BEHAVIOUR & INFORMATION TECHNOLOGY
2020, VOL. 39, NO. 1, 27–46
https://doi.org/10.1080/0144929X.2019.1634153

Exploring design trade-offs for achieving social inclusion in multi-tiered design problems

Daniela Fogli[a], Antonio Piccinno[b], Stefan Carmien[c] and Gerhard Fischer[d]

[a]Department of Information Engineering, University of Brescia, Brescia, Italy; [b]Department of Computer Science, University of Bari, Bari, Italy; [c]Department of Computer Science, The University of York, York, UK; [d]Department of Computer Science, University of Colorado, Boulder, CO, USA

ABSTRACT

The digital age of the future is 'not out there to be discovered', but it needs to be 'designed'. The design challenge has to address questions about how we want to live, work, and learn (as individuals and as communities) and what we value and appreciate, e.g.: reflecting on quality of life and creating inclusive societies. An overriding *design trade-off* for the digital age is whether new developments will increase the digital divide or will create more *inclusive societies*. Sustaining inclusive societies means allowing people of all ages and all abilities to exploit information technologies for personally meaningful activities. Meta-design fosters the design of socio-technical environments that end-user developers can modify and evolve at use time to improve their quality of life and favour their inclusion in the society. This paper describes three case studies in the domain of assistive technologies in which end users themselves cannot act as end-user developers, but someone else (e.g.: a caregiver or a clinician) must accept this role requiring multi-tiered architectures. The design trade-offs and requirements for meta-design identified in the context of the case studies and other researchers' projects are described to inform the development of future socio-technical environments focused on social inclusion.

ARTICLE HISTORY
Received 16 November 2018
Accepted 14 June 2019

KEYWORDS
Design trade-offs; social inclusion; multi-tiered design problems; end-user development; meta-design; universe of one

1. Introduction

End-User Development (EUD) encompasses methods, techniques and socio-technical systems that empower end users to develop and adapt systems at use time, by carrying out activities that are traditionally performed by software developers at design time (Lieberman, Paternò, and Wulf 2006). Problems in understanding social practices in specific application domains and coping with their dynamicity can be addressed through an EUD approach. EUD requires design environments to support end users in modifying and creating software artefacts thereby developing their own skills and knowledge (Paterno and Wulf 2017). Scaffidi's work (Scaffidi 2017) provided evidence that workers performing EUD activities obtain economic benefits by earning 10% more than peers who are not able to perform EUD activities.

The perspective illustrated so far mainly considers EUD as fundamental for all situations where end users create or adapt software artefacts for *personal use*. For instance, (Fischer, Nakakoji, and Ye 2009) describes the situation of a geoscientist who decided to spend three months in learning a programming language to become capable of developing software for analysing data he collected. This may occur in a variety of application domains (e.g.: business processing and management (Hermans, Pinzger, and van Deursen 2011), medical diagnosis (Costabile et al. 2006), interaction design (Won, Stiemerling, and Wulf 2006)), where end users are *domain experts* that, to cope with their specific problems, adapt their software artefacts or create new ones. In particular, with reference to workgroups, Gantt and Nardi call such a domain expert *local developer* or *gardener* (Gantt and Nardi 1992), namely a person who customises a software environment and creates programmatic extensions of applications for her/his purpose that, possibly, will be made available to other users working in the same group with a shared objective. Other researchers have called this 'active' end user with other names: *power user* in (Repenning and Ambach 1997), *professional end user developer* in (Rosson et al. 2007), *end-user developer* in (Fogli and Piccinno 2013), *bricolant bricoleur* in (Cabitza and Simone 2015).

