

# Design-Side Considerations: A Reaction to DUEM

Robert B.K. Brown  
School of Information Technology and Computer Science  
University of Wollongong, NSW 2522 Australia  
bobbrown@uow.edu.au

## Abstract

*Evaluation of a system's usability is a difficult task, but a new method, DUEM, addresses several of the issues. Any evaluation conducted after the fact however has limited effect, so further work is required at the design end of the lifecycle. Early-Phase elicitation tools, such as i\*, may form the basis for new design tools which simplify software development, improve usability and ensure greater system success. Informing such design tools with the principles of the latest Activity Theory-based usability evaluation methods, such as DUEM should facilitate easier testing. As an added benefit, such a design method can form the first stage of an entire software development process where these complex evaluations serve to verify and validate the product.*

## Keywords:

Usability testing, validation and verification, Activity Theory, DUEM, interaction, early-phase, elicitation.

## 1. Introduction

Software engineers are recognising that inadequate definition of requirements especially 'softer' socio-political ones, leads to increased likelihood of failure [1]. This highlights the need to elicit stakeholder requirements and identify their softer, qualitative objectives. Evaluating the usability of any given system is a non-trivial exercise and no universal method exists. Vrazalic [3] has proposed a Distributed Usability Evaluation Method (DUEM) that shows considerable promise. Evaluating a system after its release however has no effect during construction. There is at best, assuming largely the same team and similar requirements, a generational benefit whereby the experience from one system's problems informs the building of the next.

Some practical difficulties have been identified in deploying DUEM, it is expensive, cumbersome and time-consuming. Combined with the limited value of end-phase evaluation, it may of greater value to build the aspects DUEM tests into requirements in the design phase. DUEM can then become a process of validation and verification of a larger software production process which embraces the same principles.

This paper proposes that a design method can be produced which is informed by those qualities which can identify high usability after the fact, and deploy them from the earliest phases of design. A streamlining and remediation of the software lifecycle for highly interactive systems is thus envisioned by way of a complete software production process, specifically built to enhance system usability.

## 2. The Interaction Quotient (IQ)

Computerised systems do not all exhibit the same degree of reliance upon dialogue exchange with a user. Some systems operate almost entirely separately from any human agent. The computer-moderated operation of a fuel-injection system in a motorcar is such a system – the driver may remain quite unaware of its presence. Other systems function in a batch-processing mode where the systems produce the output independently of any user, once initial conditions and commands have been set. One may consider the monthly processing of salary details for a large company to fall into such a category. An increasing number of systems however, seemingly exist purely for the direct use of users and cease to function in any meaningful way without user interaction. A web-based e-commerce system would exemplify such a system.

To approach a metric for classifying systems with respect to these differences, this paper introduces the term Interaction Quotient (IQ), provisionally defined as the proportion of a system's functionality, which directly interacts with the User. In later work a more rigorous and quantitative definition will be offered, but this paper will refer to systems as being high-IQ or low-IQ, to indicate the dominance of user interaction in their core functionality.

A greater number of high-IQ systems will be encountered over time as the World Wide Web become the medium for widespread use of systems designed for use by a heterogeneous audience. A market for high-IQ systems exists in which we cannot assume any users' prior computer skills. For systems to be commercially successful in this broader market, ease of use becomes a crucial factor.

### **3. The Distributed User Evaluation Method (DUEM)**

Vrazalic [3] proposes in her PhD thesis a method for evaluating the usability of a system after its completion. This rich technique is informed by Cultural Historical Activity Theory (CHAT) and thus is concerned with the broader social context in which the system is used. The user, her social environment, the system and all mediating technologies must be considered. Vrazalic adopts a wider definition of usability in the manner of Spinuzzi [4] that incorporates assorted genres, practices, uses and goals. Under this notion of distributed usability, Vrazalic considers the typical usability laboratory to be an artificial environment that has a number of shortcomings that can skew the results [5]. DUEM thus involves an involved and comprehensive series of tests based upon Activity Scenarios that are generated from intensive observation of the user in their native work context, interviews with users and moderated focus group discussions.

DUEM consists of three interacting phases: Firstly, understanding user activities, then evaluating the role of the system in user activities, and finally analysing and interpreting the results. The first phase produces a shared understanding of user tasks and goals. The second phase produces rich qualitative descriptions of the users' interaction with the system. The third phase concentrates on identifying points of breakdown, where the system and the activity map contradict. The problems identified have a deeply contextual definition that aids in any negotiated solutions. DUEM uses the notion of distributed usability and CHAT principles to define contexts of a system's use by humans. Evaluation is adjudged against criteria derived from these initial findings, based upon user activity rather than system specific requirements. Users are deeply involved in an iterative process through interviews, workshops and observations. Evaluators must have an understanding of CHAT principles to inform their analysis and to help them guide users through the process. [3]

The evaluations that result from DUEM show considerable promise however the wide scale deployment of the method may be inhibited by several factors; namely that the method requires trained evaluators, occupies a great deal of time (and is thus quite expensive) and requires that intended users be available at the time [6]. We must also consider that any usability evaluation method that is conducted after the fact can at best only indicate the quality or otherwise of the finished product or facilitate a late stage repair. It is questionable that any meaningful savings in the software production lifecycle will result, as each test is case-specific. If such an evaluation method is widely used, we might expect to see a slow generational or evolutionary improvement in the usability of systems, as each new product is evaluated and its success measured.

