Conflict, Costs and Trade-Offs in User Interface Design

Alistair Sutcliffe(✉)

Manchester Business School, University of Manchester, Booth Street West, Manchester M15 6PB, UK
a.g.sutcliffe@manchester.ac.uk

Abstract. A framework for cost benefit analysis to inform design choices in configuration and customisation of user interfaces is presented. User costs of learning, UI complexity and operational usability are traded against benefits from better functional fit for the users needs. The design process is illustrated with case study experience from an eHealth application involving trade offs between conflicting requirements from two groups of users.

Keywords: Configuration · Customisation · Trade-offs · Cost-benefit analysis

1 Introduction

The conflict between goals, needs and requirements from different stakeholders has received considerable attention in the Requirements Engineering (RE) community, where the conventional response has been to negotiate the conflicts to arrive at a common viewpoint (Sommerville and Kotonya 1998; Robertson and Robertson 1999). Goal modelling (Mylopoulos et al. 1999; van Lamsweerede 2009) can make conflicts explicit, thereby supporting the negotiation process; however, resolution of conflicting requirements inevitably leads to compromises by some users. User interface (UI) properties, usually referred to as non-functional requirements in RE are a sub-set of the more general problem; for instance, the clash between usability, privacy and security in passwords is a well known design dilemma (Braz et al. 2007).

In HCI the requirements conflict-resolution process is an essential component of user-centred design (UCD) (Sutcliffe 2002a). However, different user needs might be accommodated by different versions of the user interface, via a process of configuration or personalisation. While surface personalisation of UI features such as menu toolbars, display colours and layouts, and message terseness/verbosity, are standard components of all major operating systems, resolution of deeper functional differences between users is more problematic. Offering users choice of UI/application versions by configuration facilities imposes a cost on users when operating the configuration user interface, and most users accept the default version. The design dilemma is how to fit the requirements of diverse user groups while minimising the configuration cost and maximising the functional fit of the application to users' needs.

This paper reports experiences in resolving requirements conflicts in user interfaces, approached through examining users' needs at a more fundamental level in the

T. Clemmensen et al. (Eds.): INTERACT 2017, LNCS 10774, pp. 71–82, 2018.
https://doi.org/10.1007/978-3-319-92081-8_7

form of their values. Values have been explored in value-sensitive design (Friedman 2008) and the related concept of worth can help to frame users' viewpoints as worth maps (Cockton et al. 2009). In Value-Based Requirements Engineering VBRE (Thew and Sutcliffe 2017), users' values are made explicit by analysis with a reference taxonomy of values, motivations and potential emotional reactions. Making users' values explicit provides a richer context for negotiation and resolution of conflicts. The VBRE method has been applied to two case studies in health informatics. This paper first presents a framework for considering design trade offs for conflicting requirements and then describes the experience and lessons learned from the ADVISES and SAMS projects; which contribute to a discussion about different approaches and implications for conflicting requirements. UI design responses to conflicting requirements has been described by a variety of terms, e.g. personalization, customization, localization, configuration, adaptation. In this paper I use the term 'configuration' to refer to choice between alternative functions, which is close to the sense of adaptation of the user interface (Fischer 2001b). Design of user interface features, such a colour, font size, menu options, which are more commonly referred to as personalization/customization are not with my remit.

2 A Trade-Off Framework for Assessing Conflict

The design response to conflicting requirements will depend on the number of user roles and the conflicting requirements demanded by those roles. As more requirements are added to the design, complexity increases, consequently the first design trade off is to consider a graded response from a single, complex product including a variety of functionality to meet each user's requirements, to separate product versions matched to each user role. A supplementary consideration is whether to provide configuration/customization facilities to enable the user to tailor a complex product to their own needs or to design separate product versions. These choices impose different costs on users and developers, summarized in Fig. 1. User costs are composed of learning to use the product and the configuration interface (if present), and then operating the configuration. Developer costs are the consequence of additional software development for versions and configuration facilities.

