

Software Design and Media Design - tools to design Mobile and Sensory Interfaces and Interactive Environments

Geert de Haan

Communication, Media and Information Technology
Section Media Technology / Human Centered ICT
Rotterdam University of Applied Sciences
P.O.Box 25035, 3001 HA, Rotterdam
The Netherlands
g.de.haan@hr.nl

ABSTRACT

This paper discusses ETAG, a formal model for design representation, and ETAG-based design, a method for user interface design, and the two principal facilities they provide to accommodate different types of users in the design of user interfaces. The paper starts with an introduction of ETAG as a design representation. This is followed by a description of ETAG-based design and using the notation to represent relevant aspects of the work context. Next, we discuss the differences between computer software design and media product design, concluding that media design is a much more flexible, iterative process and prototyping-based process in which adaptation of the design of mobile applications extends into the maintenance phase. To cover further developments towards focussing on user needs and wishes by means of co-design practices, and to cover for ubiquitous computing and interaction with sensors and interactive environments, we propose to use sensory labs and to create living labs to move the usability lab into the real world.

Keywords

formal modelling, ETAG, software design, media design, design tools,

Extended Task-Action Grammar

ETAG (Extended Task-Action Grammar; [3][8]) is a formal language to represent user interfaces in terms of the knowledge that a perfectly knowing user would have (in a mental model) about performing tasks. To create a psychologically valid description of user interface for design purposes, ETAG stratifies user interface knowledge into a number of levels using existential logic and written down in a formal grammar. The formal grammatical notation, adopted from ETAG's predecessor, Task-Action Grammar (TAG; [6]) ensures that the description is sufficiently precise for design and implementation purposes without sacrificing psychological validity. User interface representations are stratified into levels to meet the existence of levels in human knowledge and to reflect the major decisions that occur during the design process.

ETAG representations consists of a canonical basis, a user virtual machine, a dictionary of basic tasks, and a section with production rules.

The canonical basis lists the universally known concepts such as object, attributes and events which are used to define the specific objects, attributes, etc. of the user interface in the type specification and the type hierarchy of the user virtual machine. The user virtual machine describes the elements and the workings of the user interface without referring to a specific implementation. The dictionary of basic tasks lists the tasks which are available to the user and it links the workings of the user interface to the command specification of the tasks. Finally, the production rules describe, for each task, the command procedure in terms of the command syntax, the way of referring to command elements, the naming and labelling of command elements, and the physical actions to specify each element.

ETAG-Based User Interface Design

In ETAG-Based Design [3] user interface design is regarded as the incremental specification of the mental model of a perfectly knowing user. The design process is structured into a number of discrete steps, each covering a specific set of design decisions: task- and context analysis, task design or task synthesis, conceptual user interface design, and perceptual user interface design, which consists of the design of the presentation interface and the design of the interaction language between the user and the system.

In ETAG-Based Design the ETAG notation is used to represent the analysis and design results. To this purpose, it is necessary that the notation is flexibly adapted to meet the specific purposes of the design stage. For example, in modelling business procedures during task analysis, the representation is specified at a high level of abstraction without much detail, and special concepts are used to represent the decomposition of tasks and procedures and to represent agency and ownership. An advantage of using a single notation throughout design is facilitating the transitions from one design stage to the next one. Another

advantage, at least in principle, is that it is easier to create tool support. This is particularly relevant for presenting ETAG as a formal model in a way which suits the background and the way of working of other stakeholders. Instead of having to translate between many different notations, in ETAG-based design, tools only need to deal with a single notation which allows for easy to automatic translation into visual specifications or programming code. A final advantage of using a single notation is that designers themselves are not required to learn and use a variety of different notations.

ETAG was originally proposed as a competence model of the knowledge that users need to perform their tasks, much like Payne and Green's Tag Action Grammar [6]) with the special addition of an ontology so that not only the translation of tasks into actions can be described but also the objects and attributes and the transitions that take place when tasks are executed. In other words, whereas TAG is able to describe what users need to know about how to delete a file, ETAG is able to describe what a file is and what happens when it is deleted.

ETAG as a formal notation is considerable harder to create than a non-formal description but it seemed worthwhile since formalisms may also be used for other purposes than mere design specification, including automatic generation of online help, generation of user interface programming specifications and code, easy calculation of usability characteristics like consistency, complexity and learnability, and the aforementioned use of a single notation throughout the whole design process (see: [3]). Not bad for a notation that is also psychologically valid, even though, at the time, the question was raised if such advantages could really counter the difficulties associated with formal representations: this might be the case for, for instance, large, dependable or safety critical systems but what about the average windows utility?

Designing New Media

New Media of digital media refers to all forms of purposive information transfer, carried by digital means. Compared to analogue information carriers such as gramophone records, newspapers and paintings, digital media like webpages and digital video, in combination with declining hardware costs and more effective tools, allow for (almost) effortless copying and editing. As a consequence of such lack of resistance to change, the design process does not have to meet the same rigour of the design process. A similar transition has taken place with every new generation of computer hardware, from the mainframe to the mini-computer, to the personal and game computer, and finally, to the smartphone and ubiquitous computing [4].

In comparison to the design of software systems for business processes or pay rolling, the design process of new media products like interactive websites and mobile

apps is lightweight, where flexibility with respect to adapting to changes in the market or the customers' wishes is a key requirement to the design and the design process. Consequently, particularly in the media area, agile design methods like Scrum [7] and Extreme Programming (XP). we did a small preliminary study into design methods utilised by the companies, we asked our media-technology students who served as their interns to name the type of products and the design methods used. The results indicate that virtually all of these companies either utilise scrum or comparable methods, or they are in the process of moving from waterfall-like methods to agile design.

