

Alistair Sutcliffe · Stephen Fickas
McKay Moore Sohlberg

PC-RE: a method for personal and contextual requirements engineering with some experience

Received: 10 October 2005 / Accepted: 30 January 2006 / Published online: 14 March 2006
© Springer-Verlag London Limited 2006

Abstract A method for requirements analysis is proposed that accounts for individual and personal goals, and the effect of time and context on personal requirements. First a framework to analyse the issues inherent in requirements that change over time and location is proposed. The implications of the framework on system architecture are considered as three implementation pathways: functional specifications, development of customisable features and automatic adaptation by the system. These pathways imply the need to analyse system architecture requirements. A scenario-based analysis method is described for specifying requirements goals and their potential change. The method addresses goal setting for measurement and monitoring, and conflict resolution when requirements at different layers (group, individual) and from different sources (personal, advice from an external authority) conflict. The method links requirements analysis to design by modelling alternative solution pathways. Different implementation pathways have cost–benefit implications for stakeholders, so cost–benefit analysis techniques are proposed to assess trade-offs between goals and implementation strategies. The use of the framework is illustrated with two case studies in assistive technology domains: e-mail and a personalised navigation system. The first case study illustrates personal requirements to help cognitively disabled users communicate via e-mail, while the second addresses

personal and mobile requirements to help disabled users make journeys on their own, assisted by a mobile PDA guide. In both case studies the experience from requirements analysis to implementation, requirements monitoring, and requirements evolution is reported.

1 Introduction

Functional requirements have traditionally been considered as specifications that satisfy the goals of the majority of users. Although the concept of establishing sub-sets of requirements matched to different stakeholder groups has been advocated in the viewpoint tradition of requirements engineering, e.g. [1] (VORD) and [2], the concept of specifying requirements for individual users has not been explored. However, in human computer interaction, requirements are seen as an individual concern for customising the user interface and matching the mix of functional requirements to individuals [3].

Variability and specialisation of generic requirements to fit more specialised usage domains have been investigated in the product-line literature [4] as variation points that specify where generic requirements may be tailored. The individual dimension of requirements has been partially addressed in the *i** modelling language where the capability and abilities of agents can be specialised to model the skills and preferences of individuals [5], so the requirements for individual users could be matched to ability profiles. Recognition that requirements may change over time because the system environment or context changes has been explored in requirements monitoring [6], where monitors track a requirement over time to evaluate the match or mismatch between system operation and particular goals.

In spite of these initiatives, no comprehensive method for analysing individual or contextual requirements has been proposed. We argue that a framework for individual requirements is necessary as technology products

A. Sutcliffe (✉)
Centre for HCI Design, School of Informatics,
University of Manchester, P.O.Box 88,
M60 1QD, Manchester, UK
E-mail: a.g.sutcliffe@man.ac.uk

S. Fickas
Department of Computer & Information Science,
University of Oregon, Eugene, OR 97403-1202, USA
E-mail: fickas@cs.uoregon.edu

McKay M. Sohlberg
Communication Disorders & Sciences,
College of Education, 5251 University Of Oregon,
Eugene, OR 97403-1202, USA
E-mail: mckay@uoregon.uoregon.edu

become personalised. Furthermore, with the growth in ubiquitous computing, requirements may not only vary by individuals but may also change over space and time in location-aware systems [7], [8]. In this paper we propose a framework for Personal and Contextual Requirements, a method for their capture incorporating a trade-off analysis to decide how personal requirements should be implemented. The following section describes related work; Sect. 3 discusses the framework and implications for architectural requirements; Sect. 4 develops the PC-RE (Personal and Contextual Requirements Engineering) method which is applied to two assistive technology case studies in Sect. 5. This is followed by a description of the cost–benefit trade-off analysis in Sect. 6. The paper concludes with a brief discussion.

2 Related work

Several requirements taxonomies have been proposed that classify requirements into different categories of non-functional requirements (NFRs) [9], [10], functional requirements and services [11], [12]; indeed, taxonomy is an accepted means of managing requirements in RE tools. However, previous taxonomies have classified requirements according to their subject matter rather than to the agents to which they pertain. Monitoring the implemented system against requirements so it can adapt to evolving requirements over time was proposed by [6], while the impact of location on requirements was partially explored in the Inquiry Cycle [13], where the location could influence the acceptability of system output. However, there has been no systematic treatment of contextual influences on requirements.

Requirements for stakeholder groups are familiar in RE methods (e.g. ScenIC [14]; Volere [2]); however, change over time is not explicitly modelled apart from concerns over requirements creep and evolution. Change in location is rarely specified, even though globalisation and cultural effects on products are known to be important [15], [16]. Understanding cultural impact on requirements is still in its infancy, although ethnographic studies suggest that very different requirements arise in situ; for instance, privacy requirements for automatic teller machines are very different between eastern and western societies [17].

The mutability of requirements to suit individual users has been accepted in human computer interaction, where a distinction is drawn between generic task support requirements (i.e. functional requirements) and system features which can be customised for individual user needs [16], [18]. Two architectural approaches to handling personal requirements are adaptive systems, in which the system monitors the user's behaviour and then changes services or the interface look and feel to match the user's needs; and adaptable systems, where customisation is user-controlled at design time [19]. End-user programming applications often employ mixed initiative dialogues and fusion of these two strategies

[20]. Temporal and spatial change in requirements has implications for requirements capture and system architecture which have yet to be fully explored in the RE community.

3 A framework for personal, contextual RE

The motivation for the framework is to describe not only functions that meet people's goals but also characteristics of the users, and how they would like computer systems to achieve their personal goals. The framework accommodates matching requirements to individual needs, how individual needs change over time, how requirements evolve as people learn and their ambitions grow, and finally the needs for universal accessibility and the ageing user population [3].

We propose a three-layer framework for personal and contextual requirements, with two change dimensions of location and time which act on each layer, as summarised in Fig. 1. The layers progress from general requirements at the user group layer, to individual user characteristics, and to personal goals. Within each layer requirements vary over space and time to encourage not only analysis for evolving requirements but also contextual influences. Contextual influences may vary from the cultural and social system environment to effects of specific locations.

Stakeholder group In this layer, the change context affects requirements in cultural adaptation of products, e.g. language localisation, change in business processes between European and American models; while time influences how requirements change as business processes evolve and product functionality becomes more complex [21]. Contextual enquiry by surveys and prototype evaluations in different cultures are necessary to specify product versions for different cultural markets. The architectural considerations are design of products with variation points so they can be configured for different cultures, although the extent and nature of cultural adaptations are still poorly understood. Other architectural implications are design for customisation to accommodate context effects (language localisation); design of monitors and adaptive functions for mobility; customisable or adaptive user interfaces to deal with user skills change; and a flexible adaptable architecture to evolve as business processes change.

User characteristics This layer models the needs of the individuals who share a common profile of skills and abilities that influence how they interact with technology. User characteristics are an individual user ability profile, which can be used to customise the means of human computer communication for the elderly, disabled, people with different learning styles, etc. Location affects user characteristics as people's abilities change with place, such as the need for adapting communication modalities in noisy

environments, while people’s skills and abilities change over time as they adapt, for example, to computer-based training. Architectural implications are for choice and adaptive communication modalities. In the general user population, baseline knowledge will influence content in website and CAL applications, while in assistive technology this layer is key. The user characteristics layer describes the users’ physical and mental abilities, so this affects requirements for interaction directly as well as functional requirements indirectly. User characteristics requirements imply the need to develop individually tailored versions of applications that are either configured for the user at design time or automatically adapt to the user’s needs. Change over time occurs as people learn system functions and need new styles of dialogue as they become more skilled; while slower change, e.g. age-related decline in cognitive and motor abilities, necessitates change in magnified visual displays and slower response times. The range of individual user abilities needs to be analysed against inventories of modality abilities, knowledge and capabilities, and general cognitive abilities. User characteristics are assessed by psychology-based questionnaires and tests to measure cognitive, physical and perceptual abilities (e.g. [22]) or by interviewing users to gather information on general abilities, experience and skills [23]. Assessing the user’s characteristics also produces an inventory of specific skills that we assume the user possesses to successfully operate the system given the user interface requirements for communication and interaction. These user characteristics are a complementary specification of what we can reasonably expect of the user (individually and collectively) and what needs to be implemented to enable effective use of the system.

Personal goals We argue it is important to analyse requirements from an individual viewpoint, especially in domains where customisation is important. In this layer, change over time depends on the stability of people’s wishes, while the contextual interaction may be influenced by how their goals affect location and social setting (e.g. social settings may influence privacy and hence display of personal information). Personal goals are held by individuals and become important in applications where customisation of individual service is a prime objective, e.g. learning and training applications, entertainment and games, personal knowledge management and assistive technology. Models of individual users have always been important for educational applications where the abilities of each individual are matched to content and pedagogical delivery.