Benefits accrue when the product exactly fits the user's requirements while minimizing developer and user costs. Clearly many product versions tailored exactly to individual users is the ideal, but this usually imposes prohibitive costs. The UI designer has to strive for a 'sweet spot' where development costs are constrained, while maximizing the user- system task fit, and minimizing user costs from learning and operating the UI. A further trade off emerges between increasing product complexity, which imposes learning and operating costs for the user to search and find the functions they require, in contrast to configuration facilities which enable users to select their required functions and hence reduce complexity of the operational UI. However learning and operating the configuration UI is an imposed cost.

There are further complexities the design trade off space that arise from the progression from simple configuration (i.e. by a selection menu at product set up time) to complex End User Development facilities that maximize user choice with increasing

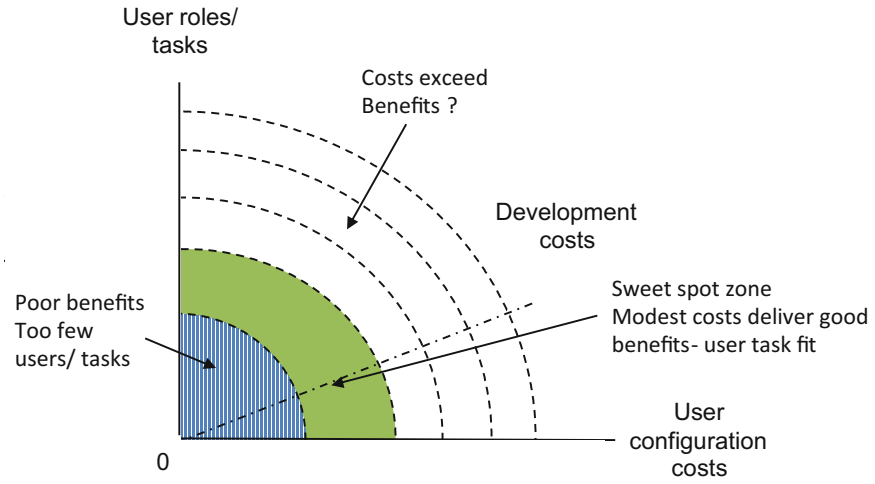

Fig. 1. A trade off framework for considering user and developer costs with potential benefits.

learning and operating costs. I have explored these questions in previous papers (Sutcliffe 2002b; Fischer et al. 2004); in this paper I will only consider simple configuration, since this part of the trade off space relates to the forthcoming project experience.

3 ADVISES Experience

ADVISES is a decision-support system for academic researchers and National Health Service public health analysts who investigate epidemiology problems (Sutcliffe et al. 2011). The two distinct stakeholder communities had different goals. For academic researchers, understanding the generic causes of childhood obesity by statistical analysis of health records was a high-level goal. In contrast, the goal of public health analysts was local health management; for example, identifying where best to target interventions, such as promotion of healthy eating campaigns. Two academic research epidemiologists (both male, age 31, 52) and seven public health analysts (four male, three female, age range 27–41) were interviewed and participated in requirements workshops. VBRE augmented UCD techniques to investigate the users' workflows to explore how new decision-support tools might be used by academic epidemiologists as well as by public health professionals.

The key issues identified were the apparent contradiction between expected and actual collaboration among the stakeholders, which suggested requirements for better collaborative tools with trust-building measures, e.g. visualisation of workflows and research activities. Security and privacy of data emerged as an important value, in particular the addition of security features to customise data access to particular stakeholder roles. Collaboration, security and trust were shared values, but differences