Because of the flexibility requirements and the lower repair costs in designing media products, the design process is not only lightweight but also tends to consist of a variety of tools to suit the job without much reliance on the design notation. The following list contains about all user interface design representations employed in a modern media-technology curriculum:

- persona's and mood boards
- a design concept and view (generally, in text)
- task lists, task descriptions and task analysis models
- usage scenario's and storyboards
- use cases, activity diagrams, entity relationship diagrams
- interface sketches, wireframes and screen designs
- prototypes, demonstrators and the actual working system

The design process of media products is based on prototypes, from low to increasingly higher fidelity prototypes, it is a features driven design, where each design cycle or Scrum sprint focuses on the next most important features to implement, and, finally, an incremental design process with iteration both during and after the design process where maintenance is regarded as including further adaptation to evolving user wishes and tastes.

Media products tend to allow for much flexibility because of the distinction between the 'frontend', the website or user interface of the system and the 'backend', the database(s) and the Content Management Systems (CMS) which act as the user interface of the application programmer. The strict separation of the user interface and the data processing part of the application allows for easy adaptation of the front-end whilst keeping the backend stable. While a website is up and running, it is possible to present different groups of users with a different frontend, depending for example, on the basis of the local webserver they use. Next, data collected online about user preferences, conversion rate or sales figures may be used to choose the most successful frontend design. Naturally, such a process of online optimization is not restricted to a single trial but may take the form of a continuous process of adapting the looks and behaviour of a website or mobile app to the behaviour of its users.

Broos et al. [2] noted that, for instance, consistency is a positive characteristic of user interfaces according to HCI theory but that in designing mobile applications other requirements may be more important. For instance, in designing a mobile social app for the skating community (board skaters, inline skaters and bmx bicyclists) it is natural to make a distinction between tasks when users are actually mobile (hence: actively skating) and those executed when the user is able to pick a steady seat to interact with the application. In the former -mobile- case, interaction should be foremost automatic and minimal, utilizing sensor information like GPS location recognition instead of the demanding the user to indicate his or her choices on a keyboard or touch screen [2].

Comparing Design Approaches

When comparing the two design approaches listed, one heavyweight and resting on a design notations as the core of the design process, and one lightweight and utilizing whatever tools seems most appropriate to the design cycle at hand, it will be clear that the media design approach is much more flexible and less regulated and thereby better able to rapidly service any changes in customer wishes and needs, exactly as Schwaber and Beedle [7] tried to address with Scrum.

In designing web applications the media design approach works fine. However, with the ongoing transition towards mobile computing, sentient interfaces and ubiquitous computing, it is our opinion that the iterative feature driven design process has to be further adapted to the new design ecology.

First, increasing focus on user requirements and wishes has increased the employment of co-design and co-creation practices. As a consequence, application design takes place behind the software engineer's work station and perhaps in the usability lab but increasingly often design 'happens' within the actual context of use. Secondly, computer applications increasingly make use of sensors in either the computer device, in the environment or both. Our experience with such interactive environments is limited and, as such, requires the integration of investigating usage and usability aspects in the real world. This provides another argument to remove the distance between the design and the application contexts.

On the basis of the utility of complex formal tools like ETAG in our work on media design, we do not opt for the introduction of new and complex tools to visualise or automate aspects of our design activities; rather, we opt to move design more into the direction of the actual context of use and away from the workstation room [5].

First, we are about to introduce the sensor lab as a middle-ground between the usability lab and the real world. The sensor lab provides all the facilities for the first crude

design iterations, including a range of pre-installed networked sensors and interactive display screens, observation camera's and microphones. In this manner we are able to experiment with and investigate the use of sensors in an environments which also provides for usability lab facilities. Later on, we plan to introduce a fablab to extend the design facilities towards interactive objects in general rather than smart phones and other pre-designed interactive objects. Secondly, we investigate the use of self-configuring sensor networks like Almende's 'sense-os' [1]. Networks like these make it possible to hook-up ones mobile phone to a network and to collect on-line sensor and usage data from the phone or other networked sensor devices thus enabling a so-called living lab which acts as a usability lab within the everyday real world environment.

REFERENCES

1. Almende (2011). Observation Systems. Available at: <http://www.sense-os.nl/>
2. Broos, M., van Gammeren, P., van Steenoven, T. en de Haan, G. (2011). Creating a context-aware mobile application to enlarge social cohesion: skating together. Accepted for ECCE 2011 - Designing Collaborative Activities, Aug. 24 - 26, 2011; Rostock, Germany.
3. de Haan, G. (1996). ETAG-based Design: User Interface Design as Mental Model Specification. In: Palanque, P. and Benyon, D. (eds.) Critical Issues in User Interface Systems Engineering. Springer Verlag, London, 81-92.
4. de Haan, G. (2011). DevThis: HCI Education beyond Usability Evaluation In: Lenior, D., Sturm, J. & Mulder, I. (eds.) Proceedings Chi Sparks, 23 June 2011, Arnhem, the Netherlands.
5. de Haan, G., Choenni, S., Mulder, I., Kalidien, S., & van Waart, P. (2011). Bringing the Research Lab into Everyday Life: Exploiting Sensitive Environments to Acquire Data for Social Research. In: S.N. Hesse-Biber (ed.) The Handbook of Emergent Technologies in Social Research (Chapter 23). New York: Oxford University Press, pp. 522-541.
6. Payne, S.J. and Green, T.R.G. (1986). Task-Action Grammars: a model of the mental representation of task languages. *Human-Computer Interaction* 2(2), 93-133.
7. Schwaber, K., and Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice Hall.
8. Tauber, M.J. (1990). ETAG: Extended Task Action Grammar: a language for the description of the user's task language. *Proceedings Interact' 90*, pp. 163-168. North-Holland.