be easily computed. We apply the tool and metrics to well known examples of web mail apps, and show that important conclusions can be drawn even before the first prototype is built. — replace with a real example statement.

Author Keywords

Experimental; Evaluation; Statecharts: UML; UML-IDEA; Testing.

ACM Classification Keywords

H.5.1 Information interfaces and presentation (e.g., HCI): Multimedia Information Systems.; H.5.2 Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION

We present an approach that allows a designer to quickly (a) understand how prone an application is to user navigation errors; (b) understand how recoverable the application is from those errors; (c) support task analysis and task / scenario design; (d) support interaction design (IxD) in terms of consistency, error-proneness, and recovery premature commitment; and further, allows different designs to be objectively compared to support designers in evaluating interaction sequences. To explain our work we present a case study comparing four different web mail front ends. We have chosen

this domain because it is very well understood by readers and yet those trivial questions lead to non trivial results.

The approach is based on UML state-chart models of the user interfaces which are automatically processed to produce *interaction graphs*. These are then used to unfold *execution traces* that are dependent on the specific usage scenarios being considered in the analysis. On traces several graph-theoretic computations can be performed to produce a dashboard of different results that provide the answers. Except for development of models and specification of the desired scenarios, the other steps are totally automatic; models of a design could be developed in a matter of a couple of hours.

Developing good user interfaces for web or mobile applications is a complex and expensive endeavour. One reason is the combination of devices, interaction modalities and workflows that need to be supported.

Adopting Usage Centred Development practices is effective, as is following established design principles [?]. Early prototyping [?] to explore part of the five-dimensional prototyping space [?] is one of the most effective techniques, especially when paired with usability investigations based on user testing or heuristic evaluations.

However, it still requires development of prototypes that are usually developed with certain tasks in mind, and therefore are quite restricted in terms of depth and breadth of supported use cases. Furthermore, usability results are always surrounded by a cloud of uncertainty, due to subjectivity introduced by participants and facilitators or by other contingency factors involved in the analysis. Thus, although an effort needs to be expended to develop and use prototypes, less than optimal results are obtained.

A designer, while conceiving and developing a solution, might benefit from answers to seemingly simple questions that should not require a significant investment of time and effort. Given one or more potential solutions and some usage scenarios, interesting questions could include: “What are the optimal execution paths?” or “How many such paths exist?”, or “Are usage mistakes recoverable?” – these seem to be software ‘executionist’ and not so much user/designer is – maybe we can say something more designer and save these to the technical bit of the paper?. At the moment, even these straightforward questions are quite complex to answer. In fact, they require inspection of prototypes, man-

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single spaced.

Every submission will be assigned their own unique DOI string to be included here.

ual tracking of which screens and widgets are used at which stage, and exhaustive searches.

This should not be the case, however. These answers provide important insights to a designer, and support decisions related to benchmarking different solutions, to identification of optimal or mistaken paths, to assessment of suitability of a design with respect to scenarios.

The contribution consists of the development of a tool that transforms models and scenario specifications into execution traces, and the definition of metrics that provide concise, precise and objective measures of a design.