The type of requirements in each layer and their interaction with the change and context dimension are summarised in Table 1. It should be noted that the layers do not imply a hierarchy of requirements; rather, the layers represent concerns that pertain to stakeholder groups or individuals.

Personal goals can be realised either by design or customisation, e.g. by configuring the toolbar in drawing applications, or by automatically adapting solutions to the individual’s needs. Personal requirements may be contextual and location sensitive; for example, NFRs such as privacy, security and information accuracy can interact with functional requirements such as information display, according to location. Some personal goals may be implemented as preference settings under user control, e.g. the presentation of information (graphs and/or tables), and aesthetic details such as screen savers and ring tones on mobile phones. Personal goals are assigned attainment levels on a 1–5 scale so we can assess how well the system and user have approached the

Fig. 1 Personal requirements framework and change dimensions

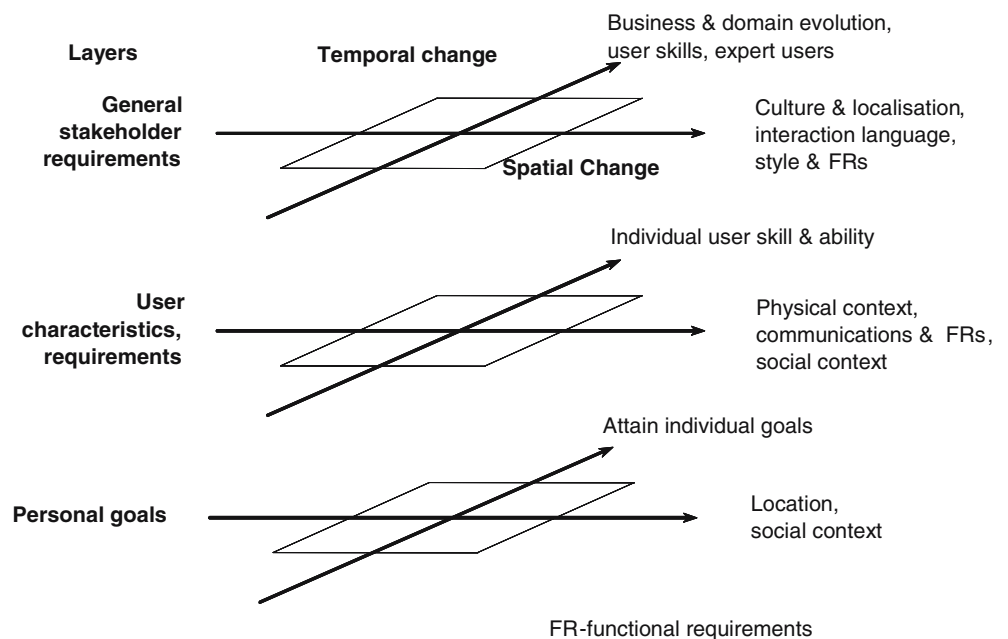


Table 1 Requirements framework and effect of time and location

Requirements layer	Examples	Change over time	Context/location change
General stakeholder requirements	Stakeholder-tailored versions of products, product lines	Expert users need power functionality, product lines	Globalisation: requirements tailored for countries, cultures, language
User characteristics requirements (interaction and communication)	Modalities and capabilities: accessibility, ageing, but also learning, cognitive and social abilities; individual ways of working, task requirements	Individuals learn and become skilled, adapt to new ways of working, individual styles of interaction	Needs change with context in mobile applications
Personal goals may be functional or non-functional requirements	Personal needs for services, task support, attainment goals for self, linked to motivations	Personal goals have different attainment time-scales	Personal goals can be sensitive to physical and social setting, e.g. privacy and social setting in group/alone

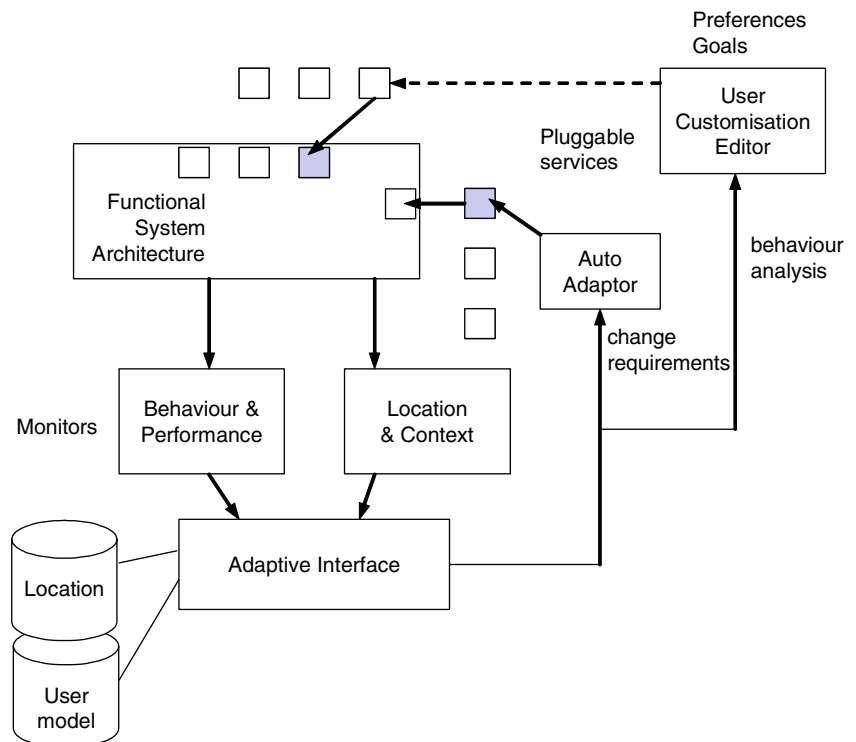
ideal achievement of each goal. The attainment levels also specify the assumptions associated with each goal, such as the necessary customisation of the software, modification to requirements (i.e. re-design) and user training.

Consideration of personal and contextual requirements produces a new set of meta-requirements: the need to develop user models, monitors and mechanisms for adaptive systems or customisation editing/design facilities. The implications of personal requirements on architecture are summarised in Fig. 2.

Three implementation paths are suggested for user characteristics requirements and personal goals. First, development as conventional system functions; secondly, implementation as user-customisable features; and thirdly as automatic system adaptations. Figure 2 illustrates the architectural considerations of the second and third paths, which also necessitate specification of

architectural requirements according to the chosen development pathway. Monitors capture information on the users' behaviours, errors, location and environmental context. Intelligent interpreters have to infer higher-level states from monitored data either using knowledge of the domain and user to make inferences, or via learning mechanisms. The use of such inferences then depends on the choice of automated adaptive mechanisms or providing information to help the users make customisation decisions supported by editing and configuration facilities. The system architecture has to be configurable following product-line specification to enable change via either route. Acquisition dialogues may also be necessary to capture personal goals and help user-driven customisation from a toolkit of services; alternatively, the system selects services to fit the user's goals. The implementation pathways have different cost-benefit trade-offs for designers and user stakeholders; for

Fig. 2 Relationship between the requirement layers framework and system architecture



instance, customisation imposes a learning burden on the user who has to learn how to customise the application. Cost–benefit analysis techniques are described in Sect. 6.

4 Scenario-based personal contextual requirements analysis

In this section we develop the PC-RE framework into a scenario-based requirements analysis method, based on our previous work with CORE and SCRAM [24], [25]. CORE used ethnographic techniques to investigate users' requirements in context, and then employed prototyping or Wizard of Oz techniques to evaluate initial designs with users and hence refine the requirements. CORE was motivated by assistive technology applications, so it emphasises a clinical profile of individual users' abilities and attainment levels which are taken forward in the PC-RE method. PC-RE also follows the scenario-driven approach of SCRAM with mock-ups and early prototypes analysed in combination with design rationale to drive the requirements dialogue with users. PC-RE adds more guidance on the use of scenarios and developing requirements for individual user profiles. The method contains four sequences of investigation that can be deployed sequentially, although in practice requirements investigation tends to interleave issues from the four threads. Two threads—*analyse cultural context* and *analyse spatial context*—are motivated by the spatial dimension in the framework. The two threads *analyse temporal implications* and *analyse evolving requirements* correspond to the time dimension in the framework. The framework layers can apply to any of these threads, although the user characteristics and personal goals layers are covered in a fifth thread. Group stakeholder requirements analysis is not made explicit as these are an assumed default. Within each thread the specification/refine stages are motivated by the generic architecture for adaptive applications, as shown in Fig. 2, and hence point to monitoring and adaptation issues. PC-RE is not intended to be a complete RE method so many issues such as modelling requirements specifications are not covered. Instead it is intended to supplement other methods (e.g. Volere [2]) with individual and contextual perspectives. Contextual requirements analysis deals with the issues of mobile applications, while the personal RE addresses the requirements for applications that focus on individual users, i.e. education, training, assistive technology, clinical medical systems and entertainment.