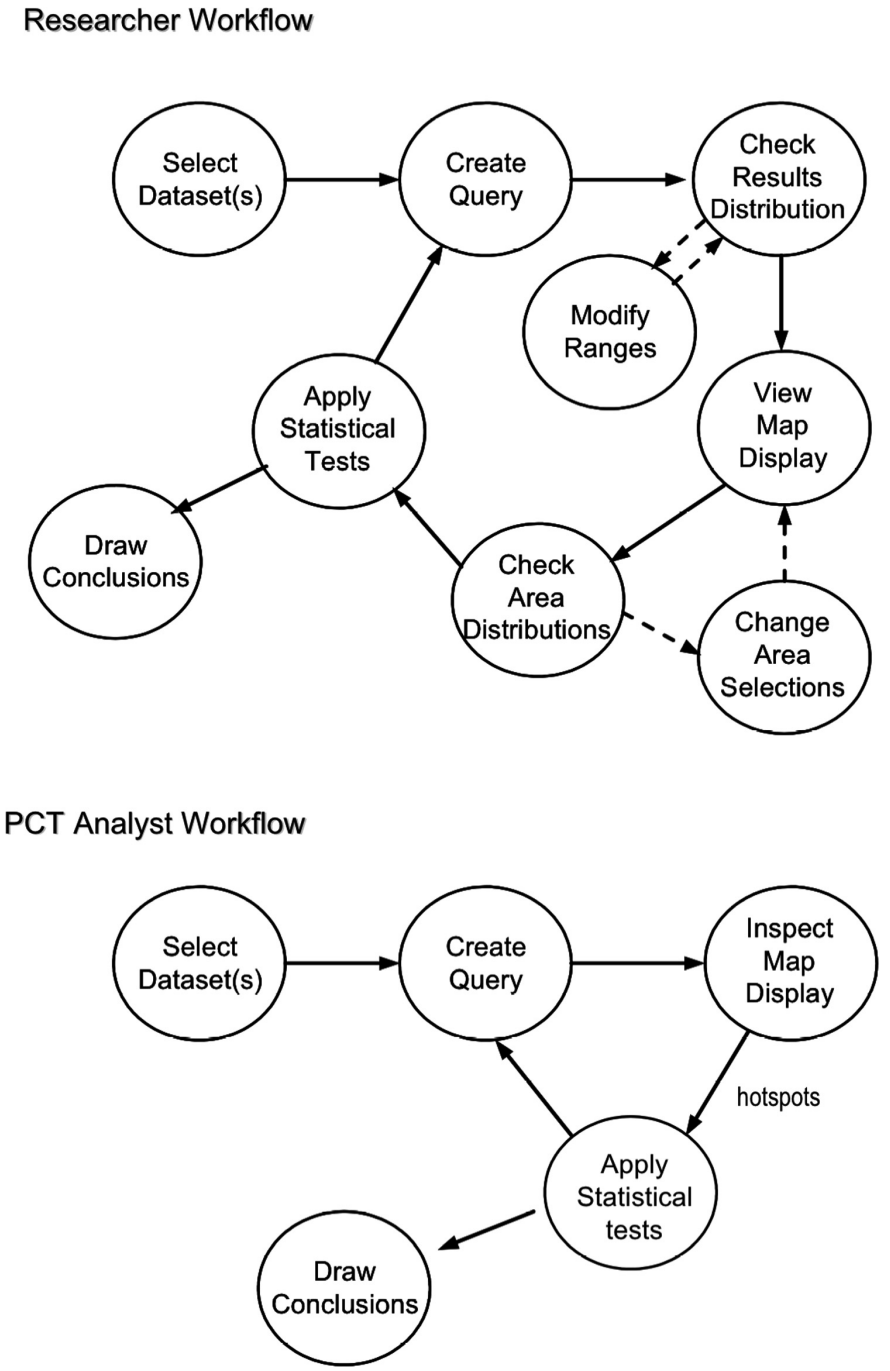

Fig. 2. Workflows for the research and public health analyst user stakeholders

between the stakeholder emerged during design exploration of prototypes, concerning customisation, adaptability and security. These were addressed by adding requirements for data security on servers, configurable workflows to match systematic or more opportunistic processes, while creative values were supported by interactive visualisation for data analysis. Collaboration and trust were fostered by an iterative user-centred RE process to build up trust, and by implementing the system as a collaborative application.

The workflows for each stakeholder group were quite different; see Fig. 2. The major functional requirements (goals) of the systems were for research and analysis support, namely database searches ranging from simple queries to complex associations between variables, leading to display of a detailed epidemiological data set in a context with map and graph overviews and functions to compare trends over time and different areas on maps. The researchers had a more sophisticated query investigation cycle and used more complicated statistical tests. In contrast, the public health analysts asked simpler questions directly related to spatial locations and used simpler statistical tests. Sociability, altruism and achievement motivations informed decomposition of stakeholder goals. For example achievement, altruism and systematic values led to a sub-goal to record analytic procedures, enabling academic researchers to track their own work, while also supporting public health analysts in sharing analysis techniques and results with colleagues. Another value clash between the stakeholders was the desire by the researchers to increase the statistical rigor of the analysts' investigation. Not surprisingly the analysts saw this as an imposition into their area of competence.