We choose here the term *end-user developer* to indicate a domain expert, not knowledgeable in information technology, who is called on to tailor, extend or create a software artefact. In this paper, we take a different perspective on EUD: we are interested in those contexts where the communities of end users and end-user developers are kept separated. The former are the actual end

CONTACT Daniela Fogli daniela.fogli@unibs.it Department of Information Engineering, University of Brescia, Via Branze 38, 25123 Brescia, Italy

users that cannot be required or are not willing to carry out EUD activities due to their specific goals, interests, and abilities. The latter are 'proxy end users', namely caregivers, healthcare professionals, teachers, and so on, who are called on to play the role of end-user developers and create and/or adapt software programs for the sake of other people (the end users) by means of EUD environments and tools. This situation has been characterised in (Carmien et al. 2005) as a *multi-tiered proxy design problem*. Most of such multi-tiered proxy design problems usually represent a 'universe of one' problem (Fischer 2001): a solution designed for one person rarely works for another person. This is true for instance in cognitive disability support (Carmien et al. 2005) or in different kinds of rehabilitation activities (Tetteroo et al. 2015). *Social inclusion* is often the main objectives of socio-technical environments designed and developed to cope with these kinds of problems (Siira and Heinonen 2015). It aims at improving the participation of people in the society, independently of their physical or cognitive characteristics, age, gender, profession, living place, etc.

Meta-design is the methodological framework proposed for empowering domain experts in the continuous development of personally meaningful socio-technical systems that foster social inclusion. Meta-design allows modelling problems in innovative ways and putting domain experts in charge with the help of socio-technical environments enabling EUD activities (Fischer and Giaccardi 2006). It promotes 'design for design after design' instead of 'design for use before use' (Binder et al. 2011), that is, it fosters the design of open systems that end-user developers can modify and evolve at use time, also for the sake of other people, in order to improve their quality of life and favour their inclusion in the society.

In this paper, we will analyse authors' experience in the meta-design of socio-technical environments in different application domains where inclusion of disabled and elderly people is considered; the main goal is to explore the trade-offs affecting this kind of multi-tiered proxy design problems; we will eventually integrate this analysis with the results of a survey study carried out with the help of research scholars addressing similar problems.

In summary, the paper contributes to the research on meta-design for social inclusion as follows:

- identifying design trade-offs underlying the wicked problem of building socio-technical environments supporting a 'universe of one';
- exploring which configurations require a meta-design approach (and which ones do not) and the role of domain experts and their expertise in the initial design;
- deriving new guidelines for meta-design supporting social inclusion.

The paper is organised as follows: Section 2 discusses related work on EUD methods and techniques with a specific emphasis on meta-design frameworks for multi-tiered proxy design problems; Section 3 describes our research methodology, based on reflective practice and a survey with research scholars in the considered field; Section 4 presents three case studies derived from the research experiences of the authors and discusses lessons learnt from related design activities; Section 5 presents a survey study with colleagues that have been involved in the design of EUD environments for multi-tiered design problems; Section 6 discusses design trade-offs, and proposes a set of guidelines for the meta-design of EUD solutions for social inclusion in multi-tiered architectures; Section 7 presents the limitations of the research; and Section 8 concludes the paper.

2. Related work

User-centred design (Norman and Draper 1986; Abras, Maloney-Krichmar, and Preece 2004) and *participatory design* (Bødker and Grønbæk 1991; Schuler and Namioka 1993) approaches have been explored for many years in human–computer interaction (HCI) to support the development of information technology (IT) that satisfies users' personal and work needs and increase users' acceptance of technology (Davis 1989). These approaches are based on the idea that user involvement at design time, through direct observation, interviews, focus groups, and prototype evaluation, is crucial to inform designers about users' languages, notations, backgrounds, capabilities, and tasks.

In all design processes two basic phases can be differentiated: *design time* and *use time*. *User-centred* and *participatory design* are primarily related to design time: system developers create environments and tools for the world as *imagined* by them to anticipate users' needs and objectives. But despite the best efforts at design time, systems need to be evolvable to fit new needs, account for changing tasks, deal with a great variety of subjects and contexts, and incorporate new technologies. *Meta-design* (Fischer and Giaccardi 2006; Fischer, Fogli, and Piccinno 2017) fosters the design of *open systems* that can be modified and extended by end users at use time. It represents a theoretical framework that allow end users (and other stakeholders) to become end-user developers in dynamic application domains (Carmien 2016) by creating the socio-technical conditions for empowering the owners of problems to participate in system evolution. End-user development activities range from simple selection among alternative behaviours already available in the artefact (customisation) to software program creation carried out through different techniques (Barricelli et al. 2019).