4.1 Method overview

The road map for the PC-RE method is illustrated in Fig. 3. The first step entails considering the type of application to determine requirements relating to individual users and/or changing requirements of a temporal or spatial nature. The results of this assessment

will invoke one or more of the pathways that advise on capture and analysis for personal and contextual requirements. Many applications will concern more than one pathway, hence the method is applied iteratively. Most context-aware applications will invoke several pathways to address requirements for the five Ws [8]: *Who*, the identity of an individual user; *What* the users are doing, interpreting their activity; *Where* they are located; *When* in time are certain services requirements, and when in a history of activity; and finally *Why*, capturing the users' intent by deeper analysis of *What*. The first pathway deals with international applications, where the advice is given on analysing the effects of culture and then defining requirements for localising the systems and specifying how it will be configured or tailored for different versions. The second pathway addresses analysis of requirements for mobile and context-aware applications. The third and fourth pathways distinguish between predictable and less predictable change of requirements over time. Predictable change is implied in applications where time drives adaptation of functionality, for example, patient monitoring system in healthcare, or environmental control system in home use or industrial process control. The essence of predictable temporal change is that functional requirements are known, and the dependency between time and those requirements can be defined. Actual time periods may vary from seconds to days or months.

Temporally unpredictable changes occur in systems where evolution of requirements can be expected, but the nature of change cannot be anticipated in detail; when change might occur is also uncertain. In this case requirements analysis is similar to product-line architectures, where variation points are created for loci of potential change and requirements are made as adaptable as possible by generalisations. Finally, for applications in assistive technology, healthcare training and education, personal requirements are likely to be important. Advice on this pathway is given on defining personal goals and user profiles. This is followed by setting achievement levels and monitoring progress towards those goals. Several pathways share common process and advice on specifying the requirements for monitoring devices and requirements for adaptation to change over time or location. Finally, the pathways converge on common tasks that appear in all requirements engineering processes such as negotiation and requirements validation; these tasks are not described further in this paper.

4.2 Contextual requirements analysis

Following the framework, contextual analysis involves considering the requirements implications of change in location at the global as well as the local context. Creating contextual scenarios for applications that are intended for an international market requires sensitivity to

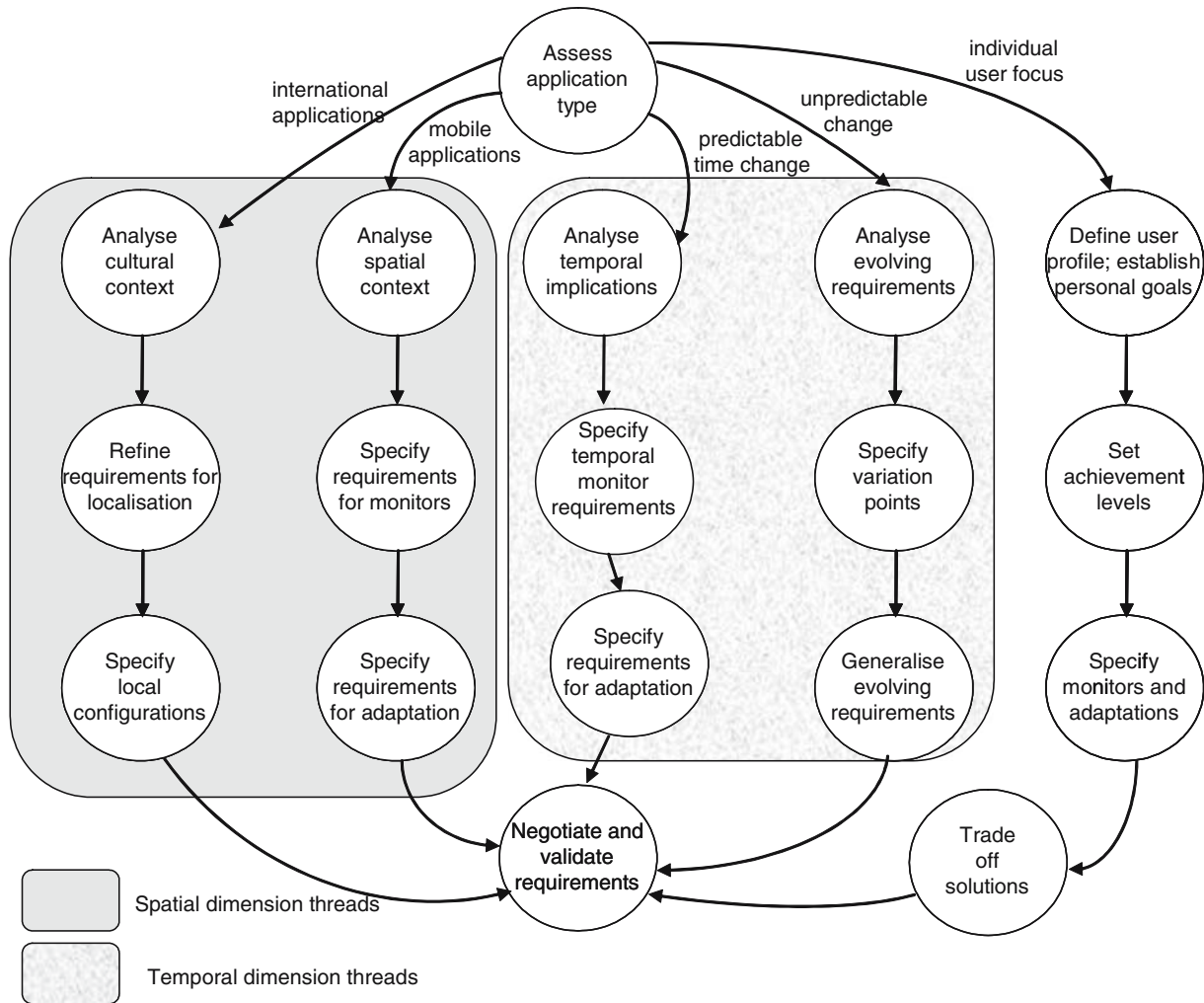


Fig. 3 Road map of the PC-RE method

local cultural conditions. Ultimately, there is no substitute for sourcing scenarios from users who belong to the cultures, nationalities and linguistic groups in the intended market place. However, if culture-located users are not available, then the following guidelines can be used to challenge scenarios sourced in one culture (i.e. usually Europe or the USA), to consider requirements implications in other cultures. The inspiration for the guidelines comes from reports of requirements investigations and ethnographic studies in the RE and HCI literature [15], [17], [26], [27].

1. Consider the *cultural impact* on the social context on the user. Cultural impact has been assessed in most studies using a combination of Hofstede's [28] and Hall's [29] dimensions, of which the important ones are:

- *Uncertainty avoidance*. Cultures with high uncertainty avoidance do not like ambiguity, and hence prefer precise instructions and fewer options.
- *Power-distances*. High power-distance cultures have a more marked hierarchical division of roles in

society which is reflected in different attitudes to authority, initiative and responsibility. The requirement implications are to align the requirements closely to stakeholder roles and their perceived power.

- *Individualism/collectivism*. More individualistic cultures place more emphasis on personal goals, which has consequences for that aspect of the PC-RE framework; in contrast, collectivist cultures are more receptive to shared goals, and collective decision making.
- *Context*. High-context cultures tend to prefer more visual and symbolic representations of content whereas low-context cultures prefer hard facts, detail and statistical evidence. This aspect has obvious implications for selecting and presenting information with different media.
- *Time*. Cultures sub-divide into single-threaded, those who prefer doing one task at a time, and multitasking. This dimension has implications for requirements in decision support, agenda management and work organisation.

Some NFRs are culturally sensitive; for example, in a study on ATM usage in India users needed more privacy than was afforded by the standard western ATM design; not only does the western model have financial transactions conducted in public places, but also high power-distance which excludes many users who did not consider ATMs were intended for use by lower status individuals [17]. Cultural impacts can be analysed using scenarios with the above dimensions to assess the implications for privacy, security, usability and other NFRs that may be related to the location of use.

2. Beware of the impact of *authority relationships* on users' goals. Consider the effect of different levels of authority and responsibility as obstacles for the acceptance of goals on users, particularly where the goals are owned by manager stakeholders.