3.1 Implementation

The system was implemented in C# using MS Silverlight for graphics and animating map displays for trend questions, so that successive displays gradually morphed into each other to enable users to see change over time within different map areas. A distributed architecture (Fig. 3) was adopted and developed as a set of web services, with major class packages in the following functional areas:

- Dataset access: loads datasets from remote servers.
- Map display: displays maps using MS Charting libraries. Map displays can be overlaid with point data (e.g. location of health clinics, sports facilities).
- Charts and statistics display: runs basic statistical analysis scripts (R script calls) then displays range split histograms, box-and-whisker plots, etc., using MS Charting.
- Dialogue management: handles the query interface, interactive query-by-pointing and sliders.
- Expert advisors: classes that implement the statistics and visualisation experts, with data set monitors to trigger advice.

The prototype UI is illustrated in Fig. 4. The statistics advisor was a direct response to the value clash between the users' over-rigorous analysis procedures. The resolution was to provide a statistical expert advisor which encapsulates the researchers' knowledge; however, use of the advisor was discretionary so the analysts could ignore it if they so wished. The visualisation expert embedded knowledge about which charts

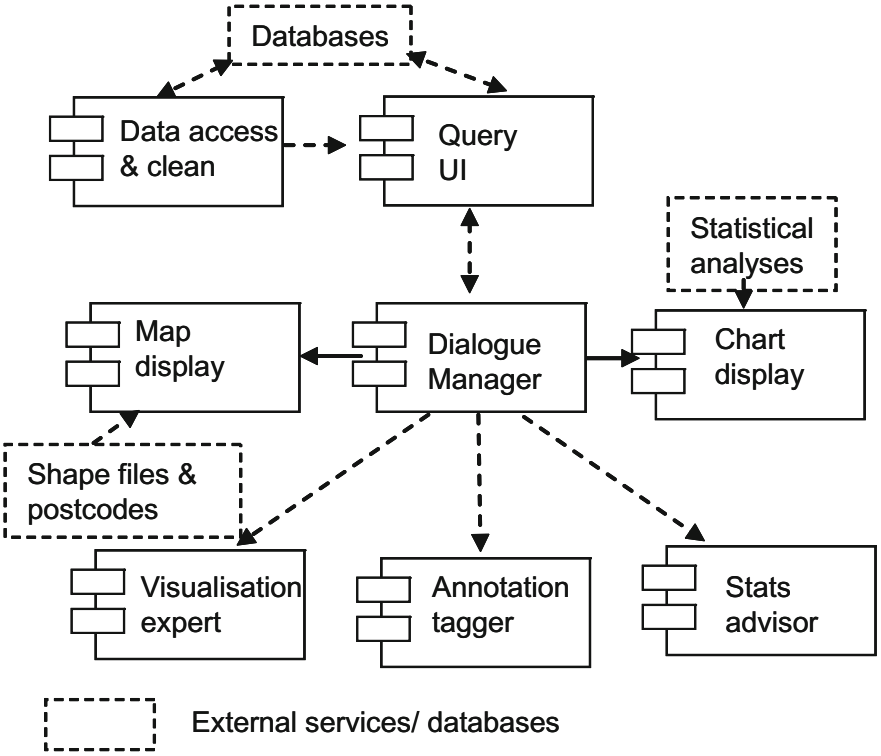

Fig. 3. System architecture of the implemented modules of ADVISES (UML package format)

to select for particular data types as well as choice of colours and shading to optimise the legibility of displays. This was a consequence of an implicit value clash between the users and system designers who wished to improve display design with cognitive knowledge (Ware 2000; Spence 2007). Fortunately both user groups were content with the visualisation expert which functioned non-obstrusively to configure the map-graph displays using a set of templates linked to frequent query types and their consequent data displays.