In most existing approaches, emphasis is given to end users developing software per se and not for third or public use. However, there are also a variety of projects that can be framed in the category of *multi-tiered proxy design problems* (Carmien et al. 2005; Carmien and Fischer 2010), where end users and end-user developers assemble or make up different communities with different goals, skills, and competencies. In these cases, end-user developers modify or create software artefacts for the benefit of end users. For example, in the education field, teachers (as domain experts) create digital materials for their students (the actual end users). The WEEV system (Marchiori et al. 2012) allows teachers to play the role of end-user developers in the creation of educational games with domain-specific visual languages. In the cultural heritage domain, the visitors of interactive exhibitions (end users) are distinguished from cultural domain experts being in charge of developing the interactive exhibitions (websites, mobile guides, or more advanced interactive systems). Design environments for cultural domain experts are for example presented in (Celentano and Maurizio 2011; Ardito et al. 2012; Fogli et al. 2018).

Multi-tiered proxy design problems are also encountered in all those situations where the end users cannot participate in EUD activities, but their caregivers, therapists, and assistants must do it for them. This is the case of *Assistive Technology (AT)* tools where domain experts create digital artefacts that end users cannot do by themselves, but that play a fundamental role to foster social inclusion. According to (RESNA 2019), AT is a technology designed to be utilised in an AT device, that is 'any item, piece of equipment, or product system, whether acquired commercially, modified, or customised, that is used to increase, maintain, or improve functional capabilities of individuals with disabilities'.

XOOM (Garzotto et al. 2017) is an EUD tool that allows therapists to create applications of Wearable Immersive Virtual Reality targeted to children with neurodevelopmental disorders. Since storytelling plays an important role in promoting both high-level and basic skills, XOOM can be used to create social stories able to foster attention skills and cause-understanding capability.

Training social skills to autistic children with humanoid robots is the main objective of the WikiTherapist project illustrated in (Barakova et al. 2013; Buchina, Kamel, and Barakova 2016). In this project, therapists use the TiViPe graphical programming environment to personalise scenarios for humanoid robots defining their expected social behaviour. Preliminary tests showed that this system could be effectively used by therapists for the personalisation of existing scenarios, while the creation of new scenarios and new behavioural components will require additional investigation. Interestingly enough, from the discussion with therapists, it emerged that they perceive the need of more control over the robot compared to other needs (safety, natural interaction, etc.), thus confirming the importance of studying suitable EUD techniques in this field.

TagTrainer (Tetteroo et al. 2014) is an EUD platform that supports physical rehabilitation after strokes, multiple sclerosis, and spinal-cord injuries. In this case, the patients are the actual end users of the system, and physiotherapists play the role of end-user developers. TagTrainer is composed of interactive boards, an interface to manage personalised series of exercises, and a visual programming tool to support authoring and modification of exercises by physiotherapists. The introduction of TagTrainer in clinical practice illustrates the variety of issues that influenced therapists' decisions to engage or not engage in EUD, providing evidence that the main issues are concerned with clinic's management and lack of time, rather than with technical aspects (Tetteroo et al. 2015).

In all these projects, the digital artefacts were used by people that did not participate in the development process and who expected interaction style and contents tailored to their needs, preferences and skills. This means that not only the EUD tools supporting end-user developers must fit their characteristics, skills, expectations, and background, but also the artefacts they create through EUD must fit end users' requests.

3. Research methodology

The research question at the basis of this work is to explore the design trade-offs in the meta-design of socio-technical environments for social inclusion. To this end, we first analyse three different projects related to the authors' research experiences, each one aimed at addressing a multi-tiered proxy design problem in the AT domain: cognitive disability support, mobility services and smart objects for elderly people. In particular, we carry out a 'reflection-on-action' activity (Schön 1983), that is, a reflection about what has been performed from the technical and social points of view to design and deploy the various socio-technical environments. We recall the choices that were made in the iterative development activities as a consequence of the evaluation with end users and domain experts, by underlining how the meta-design framework and EUD environments were necessary for the success of the projects. We finally report on the lessons learnt from the case studies, which may inform future designers of further socio-technical environments for social inclusion.