This is illustrated by a study of outsourcing development in India, with requirements implications for use of creativity tools and CSCW coordination tools to link UK and Indian software developers. The UK software house made false assumptions about personal relationships in the Indian software house and expected the Indian team to develop informal specifications creatively following their UK practice. The Indian perspective was that development was governed by more authoritarian relationships of delivering to detailed specifications.

3. Assess the impact of culture on the users' *work patterns*. Consider the effect of culture on how work is organised by the system. Cultures with a higher power-distance tend to expect and accept more rigid procedures, whereas low power-distance cultures require more flexibility and ability to adapt working practices to local conditions.

This is illustrated by the problem encountered when implementing Enterprise Resource Plans which can impose an inflexible pattern of work dictated by western business practices. In other low power-distance cultures such top-down imposition of working practices is resisted [27].

4. Assess the *literacy* of the local user population. Challenge any assumptions in scenarios made about the knowledge and skills of users. Users may not have the knowledge or literacy skills to operate complicated functions with western style dialogues.

The Indian ATM study illustrates this problem, where many users had low levels of literacy even when the user dialogue had been translated into their language. Another illustration highlights requirements for public health systems in which the assumption that simplified English dialogues and translation into local language would be sufficient for communication with a Somali immigrant community. The eventual solution was to use an iconic pictorial language to communicate health issues in a medical consultation support system [30].

5. Evaluate requirements for *local languages*. Challenge any requirements that depend on the user understanding English and beware of simple machine translation solutions to localisation.

Studies of language localisation have demonstrated that machine translation approaches are prone to many errors caused by different interpretation of words, local dialects and idioms [30]. Requirements for language localisation have to be tested by implementation of language tailored user interfaces, which can then be tested by native speakers either directly or by posting the application on a website and inviting comments from local users, ideally in their own language.

4.3 Requirements for mobile and context-aware applications

The requirement problems in this pathway are first to define the number and type of spatial contexts that will change the requirements. In some cases requirements are closely coupled to the location; for instance, in a mobile tourist or museum guide the information content and user system dialogue have to be geared to the user's location. Other requirements demand more complex interpretation of location so the system functions can be adapted to the users' social or work context. An example is the use of notifier systems and dual task displays which will change according to whether the user is in a location that demands no interruption (e.g. public lecture) or in a more private space. A useful conceptual framework for considering the implications of space on requirements is the "locales" model [31] which divides space into different categories according to social needs for NFRs of privacy, security, accuracy, etc. However, determining the nature of a social space is a non-trivial problem of sense-making from data about the user's environment.

Monitoring requirements for mobile applications need to be specified for:

- The devices and sensors that capture environmental input, e.g. video, GPS, audio, light, movement detectors. Sensors may be active transmitters such as RFID tags in instrumented environments or passive devices that can work in non-instrumented environments, e.g. video, audio capture.
- Functions and processes to interpret low-level data streams into meaningful data, e.g. from a sensor voltage output to a temperature reading.
- Interpreters that make sense of the data using a model of the domain, e.g. body temperature above a certain threshold is dangerous for a patient and hence an alarm should be signalled.
- Models of the domain, either static or preferably dynamic, and updatable to reflect changes in the environment.

Interpreters invariably rely on data fusion from several sensory inputs, so the requirements for handling several datastreams need to be investigated. A useful framework for considering the requirements for interpreters [32] is to decide the level of change which needs to be inferred, ranging from the semaphore level (an event has happened); identity (the event or agent can be identified); attributes (properties of the event/object can be inferred); to causality of the event (intent of the agent can be explained). In a spatial context location might be Cartesian coordinates in a space, approximate location in a topology (near to) or exact location within a topographic model. The results of interpreters are frequently ambiguous so further requirements need to be considered for the mediation dialogues to help the user make effective use of context-aware applications [33].

- Feedback functions are necessary to make the user aware of the system's interpretations (or guesses) about the state of its environment. Confidence-level indications help people decide whether to accept the system's decision.
- Default interpretations need to be defined when input data is not available, or is inadequate.
- Mediation dialogues should be planned so users can override the systems' decision.

Once the output from the monitoring process has been defined, a further set of requirements are necessary to specify the system's adaptation to change in location. The nature of adaptation will depend on the domain; in many cases it will be a change to users' services to deal with change in NFRs such as privacy in different locales. In embedded systems, change may need to adapt to the environmental properties of the location, for instance a computerised automated braking system in a vehicle needs to adapt according to the location being a wet or a dry road.

4.4 Predictable temporally variable requirements

This pathway covers applications where the system has to adapt its response over time, which might be at a specific point in time or periods in scales ranging from nanoseconds to years. Typical examples of systems in this class are environmental and process control systems in home use or industry (thermostats, time-based controllers of devices, time-based controllers of manufacturing process, chemical process control, etc.). Requirements for temporal monitoring are reasonably simple since a system clock will usually suffice, although interpretation of significant temporal events may not be non-trivial. Temporal logics as found in some requirements languages (e.g. KAOS [34]) may be necessary to specify complex temporal dependencies, e.g. synchronisation, overlapping time intervals, less predictable temporal dependencies (until, eventually, etc.).

Once temporally significant events have been specified, in common with other pathways, the next task is to

specify how the system should adapt either at a point in time or for a duration. Some examples of predictable temporal change in requirements for user-centred systems are changing services during the day. In a notifier system, stock market updates may be sent between 9 a.m. and 5 p.m. but after 5 p.m. schedules of TV shows or local entertainment would be more appropriate. This class of requirements and the following less predictable requirements are both deferred [35], in the sense that they are analysed when the system is designed, but implementation is delayed until the context is appropriate, either at a point in time or when the system environment has evolved.

4.5 Unpredictable but evolving requirements

Many applications potentially fall into this class, as requirements need to evolve as business changes. Product-line engineering is a response that attempts to anticipate evolving requirements by carrying out an exhaustive domain analysis, and then specify systems with stable core functionality and variation points where requirements change can be anticipated. Requirements can also be generalised to make them more adaptable for future implementations [36]. Some guidelines for making requirements more generic are to:

- Parameterise processes and procedures so the behaviour of a system can be changed by adjusting parameters on an initialisation file. In a computer-aided learning application, parameterisation could be used to change the pedagogical strategy (explanation, demonstration, learning by example) used by the system according to the sophistication of the learner.
- Enable run-time binding of procedures and data structures, so the system can adapt by changing component libraries.
- Provide facilities for adding new functions at variation points.
- Make system architectures loosely coupled, so new modules can be added without upsetting too many dependencies.

Requirements for evolving systems become a process of design for reuse (see [18]) in which as many functional requirements as can be anticipated in the future are specified; the system architecture is then designed to be sufficiently flexible for new components to be added and redundant ones deleted.

4.6 Personal user requirements

In this pathway the individual user rather than change is the focus. However, many connotations of requirements may be shared with other pathways. A central dilemma in user interface design is to specify user interfaces that suit the requirements of individual users, while delivering

a general product that can be used by many (individually different) users. Requirements for adaptation and monitoring processes are therefore necessary, for instance as users' experience with a system grows they require less help and supportive dialogues, and access to more sophisticated power functions. System functionality should therefore be revealed gradually so users can learn without becoming overwhelmed by complexity [37].

The processes in this pathway are first to define an individual user profile. These requirements are frequently set by another, expert stakeholder, e.g. the designer of a computer system defines user stereotypes, a clinician specifies an ability profile of an individual for a healthcare system, a teacher defines a profile for a student's learning style and abilities. Personal goals are elicited directly from individuals and owned by them. For both personal goals and user profiles, attainment targets can be sets which become benchmarks for monitoring processes. Trade-off analysis is particularly important in this pathway because user profile goals set by an expert stakeholder may conflict with personal ambitions; furthermore, because goals are personal there are different costs and benefits that may be accrued in achieving them. In common with other pathways, requirements for adaptation are defined to complete the process.

5 Case studies

The case studies form part of a long-term investigation into developing assistive technology for people who have suffered traumatic brain injury (TBI) in Eugene, Oregon, USA [24], [38]. The users live either in sheltered accommodation or on their own with supporting care providers, but have limited social interaction. The objective of the Think And Link (TAL) project is to increase the user's independence and mobility by use of technology. The challenge is to match the technology to individual users' requirements when each person has unique requirements resulting from the nature of their

injury. The CORE method was developed on preliminary initial study of the e-mail application [24]; this paper reports application of PC-RE to more recent investigations.

5.1 E-mail for cognitively disabled users

We applied the PC-RE method to two projects: an e-mail system and a navigation support system for cognitively disabled users. An action research diary recorded our experiences in applying the method during the case studies, supplemented with debriefing sessions among the developers. The top-level goals in both projects were to empower these users socially and personally so they can communicate with one another and use public transport to meet others, etc.