The original vision of ADVISES was designed to be a configurable system which could be adapted to other epidemiological applications, and in time to other e-health decision-support systems. This entailed developing adaptive data access and cleaning modules which could automatically adapt to new databases and data formats. However, it transpired that few external data sets have metadata description enabling such adaptation. Further configuration editors would have been necessary for tailoring output displays and the query interface. During the project it became clear that, technical difficulties notwithstanding, there was little appetite for developing more portable, configurable software since this served only the interests of the UK e-science programme, a remote stakeholder with less influence than the local, directly involved stakeholders (academic researchers and health analysts).

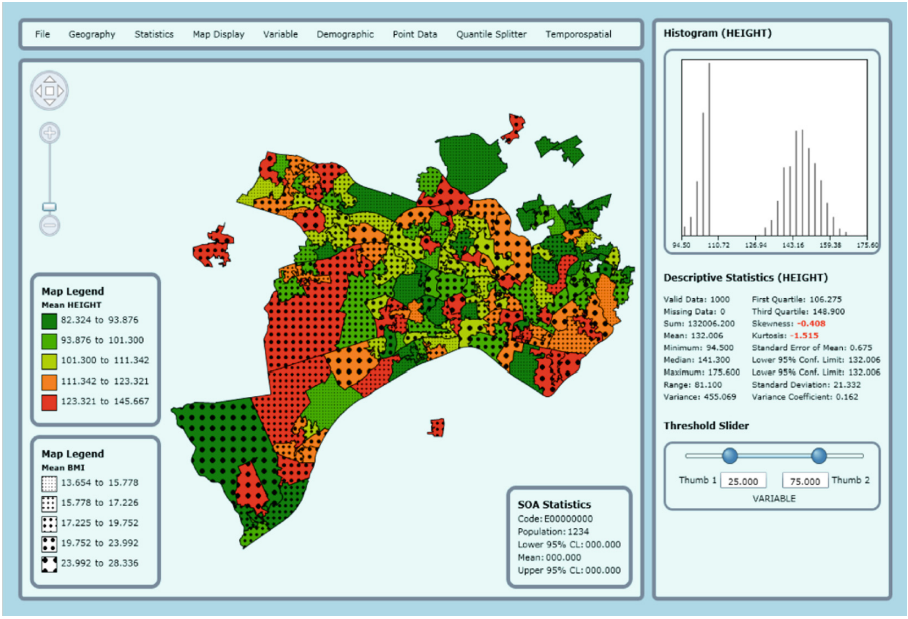

Fig. 4. ADVISES user interface showing the query results in map and graph displays

4 SAMS Experience

The SAMS (Software Architecture for Mental health Self management) project's main aim was to increase the proportion of dementia sufferers receiving an early diagnosis by detecting changes in their pattern of computer use (Stringer et al. 2015). At its core was a set of passive monitors that collect data as the user interacts routinely with the computer. This data is analysed to infer the stakeholders' cognitive health against a set of clinical indicators representing memory, motor control, use of language, etc. If the system detected potential problems, an alert message was sent to the user urging them to self-refer themselves to their GP for a check-up. There was a potential conflict between the clinical motivation to ensure that users responded to warning alert messages and users' need for privacy and self control.

The VBRE method was applied during interviews, scenario-storyboard requirements exploration sessions, and requirements analysis workshops. Requirements analysis was initiated with five workshops, conducted with a total of 24 participants (14 male, 10 female, age range 60–75). In the first session, the system aims, major components and operation were explained by presentation of PowerPoint storyboards illustrating design options (see Fig. 5), for the alert-feedback user interface, such as choice of media (video, text, computer avatars), content (level of detail, social network) and monitoring (periodic feedback, alert-only, explicit tests). Discussion focused on privacy issues in monitoring computer use, data sharing and security, ethical considerations, emotional impact of alert messages, stakeholders' motivations and their likelihood of taking follow-up tests. Requirements issues raised in the workshops were

explored further in 13 interviews presenting scenarios to illustrate similar design options with discussion on privacy, security and ethical issues. The scenarios used in both sessions were designed to test different design approaches that tacitly explored values, such as human-like presence in exploration, social networks (trust, sociability values) and explicitly probing issues of security and privacy.