Then, we illustrate a survey study based on a questionnaire administered to research scholars who addressed multi-tiered proxy design problems; the goal was expanding the analysis to other application domains and collecting information, opinions, and comments from different research groups about their design experience, the involved stakeholders, and the benefits, issues and challenges they encountered. The lessons learnt from the survey study were successively triangulated with the outcome of the reflective exercise on our own projects, allowing us to identify the design trade-offs and define the guidelines discussed in Section 6.

4. Case studies of multi-tiered design problems

This section describes and analyses in detail three case studies derived from the direct design experience of the authors in different application domains where the goal was fostering social inclusion. These projects address multi-tiered proxy design problems involving the following stakeholders:

- *end users*, who are unable to completely describe their needs and design requirements;
- *end-user developers*, who are able to articulate what should be developed, even if they have no software programming competencies;
- *software developers,* who know how to develop complex systems, but lack the detailed knowledge of end users' needs, by being no domain experts as the end-user developers are.

The first two roles form a *dyad* (Carmien 2016) where the end user is a person who uses the AT tool and the end-user developer is a caregiver (or, in general, the owner of the knowledge in the domain, expert in the needs and skills of the end user and a more technically savvy member among the two), who is in charge of creating, customising, and evolving the AT tool at use time.

4.1 Cases

4.1.1 Case 1: cognitive disability support – the Memory Aiding Prompting System (MAPS)

Designing a tool for people with cognitive disabilities represents a '*universe of one*' problem (Fischer 2001). The 'universe of one' conceptualisation includes the empirical finding that (1) unexpected islands of abilities exist: users can have unexpected skills and abilities that can be leveraged to ensure a better possibility of task accomplishment; and (2) unexpected deficits of abilities exist (Cole 2006). Accessing and addressing these variations in skills and needs, particularly with respect to creating task support, requires an intimate knowledge of the user that only caregivers can provide (Cole 2006).

The Memory Aiding Prompting System (MAPS) aimed to provide a response to a 'universe of one' problem. Individuals with cognitive disabilities are often unable to live on their own because of deficiencies in memory, attention, and executive functionalities. These deficits can create an inability to consistently do normal domestic and work-related tasks such as cooking, using public transportation (see Case Study 2), taking medications (see Case Study 3), and simple employment tasks. Meta-design is particularly appropriate in creating systems in a 'universe of one' situation; it deals with situatedness to fit new needs at use time, to account for changing tasks and to embed computer artefacts in daily life and practices.

MAPS used a Personal Digital Assistant (PDA) platform to display verbal and pictorial prompts in a sequence that comprised a script related to a daily task (Figure 1).

A PC-based application provided tools for script creation, modification and sharing with other users via a web-based repository of scripts (Figure 2). The web repository had a browser-based search, storage, and retrieval engine facilitating sharing and building of a body of successful scripts. The script editor was an EUD environment that allowed an end-user developer (i.e.: caregiver or family member) to assemble self-recorded verbal prompts and personal photographs into filmstrip-like scripts, and to transfer them to a PDA that was used by the end users to perform tasks that, otherwise,

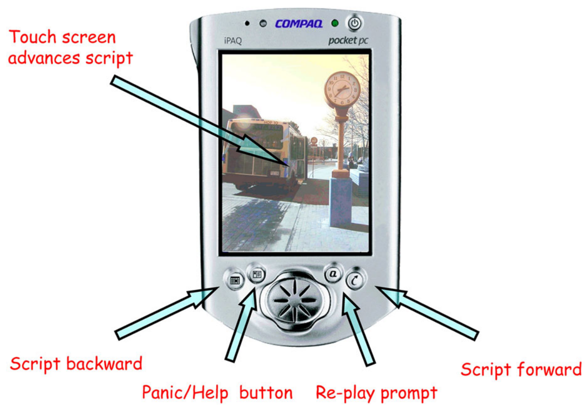


Figure 1. MAPS PDA platform.