The requirements for the TAL e-mail case study are illustrated in Table 2. The first layer gives the group-level view of an e-mail system for cognitively disabled users, which is a simplified version of standard e-mail functions. The subsequent layers focus on two individuals, as space precludes describing the breadth of requirements for the six individuals in the study in this paper, although the diversity of requirements encountered will be discussed.

Group requirements Requirements were gathered by interviewing and focus group techniques with reference to the functions provided by existing e-mail products. Requirements have to take account of the diversity in cognitive disability, which include language problems (aphasia, dyslexia), working memory and wandering attention, planning and executive function disorders, poor learning and problem-solving abilities. These lead to requirements which supplement the e-mail application requirements, with communication and support needs for this user population, e.g. task completion lists, help and tutorial wizards, etc. Note that these requirements are present for all users, in particular to support learning and help systems; however, in assistive technology,

Table 2 Requirements framework for the TAL e-mail application illustrating one personal profile out of six users

Requirements layer and goals	Functional requirements	Temporal change	Spatial context
Group: cognitively disabled users	Compose, send, read, delete, reply to message, incoming message summary, identity of sender	More sophisticated functions may be needed as confidence grows, e.g. save, print message, folders, copy, forward, add address	Privacy requirements: use in public place may be inhibited
User characteristics: Michael's problems are attention, initiation, short-term memory	Reminders, hints, simple commands and displays	Some improvement learning more commands	None
Personal goals: learn new skill (e-mail); communicate with friends and family; improve social skills; expand range of social partners	Supportive learning environment, training. Controlled address list, e-mail filters. Style checkers on message, review function. Solicit new partners	As skills improve may need to add: <ul style="list-style-type: none"> • New functions • More partners in list • Improve style checkers 	Failure anxiety in public use

communication and learning support, requirements assume a more prominent role. We have not illustrated requirements in detail for other stakeholder groups, but care providers and family respondents are two key groups who have requirements for e-mail communication and abilities to monitor the cognitively impaired users.

Temporal and spatial implications For the whole group requirements were expected to change over time as the users might respond to the e-mail system with better social communication; however, analysis was more logically applied in the user characteristics and personal goal layers.

Monitoring and adaptation implications Change over time suggested the need for monitors to keep users on task with reminders and task-completion lists, as well as tracking their performance to detect improvement in e-mail use. However, these concerns were refined in the following layers.

Individual characteristics Requirements were gathered by the CORE method [24], [39] which uses a combination of focus groups, user interviews, scenario-based exploration with prototypes, coupled with expert judgement and performance tests for application and computer operation tasks. This creates a capability profile which is employed to select a sub-set of the group communication and learning support requirements to match the individual, as “prescription” for his/her profile. The requirements have to be inferred from the clinical profile, so user characteristics are similar to NFRs in that they have to be satisfied by computer support functions or training. In the e-mail domain we developed a 50-item skill inventory that complemented user characteristic requirements, rating each skill on a 1–3 scale as missing/sporadic/possessed [24]. In Michael’s case (not his real name) attention focus and short-term memory problems indicated requirements for reminder agents to help with task initiation, with an agenda of task steps. Some examples of his skills profile were assumed (possessed) skills for motor coordination using the keyboard and mouse, and visual acuity suitable for standard VDU technology. Mary’s profile notes problems of social impulsiveness, and language and learning problems that indicate the need for a supportive system that also vets her messages to check for inappropriate sentences and words.

Temporal and spatial implications For both Michael and Mary their performance was expected to improve over time as they learned the e-mail system and as their motor coordination skills improved with practice. There were no particular spatial considerations.

Monitoring and adaptation implications Change over time suggested the need to monitor users’ time and frequency of sessions, as well as tracking response times to detect improved motor coordination. Since monitoring

had privacy implications, these were discussed with the users and monitoring only proceeded with their permission. No requirements for automatic adaptation were planned because the data had to be interpreted in light of the users’ clinical profiles by a skilled medical expert.

Personal goals These are refined into specific requirements. In some cases the goals are attainment targets for group requirements; however, others need design suggestions from the requirements engineer. Furthermore, some solutions may be manually implemented in the social system, such as soliciting new e-mail partners who will then be registered by adding them to Michael’s recipient list. Personal goals can be non-functional in nature in the sense that they set aspirations which Michael wants to achieve. The attainment scale for achieving the overall personal goal “to use e-mail” was set in consultation with the user, to produce the following list, with level 1 representing “not attained” and level 5 “fully attained”:

- | | |
|---------|---|
| Level 1 | not able to learn how to use e-mail even after 3 months of training and practice. |
| Level 2 | can e-mail, but only with continuing prompting and help by a co-present helper. |
| Level 3 | can e-mail with some prompting and help (machine-based, and care provider on call). |
| Level 4 | will can e-mail with no prompting or help. |
| Level 5 | can teach others how to e-mail. |

Mary’s personal goals were motivated by the desire to use e-mail to contribute to a newsletter circulated among her community. Her goals are illustrated within the attainment level framework as follows:

- | | |
|---------|---|
| Level 1 | not able to learn how to use e-mail. |
| Level 2 | will be able to write and e-mail an opinion on a topic to friends. |
| Level 3 | will be able to write an e-mail opinion that meets style constraints and submit it to the editor. |
| Level 4 | will have her (e-mail) letter published in the newsletter. |
| Level 5 | will be invited to write a letter or opinion. |

Mary’s goals need to be refined by the requirements engineer into functions that support her needs, for instance style advisor and checkers to help her improve the quality of her opinions.

Temporal and spatial implications For both Michael and Mary the temporal implications were described in the monitoring levels described above. There were no particular spatial considerations, although the issue of public use in the common room or more private use in her own room was an issue for Mary. Space and availability of the computer in the common room settled this issue.

Monitoring and adaptation Change over time can be anticipated as users acquire skills, although the exact nature of change can only be measured at the individual level. The time-scale for attaining goals at the personal level tends to be shorter term, but this is not always so. Change in user characteristics can be rapid if a treatment is effective. For this application there are fewer implications for spatial change. However, two contexts of use are anticipated: one in the user's home, while the other will be on a shared computer placed in the common room of sheltered accommodation. The latter scenario has implications for privacy since other people could read personal messages being composed by the user. There is no immediate technical solution for this problem, although a pragmatic work-around is to suggest e-mailing at quiet times in the day when few other people will be in the common room.

Change over time can be tracked at the individual level either by automated monitoring software, e.g. the REQMON system [40] which works in model-based mode by tracking changes in user behaviour and performance linked to a personal goal; or in pattern-based matching using data mining techniques to infer changes over time and probable causes. We also monitor by questionnaires and interview sessions with the users and care providers. Automated monitoring is economical but limited in the inferences that can be made from low-level data streams, i.e. messages sent, distribution of message by partners, words per message, latencies and time to compose messages, etc. Currently we intend to use latencies to drive a simple task list reminder function, while other monitored data, e.g. errors, functions used, frequencies and distribution of messages sent will be collected for manual analysis. Interviews will be employed to monitor the change in personal goals over time; for instance as Michael's confidence grows, new partners and functionality (print and save messages) may be added to his configuration. In Mary's case the editor will be added to her mailing lists once she has attained sufficient competence for her level 3 personal goal. Adaptations over a longer time period are anticipated in the individual's e-mail recipient lists and the e-mail functions as their confidence grows. In the shorter time-scale the system needs to adapt its reminding function to help the individual users to complete mes-

sages, while not becoming annoying and intrusive. Getting the level of system initiative right is a difficult judgement which we believe has to be tuned with advice from the users. Given the simplified e-mail system we are using, most adaptation can be anticipated at design time; however, we do not claim perfect foresight so the architecture of the e-mail system has been constructed to enable new functions (e.g. forward message, multiple replies) to be added without major redesign.

5.2 Navigation support system

The requirements for the second application are summarised in Table 3. In this case the top-level goal was to help individuals make more unassisted journeys by all modes of transport: bus, taxi, walking, or community mini-van. The personal goals for six users in this case study were to increase their independent mobility for social meetings with friends, shopping, recreation (cinema, bike rides) and volunteer work.

Group requirements Possible requirements for the whole group are for navigation support, e.g. in vehicle route finding, providing maps, route-following instructions, location of self, re-orienting help when lost, and progress indicators, pathway history, frequently followed routes, highlighted key landmarks, and viewpoint controls in advanced systems. Navigation is a demanding task for most people. For our users only a sub-set of this functionality would be appropriate to keep the system simple and easy to learn.

Temporal and spatial implications For the whole group, requirements were expected to change over time as the users learned to navigate more effectively.