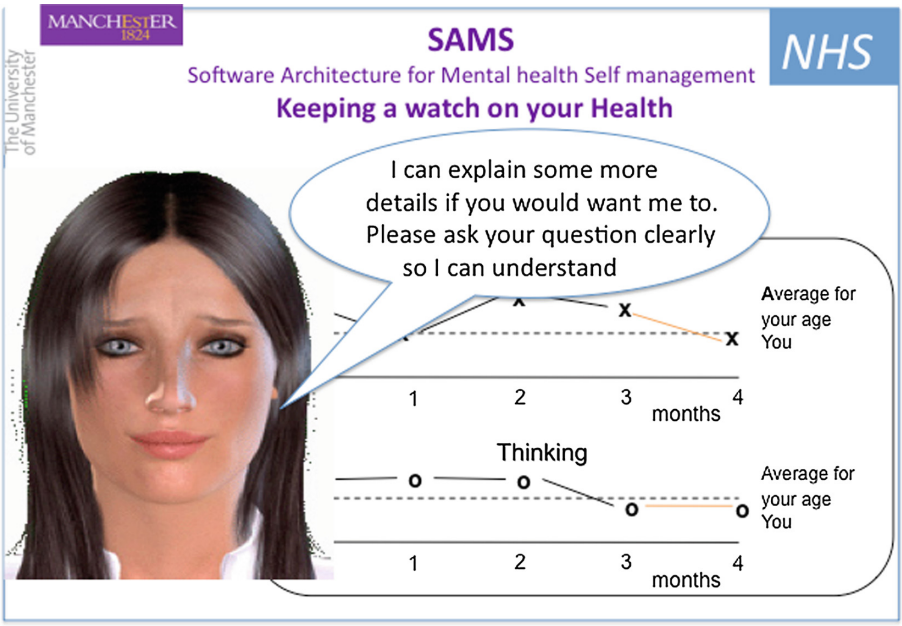

Fig. 5. Design options mock up illustrating avatar explaining feedback information display

Conflicts emerged in the values and requirements held by individual users as well as between end users and clinical-researcher stakeholders. End users expressed concerns over privacy and security arising from monitoring their computer use. Although they were reluctantly willing to share their data with the researchers for analysis, most participants insisted they should have control over their own data. Sharing data with their close kin/friends had to be under their control and the majority would not share information or the alert with their doctor. The majority were willing to allow monitoring of their computer use and e-mail text content, if it was anonymised to protect identity. Most participants expected to experience anxiety and fear if they received an alert message. Contact with a human expert or carer was cited as important support, with connections to support groups (e.g. the Alzheimer's Society) for reassurance (empathy) and as additional sources of information to motivate people to take follow-up tests.

Users had conflicting values (privacy, efficacy, altruism) which impacted on system reliability and accuracy. While these concerns were not UI properties they did influence non-functional requirements and design of the feedback UI. Users' motivations for self

control over their own health care, demanded a reliable and accurate system that detected early signs of dementia. Signs of change and usual behaviour patterns in the recorded data might indicate dementia, but they could have many other causes, such as mental health problems, e.g. depression, and not pathological causes such as mood changes. Teasing apart the signal of potential pathology from the noise of normal variation was part of the research problem. The user implications were to avoid false positive alarms. Furthermore, even true positive indications were unlikely to be 100% accurate, so potentially disturbing messages had to be delivered sensitively. This posed a further requirements dilemma. On one hand, the feedback messages needed to urge users to self refer themselves to their doctors for a check-up, but on the other, messages should not alarm people unnecessarily. The 'fear of diagnosis' problem implies complex persuasive UI design which is part of our continuing research.

Privacy and security were the most common values, with implications for controls over any data sharing, encryption, secure transmission and depersonalised data for research. These values clashed with users' motivations for monitoring so they could manage their own health (efficacy, empowerment), the desire for self control, and altruism by participating in the research which might help research on dementia. Self control was prioritised by implementing a user control to 'stop recording', and information visualisation so users could view summaries of their own activity.