This paper proposes that remediation of the software production process may be best achieved by modifying the design-side of the process. DUEM offers some crucial elements to be addressed in this design process, any resulting method would not only benefit from the principles of DUEM, but make DUEM itself easier and cheaper to apply. DUEM is based on user-system interactions rather than system requirements, because these requirements do not address such issues. If they did however, DUEM need not be such an end-phase-heavy process.

### **2. Early Phase Requirement Elicitation with $i^*$**

If enhanced usability of a system is a core requirement, as it is for high-IQ systems, then it is more likely that considering the necessary factors at the time of design will entail success. Stakeholders generate their own notations and terminologies, complicating the business of capturing such details [7]. This difficulty informs approaches to requirement elicitation that are sensitive to consistency and viewpoint. It is necessary therefore to use a systematic approach when capturing requirements [8]. A significant risk of failure exists in marginalizing the stakeholder's softer objectives, despite their inherent messiness. One approach to identifying stakeholder preferences adopts a goal-orientation and asks 'what does the stakeholder want to achieve'. Goal formulations express intended system properties [9]. For high-IQ systems, the analyst will need to concentrate on issues of usability and identify these from stakeholder utterances.

DUEM requires a shared understanding of system use and tasks. A desirable early phase requirements elicitation tool would be one that facilitates reaching just such an understanding. A semi-formal notational tool with the flexibility to accommodate multiple stakeholders' utterances would compliment and enhance the process. One such toolset can be found in Eric Yu's  $i^*$  method. Initially conceived as a business-process engineering tool,  $i^*$

(which stands for “distributed intensionality” [*sic.*]) frames processes as social activities between actors who depend upon each other for goals to be obtained and tasks performed [10].

As a notation, *i\** incorporates two main diagrammatic tools, the Strategic Dependency model (SD) and the Strategic Rationale model (SR). In examining a given activity, SD’s deal with several entity types: the Actor entity represents an agent capable of decision. Actors are linked by their dependency upon other actors to furnish Resources, perform Tasks and satisfy Goals. Qualitative Softgoals, such as “the system shall be user friendly” cannot be formally satisfied. As identified by Herbert Simon in his seminal work *Bounded Rationality* [13], such Softgoals can at best be *satisficed* (sufficiently satisfied). *i\** dependency links can indicate the degree to which one actor contributes positively or negatively towards satisfying Softgoals.

The SR model, which is drawn as a dashed ovoid, extending from the actor entity node explores actor rationale. Within the SR are defined links not unlike those in the SD, however the linkages are more specific. Tasks are decomposed via Task-Decomposition links indicating an ordinal ranking. Goals may be linked to other entities with Means-End links indicating tasks or resources required for satisfaction. Softgoal have Means-Ends links that are qualified with an indicator of the degree to which the linked entity contributes towards satisficing.

*i\** is not yet a complete design tool and there are some elements of its conception which appear incomplete at this time. It is not clear if Yu’s use of the term distributed in any way reflects the sense in which Spinuzzi uses it. Rather, it is likely that Yu refers to the distribution of intentions among a number of disparate stakeholders. It currently remains unclear if Yu referred to the intentionality of the actors within his modelled tasks (being presumably a set of their agendas), or the intensional definition of their dependency-relationships, as he has used the two terms somewhat interchangeably over time. Intension is a more subtle and precise philosophical term, which sits opposite extension. The intension of a term in a language leads to a larger set of decomposed internal attributes and defining characteristics. Extension however generates fewer terms under what is essentially a process of abstraction. Intension is to connotation as extension is to denotation. Whilst this observed discrepancy will be more deeply analysed in later works, the current paper presents this anomaly simply as further evidence of a possible lack of rigour in the *i\** method and its conception.

The *i\** method, whilst imprecise in its current form, has the useful quality of capturing qualitative stakeholders preferences as Soft-Goals [11]. Many identified aspects of usability may be considered as Non-Functional Requirements and, as such, qualitative traits. A great deal of research is dedicated to identifying software errors based on satisfaction of requirements, however such systems typically require that requirements be reduced to digitally satisfiable states [12].

Unfortunately, *i\** seems to be under equipped to deal with a number of issues. It seems necessary to extend the method to incorporate notions of consistency both in terms of actor role and in terms of organisational hierarchy [5]. Further extensions to *i\** may be necessary to incorporate temporality. The *i\** tool may however prove to be an excellent candidate as the basis of a new design tool. It’s ability to capture social relations of the user and system in a wider context may be complimentary with the paradigm of the AT-flavoured DUEM end-phase tests.