Monitoring and adaptation implications Change over time suggested the need to monitor the number of journeys, their duration, successful completion, etc., as well as tracking different types of journeys. However, these concerns were refined in the following layers:

Individual characteristics Requirements for individual users were captured using an adaptation of the CORE

Table 3 Requirements for the GO navigation support application

Requirements layer and goals	Functional requirements	Temporal change	Spatial context
Group: enhance independent mobility	Schedule reminder, route instructions, route map, bus transport guide	Learning effects: add new routes, select route	Appropriate instructions, map display, re-orientation help
User characteristics: short-term memory loss, forgets purpose, location, time and destination	Schedule reminders, route following walking, bus journey instructions, destination reminder	Add new routes, instructions for new modes of transport	On route tracking for appropriate instructions, destination reminders, off route re-orientation
Personal goals: social meeting; recreational trips; journeys to job	Fixed route support, reminders for regular journeys, new routes	As above. Add new routes	As for user characteristics plus privacy of instructions in public places

method with Wizard of Oz simulation of the route-following task. Normal and TBI users were asked to follow the navigation instructions given on a Personal Digital Assistant (PDA) display with speech in an earpiece. Two observers followed the user, one simulating the navigation instructions and dialogue using wireless communication between PDAs, with the other acting as an ethnographer recording user behaviour, problems and carrying out problem diagnosis interviews. This produced individual requirements profiles as well as contributing to the group requirements gathering. The individual user focus is illustrated with requirements for John, whose clinical characteristics are severe short-term memory loss leading to attention failure, forgetting to turn up for regular trips, the purpose of a journey and the destination.

Temporal and spatial implications The spatial and temporal implications for John within each journey were deviations from the desired route. In the longer term more complicated and longer journeys were expected to be undertaken as John became familiar with the technology as well as learning the local geography.

Monitoring and adaptation implications Change over time suggested the need to monitor the pathway on each journey, and the duration of stages, so reminders of the route could be given and directions for regaining the route if John lost his way.

Personal goals Requirements to fulfil John's personal goals are clear instructions for a limited set of routes, a bus journey guide, reminders for timetabled journeys and monitors with reminders when the destination is close. John's personal goals are increased independent mobility for meeting his family and for recreation (sightseeing, cinema, festivals and undertaking his volunteer job at the YMCA). These goals specify the instructions and information content (route maps, landmark cues); however, while some routes are fixed (family, job) the recreation trips are time- and location-variable, so there is a need for his care provider to configure new routes.

Temporal and spatial implications The spatial and temporal implications for John's personal requirements were set by his attainment levels which set out targets for successful journey completion and the types of journeys which could be undertaken. Over time John wished to make journeys for an increasing variety of different purposes as described above. More adventurous journeys involved navigating in more complex spaces (public festivals, cinemas).

Monitoring and adaptation Time-sensitive requirements are task initiation reminders since subjects often forget to make regular scheduled journeys and mistake their destination. Temporal change also has to account for learning effects, adding new routes, and more

importantly the dynamic change of instructions according to the subjects' location. Location is vital for delivering appropriate instructions, and triggering re-orientation help if users deviate from the expected route. Requirements for the navigation system were several monitors for location and progress tracking using GPS, progress tracking with a pedometer when walking and simple location awareness by light intensity and noise levels to improve reasoning about location when off-track (e.g. noisy or quiet streets, in building or outside). Use of multiple sensors creates a further requirement for an interpreter function which integrates all the sensor inputs checks with the route model and outputs an "on path/near path/not on path" message with a confidence rating. The more difficult task is to infer the user's location in the near path/not on path conditions. The adaptive sub-system is a key component because of the need for re-orientation and error correcting dialogue, so system initiative will be used to help the user to regain the route, with instructions on looking for local landmarks and re-orientation. This will be backed up with a simple "find me" button for emergencies, connected directly to a care provider who can locate the user's approximate position with GPS. We are also considering voice communication to the care provider to help reassure the user.

For user characteristics and personal goal, temporal change in skills is anticipated. John is relatively young so there is a reasonable chance that he may improve with experience, hence the system may need to be adapted with more advanced facilities and more routes. Requirements for a route editor were added so care providers can adapt the system with additional routes. The system architecture is designed to allow new monitors to be added, although the interpreter will need to be changed to adapt a multisensor integration algorithm. Spatial context implications are similar for all users, namely the need for tracking to make sure the individual is on the route; re-orienting help is given if not. When we proposed technical solutions to personal requirements, it became clear that these solutions had different impacts on individual users, and on other stakeholders. This led us to consider how the costs and benefits of different technical solutions might be assessed.

6 Cost-benefit analysis

Personal requirements at all levels imply a cost to individuals, either in customising the system themselves or in adapting their behaviour to a system that has changed automatically. For example, the costs of learning and using a computer system can outweigh the benefits for our TBI users even though the social and personal self-esteem rewards for many individuals are very important. The chosen implementation pathway also influences the distribution of costs and benefits between users and designers; for instance, configurable applications

outsource design effort to the user. Furthermore, trade-off between the user's costs and benefits influence other stakeholders, particularly the care providers in our case studies. A conventional RE approach would be to use design rationale or goal modelling to investigate such trade-offs; however, such approaches do not afford comparison. To address this problem we propose a simple cost-benefit modelling technique that supplements design rationale or goal modelling. The technique consists of the following:

- Estimated benefits of achieving the desired goal. In our personal RE framework this will be assessed as a collective group or individual benefit according to the type of goal.
- Costs of each design alternative (sub-goal/functional requirement) proposed to achieve the goal. We distinguish between costs of learning to use the system, operational costs imposed on the user, and costs imposed on other stakeholders by that solution.
- Cost penalties if the solution alternative in question does not achieve the goal. This is a measure of reliability, as a composite assessment of the probability of failure, the severity of impact of failure, and the cost of recovery.
- Other NFRs can be added as potential costs if they are infringed. Alternatively, the achievement of NFRs may be considered to be part of the benefit of achieving the overall goal.

Benefits are assessed by first asking users to estimate the potential satisfiability of the goal by the proposed technical solution (on a 1–10 scale). Benefits may also be assessed for user characteristics by expert judgement of value to the individual and how the system's functions might fulfil their motivations, e.g. self-esteem, learning, social inclusion, etc. User costs are learning to use the system, operational effort, and cost of error recovery, which are added to indirect costs of customisation effort, and learning how to use customisation facilities. Costs may be estimated by requirements engineers, based on observations of user problems, or by interviews with users to acquire their perception of learning and operational cost.

Benefits estimate the potential for a given design option achieving the goal. A simple percentage-satisfaction metric is calculated using the benefit estimated on a 1–100 scale, depending on the probable contribution of the solution alternative to achieving the higher-level goal, minus the sum of all the costs:

$$\% \text{ Satisfaction} = \text{Benefits} - \text{sum}(\text{costs}^{1-n})$$

where $\text{sum} \max(\text{costs}^{1-n}) = 100$.

The analysis also guides functional allocation decisions, i.e. whether to automate a requirement; implement it partially and supplement the computer system by training the user; or allocate the duty to another stakeholder.

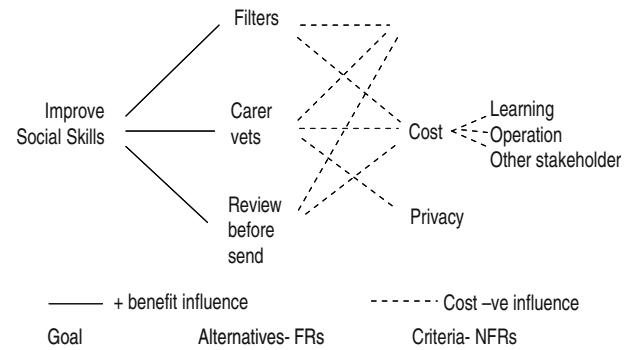


Fig. 4 Design rationale diagram for the goal Improve Social Skills, using gIBIS notation [41]. The *solid line* represents the positive influence of benefit, and the *dashed line* the negative influence of cost

For example in Fig. 4 the options for achieving the Improve Social Skills goal for Michael are to implement filters to trap any anti-social words and phrases with a stop list and style checker, or to add an enforce-review-before-send, so Michael has to read and check his message before sending, or to send his message to his care provider so he/she can check it for any offensive words or phrases. The NFRs or argument criteria that assess the trade-off are the costs of implementation, the reliability of attaining the goal, and avoiding infringing Michael's privacy.