Trust in the SAMS system was closely related to security, but it also involved accuracy of system information and diagnosis as well as organisational trust in the healthcare professionals. Trust-building was helped by a co-design consultation processes that involved users in the research and its results. The value clash between the need for privacy and continuous recording of users' activity resulted from the need to record as much data as possible to improve the fidelity of the analysis. This improved the effectiveness of SAMS as a research tool, and its subsequent version as a healthcare self-management system, aligned with users' self-efficacy and altruism (help research) values. The privacy goal also clashed with the researchers' motivation to record as much data as possible for research purposes. Data security was a shared concern for all stakeholders.

4.1 Implementation

To resolve the privacy clash, a UI function was provided so users could turn off data recording at their discretion. The system then prompted users to turn the recording back on after set time intervals of 5 and 10 min. If users did not comply after three reminders this was visible to the researchers from recording log files. They had the choice to phone the user to ask them to re set the recording. Data security was ensured by encryption of the recorded data and secure transmission to the university's server. Data depersonalisation also protected user privacy.

Preferences between users for different styles of feedback UI was addressed by providing a limited number of options which users could select when the system was set up, e.g. verbosity and tone of messages (empathetic/terse); delivery modality (text only, speech, speech plus avatar) and information provision (on/off). The latter choice allowed users access to visualisations and graphs of their recorded data on demand, with a limited set of display options of the quantity of data and summarisation. Choices

were limited by the cost of configuration and developing different UI displays. To date only a limited implementation of the feedback UI has been attempted, backed up by human intervention when the system detects potential problems. The persuasive UI design with its inherent conflict between the designer's goal of persuading people to take a course of action and possibly infringing personal freedom is still to be resolved.

5 Discussion and Conclusions

Conflicting UI properties and, more generally, conflicting user requirements, are inherent in many systems. This paper has reported some experiences in trying to make these conflicts explicit so they can be resolved by negotiation or design. Conflicts may appear as explicit differences in stated requirements; however, frequently different viewpoints between users are tacit and need to be analysed in terms of values and motivations. Methods such as VBRE (Thew and Sutcliffe 2017) and Value Sensitive Design (Friedman 2008) help in this endeavour.

If negotiation fails to resolve requirements conflicts, then a design response is necessary. Configuration at design time or adaptation at runtime are the usual choices. Configuration has the advantage of user participation, so they are aware of their choice and can pick design options that match their needs (Sutcliffe et al. 2006). However, configuration involves user effort and most users do not use customisation features provided by operating systems, and resent having to spend time choosing configuration options. Adaptation via an intelligent monitoring and automated change saves the user effort, but the changes are chosen by the designer and the change may produce inconsistency in the UI and induce usability problems (Fischer 2001). Apart from specialised areas such as recommender systems (Bonhard et al. 2006), manual adaptation or configuration has been preferred.

However, configuration imposes learning and operational costs on users. Furthermore, the configuration options are provided by designers, and this may limit the fit between the users' needs and the design options offered. In the ADVISES system we did not implement most configuration facilities because of constraints on developer resources. This decision was a trade-off between the perceived user demand for configuration, which was estimated to be low, and the considerable software development effect necessary. ADVISES implemented a resolution of clashes between user groups by giving users control over which facilities they chose to use, in particular the statistics advisor. This choice was a compromise since it failed to satisfy the researchers' wish to enforce statistical rigor in the public health analysts' work, although it did preserve the freedom of the analysts to control their own workflow.

In SAMS the value clashes between users' desire for privacy and their self-efficacy/empowerment motivation for healthcare self-management was partially resolved by provision of a UI control to temporarily halt data recording. The potential clash between the outcome of the monitoring where emotive messages had to be conveyed has not been resolved. This is an ongoing research issue concerning persuasive technology (Fogg 2009) where the designer or system owner's goal, i.e. to persuade the use to take a particular course of action, conflicts with ethical concerns that technology should not control people's behaviour by explicit or covert means.

In the perspective of the trade off framework in Fig. 1, in the ADVISES project the design choice was to increase complexity of the UI and impose more operational costs on users who had to search for the functions they needed. However, these costs were mitigated by providing limited configuration in the ability to turn the statistics advisor off. Avoiding product versions for the two user groups contained development costs. In SAMS the complexity of the monitoring UI was not an issue; however, the value analysis demonstrated than configuration at run time was vital by giving users the ability to turn off monitoring. Configuration of the persuasive users interface was not resolved during the project, although in future research we expect to provide configuration to match the style of persuasion to the user profile. This exposes the ethical dimension of user v. system control that needs to be added to the Fig. 1 framework.