## 5. A New Design Approach

With the introduction of quasi-automated software generation tools based on state diagrams [14] and other concepts, the lifecycle of software production is contracting. Increased emphasis is being placed on either end of the process, both requirements specification and verification and validation. There is considerable value in extending requirements capture into a design paradigm, addressing in particular the qualitative issues of usability for high-IQ systems. Whilst requirements capture currently serves as an *entry* stage to the production cycle, it could encapsulate design and become integral to the production cycle.

Similarly, evaluation, validation and verification occupy the end stage. Since DUEM provides us with a suite of contextual usability evaluation checks, it is logical that complimentary terminology be employed at the outset. The extensive interviews, observations and focus groups of DUEM can thus be conducted as a component of early phase stakeholder requirement elicitation, thus simultaneously establishing requirements the criteria by which they are checked. Thus the pitfalls of DUEM may be minimised. In this manner, DUEM becomes part of a larger paradigm.

As the central processes of the production lifecycle contract, and soft goal requirements such as usability increase in importance with the rise of high-IQ systems, we may expect a usability centric design-side paradigm to be a requirement for the efficient and successful production of usable software systems.

## 6. Conclusion

Systems power and commercial success are strongly linked to notions of usability, especially for highly interactive (high-IQ) systems. The assessing of usability has been at best a variable process and so little consistent and comparable feedback is evident in the design stages of the software lifecycle.

The new Distributed Usability Evaluation Method (DUEM) promises a more meaningful measure of usability, which may yield results from which trends and principles for enhanced usability may emerge over time. Informed by Activity Theory, DUEM strives to evaluate usability across numerous hand-crafted activity scenarios that take account of the users social environment and all mediating technologies. DUEM unfortunately is expected to be somewhat cumbersome and expensive to deploy and at best will yield a measure of a system's usability *after the fact*.

Early-phase stakeholder requirements capture tools, such as *i\**, may serve as a solid basis on which to construct a usability-centric design method. If such a method were framed in terms complimentary to the Activity Theory approach of DUEM, then the process of evaluation can be considerably streamlined as the activity scenarios can be identified at the time of requirement elicitation, and the evaluation metrics established.

## 7. References

- [1] Goguen, J.A. and C. Linde, "Techniques for requirements elicitation", *Proceedings of the IEEE international symposium on requirements engineering*, 4-6 Jan. 1993, pp. 152-164.
- [2] Papert, S. "Uses of Technology to Enhance Education", MIT *Artificial Intelligence Memo* No.298, 1973
- [3] Vrazalic, L., *PhD Thesis* School of Information Systems, University of Wollongong, 2004
- [4] Spinuzzi, C., "Grappling with distributed usability: A cultural-historical examination of documentation genres over four decades". *Proceedings of the 17<sup>th</sup> annual international conference on Computer Documentation*, 12-14 Sept. 1999, New Orleans, USA.
- [5] Vrazilic, L., Gould, E., "Towards and Activity-based Usability Evaluation Methodology". *Proceedings of the 24th Information Systems Research Seminar in Scandinavia*, Ulvik, Norway 2003
- [6] Vrazalic, L., Hyland, P. "Towards an Activity Scenario Based Methodology for Usability Testing of Websites" *Proceedings of HF 2002 Human Factors Conference*, November 25-27, Melbourne 2004
- [7] Sommerville, I., P. Sawyer, and S. Viller, "Viewpoints for requirements elicitation: a practical approach", *Proceedings of the 3rd IEEE international conference on requirements engineering (ICRE '98)*, Colorado Springs, USA, 6-10 Apr. 1998, pp.74 – 81.
- [8] Brown, R.B.K., Ghose, A.K., "Hierarchic Decomposition in Agent-Oriented Conceptual Modeling", *Proceedings of the 4<sup>th</sup> International Conference on Quality Software (QSIC'04)*, Braunschweig, Germany.
- [9] Lamsweerde, A.van., "Goal-oriented Requirements Engineering: a guided tour", *Proceedings of the 5th IEEE international conference on requirements engineering (ICRE '01)*, Toronto, Canada. 27 –31 Aug. 2001, pp. 249 – 262.
- [10] Yu, E.S.K., J. Mylopoulos and Y. Lesperance, "AI models for business process reengineering", *IEEE Expert*, 11(4), Aug. 1996, pp.16-23.
- [11] Yu, E.S.K., Strategic Modelling for Enterprise Integration, *Proceedings of the 14th World Congress International Federation of Automatic Control*, Beijing, China, 1999.
- [12] Yu, Y.T, Lau, M.F., "Comparing Several Coverage Criteria for Detecting Faults in Logical Decisions" *Proceedings of the 4<sup>th</sup> International Conference on Quality Software (QSIC'04)*, Braunschweig, Germany.
- [13] Simon, H.A., "Theories of Bounded Rationality", in *Decisions and Organisations*, Radner and Radner Ed., 1972
- [14] Harel, D., Kugler, H., Pnueli, A., "Synthesizing state-based object systems from LSC specifications" *International Journal of Foundations of Computer Science (IJFCS)* 13(1):5-51 Feb 2002

## **COPYRIGHT**

The following copyright statement with appropriate authors' names must be included at the end of the paper

[Author/s names] © 2004. The authors assign to OZCHI and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to OZCHI to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.