A sample of the cost-benefit analysis for the Improve Social Skills goal is given in Table 4. Benefits are estimated on a 0–100 scale, while costs are scored on 100/cost variables $\times n$, so in Table 4 each cost is rated on a 1–20 scale, yielding a potential net benefit of zero with maximal costs and maximum benefit, and a negative trade-off if costs are high and benefits small. Operation and learning costs map to the design rationale diagram, as do reliability of the option being effective, and privacy. The care provider vetting outgoing e-mails has the highest probability of achieving the goal, compared to filters which have a 50% chance of preventing unwanted e-mails, while the review-before-send relies on Michael's diligence in checking his messages before sending. This was judged to be more risky. For Michael the costs of operation and learning were small for the filters option,

Table 4 Cost-benefit analysis for the Improve Social Skills personal goal

Alternative solutions	Filters	Carer vets	Review before send
Benefits	50	80	30
Criteria: costs			
Operation	5	0	6
Learning	2	0	7
Reliability	9	5	10
Other stakeholders	0	20	0
Privacy	0	20	0
Net cost-benefit	34	35	7

but more effort would be required for reviewing messages. The care provider option was cost-free for him, but a burden for the care provider. The potential impact of errors reflected the cost of not achieving the goal and the costs of recovery, i.e. apologising to offended friends and family. The downside of the care provider option becomes clear when Michael’s privacy is considered, since all his e-mails would be available to the care provider. Furthermore, this option imposes a considerable workload on the care provider who has to check all outgoing e-mails and edit them. These penalties reduce the net benefit of this option to nearly the same as the filters option even though it has a higher initial benefit. While the estimates are subjective judgements, the value of the cost–benefit analysis is in using it as a sensitivity analysis tool to explore requirements trade-offs. A similar trade-off analysis occurred for Mary’s improve letter quality personal goal. In this case the design options were to screen output with automated style checkers, provide an online wizard to suggest topics and provide template letters, or to ask her friends to help as tutors. The last option was selected with the style checker, because these provided the better benefits, even though incurring more costs for the friend stakeholders. Help from her friends improved her motivation and confidence, in a manner that no computer wizard could achieve. Privacy was less of an issue with friends than it would have been with a professional care provider.

The design rationale for a second goal, Keeping the User on Track so their attention does not wander, is shown in Fig. 5. The issue in this case is helping the user to complete the e-mail composition task, when users’ attention frequently wanders. Design alternatives are to display a task-completion agenda, similar to reminders in standard office products, to make the system actively monitor and remind the users if no action had been taken after a set time period. Finally the care provider could monitor the user via a video link. The criteria or costs by which the alternatives can be judged follow the pattern we describe earlier. The cost–benefit analysis for this user goal is given in Table 5. The benefits of the active reminder and the care provider monitoring performance were judged to be similar, since the care provider would have to monitor the users continuously to achieve the goal, and this was open to some doubt. Operation and learning costs were minimal for the agenda and reminder

Table 5 Cost–benefit trade-off analysis for the Keep on Track goal

Alternative solutions		Agenda	Active reminder	Carer monitor
Benefits		30	70	80
Criteria: costs	Operation	2	1	0
	Learning	3	1	0
	Reliability	4	10	2
	Other stakeholders	0	0	20
	Privacy	0	3	5
Net benefits		21	55	53

options; however, reliability impact of reminders was often in doubt since setting the frequency and latency for system intervention was not easy to specify. Privacy was an issue for the reminder and care provider options since both would disrupt the user’s task or thought processes. The net balance was similar for the care provider and reminder options for this goal, but the reason why they differ becomes transparent. In this case the technical solution for an automatic reminder was chosen, while the care provider-vets option had already been chosen for the Improve Social Skills goal.

Space precludes reporting further analysis of individual goals. For group requirements the technique clarified the trade-offs between different design solutions and costs imposed on primary users and other stakeholders (i.e. care providers). Customisation (e.g. add new filters or e-mail correspondents) by the users was ruled out because the additional learning and operating costs reduced the percentage-satisfaction to zero.

Considerable help would have to be available from the designers or care providers (thereby increasing their costs) to make customisation effective, so the overall balance was not favourable even though it was still desirable to increase the fit between system functions and the user’s characteristics. However, some set-up configuration costs would be necessary for the solution to scale beyond the immediate locality where the designers can provide support. The cost–benefit analysis can be applied to the care provider stakeholders, assuming a large benefit will be altruistic motivation to overcome their costs of learning and setting up the system. While this assumption might be true for family and very dedicated professional care providers, it is doubtful for other less committed care providers. This outcome led to the investment to reduce learning and configuration burden as far as possible by CD-ROM advisors and training programmes and trying to distribute the support costs among a large stakeholder group.

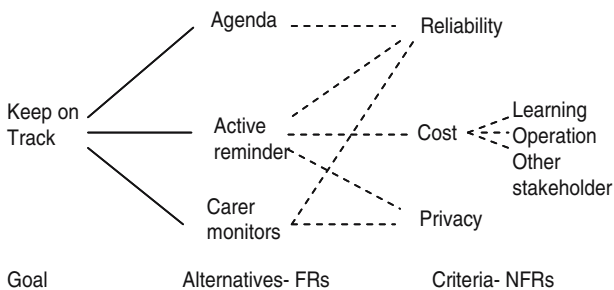


Fig. 5 Design rationale for the Keep on Track user goal

7 Reflections on method use

Since the case studies focused on the individual user most of the experience, not surprisingly, involved user characteristics and personal goals. The method provided a framework of questions that drove the

requirements investigation to consider spatial, temporal and individual concerns, followed by requirements for monitoring and adaptation. As such it functioned more as a checklist of issues rather than a prescriptive “cook book” of steps to follow. However, the setting of attainment levels, monitoring against them and the use of design rationale for trade-off analysis were specific techniques which guided requirements investigation more precisely. Some reflections on our experience with the trade-off analysis are that goals between different layers can conflict, e.g. the user can set a personal goal that is unrealistic in terms of their user characteristics, derived from expert clinical judgement. The cost–benefit estimation was very subjective so the numbers were not particularly trustworthy; however, the process did lead to useful debate for comparing trade-offs in the design rationale diagrams. We did consider ranking costs and benefits but this obscured the overall cost–benefit assessment. Finally, the method added a different perspective to conventional RE goal analysis by focusing on quantification of individual achievement, rather than collective functional requirements.

8 Discussion

This paper has proposed a new framework which introduces the concept of the individual user and context as a focus for requirements engineering. The method has synthesised and developed our previous research on requirements monitoring [6], [39], socio-technical requirements and functional allocation [18]. The PC-RE method extends current RE methods, such as Volere [2] and ScenIC [14], by introducing requirements for individuals and providing guidance for requirements analysis that accounts for variation in requirements over space and time. PC-RE has drawn on influences from domain analysis methods for the concept of variation points familiar in product-line requirement methods and extends the stakeholder concepts present in most RE methods [2], [41] to individual users. In assistive technology applications, the focus on individuals is a necessary consequence of the wide range of abilities which affect not only human computer interaction but also matching functional requirements to individual capabilities. We argue that this approach generalises beyond assistive technology applications. Educational technology is one area where individual characteristics are important influences on design as learner profiles; another area is groupware applications where individual profiles often determine security and privacy requirements. However, with the growth in customised products, personal RE is becoming generally applicable to a wide range of office and home/entertainment applications.

Personal RE has extended the boundaries of skills-preference analysis [42] by connecting requirements to architecture, introducing cost–benefit analysis, and drawing the distinction between expert assessment of user characteristics and individually held goals. Hui

et al.’s [42] framework, based on *i** and goal modelling, accounts for the skills profiles using a similar ontology to ours, and their preferences are similar to personal goals although they treat them only as soft goals (non-functional requirements) whereas we distinguish between the functional goal and quality in the levels of attainment. We have integrated advice from studies in human computer interaction where requirements-related issues of culture, localisation and context-aware systems have received considerable study. However, further work is required to explore the implications of context requirements in different types of mobile and context-aware applications. For instance, mobile applications with safety and security NFRs will require more high fidelity monitoring, and the nature of adaptation will be more challenging compared with notifier systems for general business tasks. The implications of requirements on system architecture has only started to be explored in security [43]. The PC-RE framework and method draws attention to the need to analyse latent requirements for monitoring processes, interpreters and design for adaptation. As requirements are increasingly seen to be deferred over time or mutable over space, architectural design will become a foreground requirements concern, to specify context-aware and adaptable systems. Architectural trade-offs will need to be assessed to judge the mix of adaptable components (e.g. changed at design time) versus adaptable systems [19] in which the change process is automated with monitors and interpreters executing adaptations. Another debate is between system initiative leaving the user in control of adaptation, or a combination in mixed initiative systems [20].