Conflicts in the user interface may be overt in the form of different tasks, workflows or functional requirements owned by different user groups, as was the case with the researchers and public health analysts in ADVISES. In this case provision of tools to fulfil both sets of tasks is the answer. Harder to resolve are conflicts involving clashes between user values or non-functional requirements. These have to be refined into design choices which may partially satisfy one or more stakeholder groups; but as our experience has demonstrated, conflicts can often pose deep-seated irreconcilable dilemmas.

Acknowledgements. This work was partial funded by EPSRC Project ADVISES (Adaptive Visualisation Tools for e-Science Collaboration and SAMS (Software Architecture for Mental Health Self Management). The author wishes to thanks colleagues and participants in both projects for their contributions towards the production of this paper.

References

Bonhard, P., Harries, C., McCarthy, J., Sasse, M.A.: Accounting for taste: using profile similarity to improve recommender systems. In: Proceedings of Conference on Human Factors in Computing Systems, pp. 1057–1066 (2006)

Braz, C., Seffah, A., M'Raihi, D.: Designing a trade-off between usability and security: a metrics based-model. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4663, pp. 114–126. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74800-7_9

Cockton, G., Kirk, D., Sellen, A., Banks, R.: Evolving and augmenting worth mapping for family archives. In: Proceedings of the 23rd British HCI Group Annual Conference on People and Computers. British Computer Society, London (2009)

Fischer, G.: User modeling in human-computer interaction. Interact. Comput. **11**(1–2), 65–86 (2001a)

Fischer, G.: User modeling in human-computer interaction. User Model. User-Adapt. Interact. **11** (1–2), 65–86 (2001b)

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N.: A framework for end-user development: socio-technical perspectives and meta-design. Commun. ACM **47**(9), 33–39 (2004)

Fogg, B.J.: The behavior grid: 35 ways behavior can change. In: Proceedings of the 4th International Conference on Persuasive Technology, p. 42. ACM (2009)

Friedman, B.: Value sensitive design. In: Schular, D. (ed.) Liberating Voices: A Pattern Language for Communication Revolution. MIT Press, Cambridge (2008)
van Lamsweerde, A.: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley, Chichester (2009)
Mylopoulos, J., Chung, L., Yu, E.S.K.: From object-oriented to goal-oriented requirements analysis. Commun. ACM **42**(1), 31–37 (1999)
Robertson, S., Robertson, J.: Mastering the Requirements Process. Addison Wesley, Harlow (1999)
Sommerville, I., Kotonya, G.: Requirements Engineering: Processes and Techniques. Wiley, Chichester (1998)
Spence, R.: Information Visualisation, 2nd edn. Pearson Education, Harlow (2007)
Stringer, G., Sawyer, P., Sutcliffe, A.G., Leroi, I.: From click to cognition: detecting cognitive decline through daily computer use. In: Bruno, D. (ed.) The Preservation of Memory: Theory and Practice for Clinical and Non-clinical Populations, pp. 93–103. Psychology Press, Hove (2015)
Sutcliffe, A.G., Fickas, S., Sohlberg, M.M.: PC-RE: a method for personal and contextual requirements engineering with some experience. Requirements Eng. **11**, 157–163 (2006)
Sutcliffe, A.G., Thew, S., Jarvis, P.: Experience with user-centred requirements engineering. Requirements Eng. **16**(4), 267–280 (2011)
Sutcliffe, A.G.: The Domain Theory: Patterns for Knowledge and Software Reuse. Lawrence Erlbaum Associates, Mahwah (2002a)
Sutcliffe, A.G.: User-Centred Requirements Engineering. Springer, London (2002b)
Thew, S., Sutcliffe, A.G.: Value based requirements engineering: method and experience. Requirements Eng. (2017). https://doi.org/10.1007/s00766-017-0273-y
Ware, C.: Information Visualization: Perception for Design. Morgan Kaufmann, San Francisco, CA (2000)