The scenario-based approach in PC-RE extends methods such as ScenIC [14] in which obstacles are posited in the system environment as a stimulus for validating requirements. By varying scenarios over time and space, new obstacles and challenges can be identified. More formal reasoning could be applied to temporary deferment requirements, and our work with the KAOS method [34] has taken some initial steps in this direction by exploring specification with temporal operators: always, in the future, unless, eventually, etc. However, formal reasoning over space and time is more complex due to the state-space explosion when multiple spatial contexts need to be considered, so formalisation of such requirements may not be productive.

Our technique augments trade-offs between NFRs using decision table-style representation by specifically considering costs imposed on users and other stakeholders, which can then be integrated into more general NFR trade-offs. It also focuses on the evolution of requirements where personal goals tend to be shorter term than generic requirements and user characteristics, although in some cases personal goals may take a long time to achieve. Setting attainment levels for personal goals was prompted by the GQM framework [44], although we have extended the metrics to include assumptions necessary for level of attainment. These dependencies could be further formalised in goal-oriented

modelling methods such as KAOS [34]. Another contribution of the trade-off analysis proposed in PC-RE is explicit consideration of socio-technical solutions, which challenges the notion that software automation is the default answer to a problem. PC-RE therefore reinforces the systems view of requirements [43].

In conclusion, personal and contextual requirements engineering has extended RE methods to account for the individual user and the context in mobile communication and global user interfaces. In many applications where the computer's role is to influence the user's behaviour and knowledge (e.g. training, education, augmented communication, mediating collaboration, games, entertainment, environmental control), personal goals and user characteristics will assume considerable importance. The application of our approach in the field of assistive technology has produced not only more detailed and better structured requirements but also insights into software and socio-technical systems design, which need to be resolved early in the requirements process. However, this work is a first step towards development of a more widely applicable method, so our next step is to test its generality in other domains, such as educational and home applications.

Acknowledgements This research was partially supported by the US Department of Education and National Institute on Disability and Rehabilitation Research *Think and Link* project: <http://www.think-and-link.org/>

References

- Sommerville I, Kotonya G (1998) Requirements engineering: processes and techniques. Wiley, Chichester
- Robertson J, Robertson S (1999) Mastering the requirements process. Addison Wesley, Harlow
- Newell AF, Gregor P (2000) User sensitive inclusive design: in search of a new paradigm. ACM Conference on Universal Usability. ACM Press, New York, pp 39–44
- Clements P, Northrop LM (2001) Software product lines: practices and patterns. Addison-Wesley, Reading
- Yu Y, Leite J, Mylopoulos J (2004) "From goals to aspects: discovering aspects from requirements goal models," 12th IEEE international requirements engineering conference. IEEE Computer Society Press, Los Alamitos, pp 38–47
- Fickas S, Feather MS (1995) "Requirements monitoring in dynamic environments," 1995 IEEE international symposium on requirements engineering (RE '95). IEEE Computer Society Press, Los Alamitos, pp 140–147
- Cheverst K, Davies N, Mitchell K, Friday A, Efstratiou C (2000) "Developing a context-aware electronic tourist guide: some issues and experiences," Conference on human factors in computing systems, ACM Press, New York, pp 17–24
- Abowd GD, Mynatt ED (2000) charting past, present and future research in ubiquitous computing. ACM Trans Comput Hum Interact 7:29–58
- Chung L, Nixon BA (1995) "Dealing with non-functional requirements: three experimental studies of a process-oriented approach," 17th international conference on systems engineering. Los Alamitos CA: IEEE Computer Society Press, pp 25–37
- Pohl K (1993) "The three dimensions of requirements engineering," CAISE '93, Springer, Berlin Heidelberg New York
- Roman GC (1985) A taxonomy of current issues in requirements engineering. Computer 18:14–23
- Zave P (1995) "Classification of research efforts in requirements engineering," 1995 IEEE international symposium on requirements engineering (RE '95). IEEE Computer Society Press, Los Alamitos CA, pp 214–216
- Potts C, Takahashi K, Anton AI (1994) Inquiry-based requirements analysis. IEEE Softw 11:21–32
- Potts C (1999) "ScenIC: a strategy for inquiry-driven requirements determination," 4th IEEE international symposium on requirements engineering. IEEE Computer Society Press, Los Alamitos CA, pp 58–65
- Marcus A (2000) Multimedia interface design studio. Random House, New York
- Norman DA (2004) Emotional design: why we love (or hate) everyday things. Basic Books, New York
- De Angeli A, Athavankar UA, Joshi A, Coventry L, Johnson GI (2004) Introducing ATMs in India: a contextual enquiry. Interact Comput 16:29–44
- Sutcliffe AG (2002) User-centred requirements engineering. Springer, London
- Fischer G (2001) User modeling in human-computer interaction. User Model User-Adapt Interact 11:65–86
- Lieberman H (2001) Your wish is my command: programming by example. Morgan Kaufmann, San Francisco
- Fischer G (1993) "Beyond human computer interaction: designing useful and usable computational environments," Proceedings of the HCI 93 Conference. Cambridge University Press, Cambridge, pp 17–31
- Folstein M, Folstein S, McHugh P (1975) Mini-mental state: a practical method for grading the state of patients for the clinician. J Psychiatr Res 12:1889–198
- Sutcliffe AG (2003) Multimedia and virtual reality: designing multisensory user interfaces. Lawrence Erlbaum Associates, Mahwah
- Sohlberg MM, Ehlhardt LA, Fickas S, Sutcliffe AG (2003) A pilot study exploring electronic mail in users with acquired cognitive-linguistic impairments. Brain Injury 17:609–629
- Sutcliffe AG, Ryan M (1998) "Experience with SCRAM: a scenario requirements analysis method," IEEE international symposium on requirements engineering: RE '98. IEEE Computer Society Press, Los Alamitos, pp 164–171
- Choi B, Lee I, Kim J, Yunsuk J (2005) "A qualitative cross national study of cultural influences on mobile data service design," 2005 Conference on human factors in computing systems, CHI 2005. ACM Press, New York, pp 661–670
- Krumbholz M, Maiden NAM (2001) The Implementation of ERP Packages in Different Organisational and National Cultures. Inf Syst 26:185–204
- Hofstede G (1980) Cultures consequences: international differences in work related values. Sage, Thousand Oaks
- Hall ET (1976) Beyond culture. Anchor Doubleday, New York
- Somers H, Whyman EK (1999) Evaluation metrics for a translation memory system. Softw Pract Exp 29:1265–1284
- Fitzpatrick G (2003) The locales framework: understanding and designing for wicked problems. Kluwer Academic, Dordrecht
- Dix A, Rhodden T, Davies N, Trevor J, Friday A, Palfreyman K (2000) Exploiting space and time as a design framework for interactive mobile systems. ACM Trans Comput Hum Interact 7:285–321
- Dey AK, Mankoff J (2005) Designing mediation for context aware applications. ACM Trans Comput Hum Interact 12:53–80
- Van Lamsweerde A, Letier E (2000) Handling obstacles in goal-oriented requirements engineering. IEEE Trans Softw Eng 26:978–1005
- Fickas S, Robinson W, Sohlberg MM (2005) "The role of deferred requirements in a longitudinal study of emailing," 13th IEEE international conference on requirements engineering. Los Alamitos, pp 145–154
- Lam W, McDermid JA, Vickers AJ (1997) "Ten steps towards systematic requirements reuse," ISRE '97: 3rd IEEE international symposium on requirements engineering. IEEE Computer Society Press, Los Alamitos, pp 6–15

37. Carroll JM (1990) *The Nurnberg funnel: designing minimalist instruction for practical computer skill*. MIT Press, Cambridge
38. Sutcliffe AG, Fickas S, Sohlberg MM, Ehlhardt LA (2003) Investigating the usability of assistive user interfaces. *Interact Comput* 15:577–601
39. Fickas S (2005) “Clinical requirements engineering,” ICSE 2005: 27th International conference of software engineering. IEEE Computer Society Press, Los Alamitos
40. Robinson W (2003) “Monitoring web service requirements,” 11th IEEE international requirements engineering conference (RE’03). IEEE Computer Society Press, Los Alamitos, pp 65–74
41. Conklin J, Begeman ML (1988) gIBIS: a hypertext tool for exploratory policy discussion. *ACM Trans Office Inf Syst* 64:303–331
42. Hui B, Laiskos S, Mylopoulos J (2003) “Requirements analysis for customisable software: a goals skills preferences framework,” IEEE joint international conference on requirements engineering. IEEE Computer Society Press, Los Alamitos, pp 117–126
43. Nuseibeh B, Easterbrook S (2000) “Requirements engineering: a roadmap,” international conference on software engineering (ICSE-2000). ACM Press, New York
44. Basili VR, Rombach HD (1988) The TAME project: towards improvement-orientated software environments. *IEEE Trans Softw Eng* 14:758–773