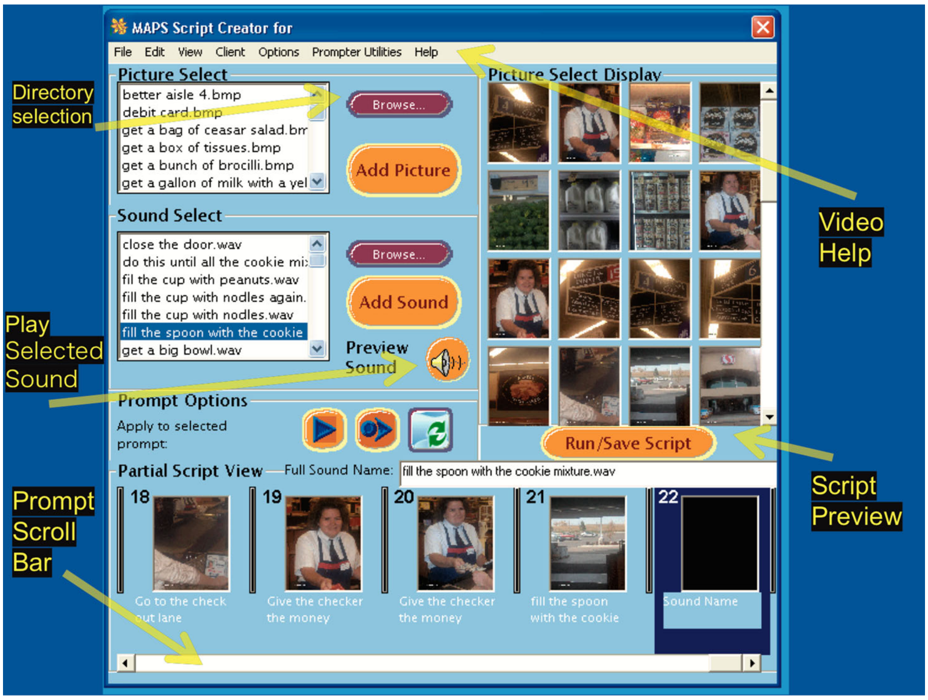


Figure 2. MAPS script editor.

would be hard or impossible to do. As a script was played, the events and context were logged, providing information for script refinement and analysis, as well as for possible immediate alternative prompts in case of breakdown situations.

MAPS end users were young adults with cognitive disabilities. The end-user developers needed to only be able to use email, take photographs, record audio, and use the multimedia editor/script installer. The script editor was designed to require very minimal computer skills. This need was a result of a trade-off between ease of use and effectivity in a 'universe of one' situation. Driven by this need, the effort in producing (design and coding) the MAPS editor was two or three times the work required to create the hand held prompter. This asymmetry is a typical split of effort in multi-tiered architectures. The end-user tool was basically a multimedia filmstrip player (modulo the experimental task forking option) whereas the script editor was a complex multimedia programming tool designed to be used by non-programmers. The tool designer required both end-user developers as well as other stakeholders (e.g.: special educators, clinicians, and occupational therapists) to participate in making the basic structure and operation of the MAPS system, especially the editor. Figure 3 illustrates the main tools and stakeholders involved in MAPS (Carmien and Fischer 2008).

The special needs of the end users dictated several design trade-offs. Since the prompts, consisting of images and verbal instructions, could not use standard icons or symbols, photos of the exact steps to be taken must be used to insure success (e.g.: a script instructing a 33 years old man how to fold his pants after cleaning required images of *his* pants in *his* room). Because of this, changing scripts sometimes required new images, and using a successful script as a basis for a template required stripping the image and replacing it with a description to be stored in the MAPS template database. Additionally, there were issues with co-morbidity, such as trembling digits or perseverating in use of touch pad that needed compensation as well as capturing and mitigating errors in following the multimedia scripts that guided the user in performing the task.

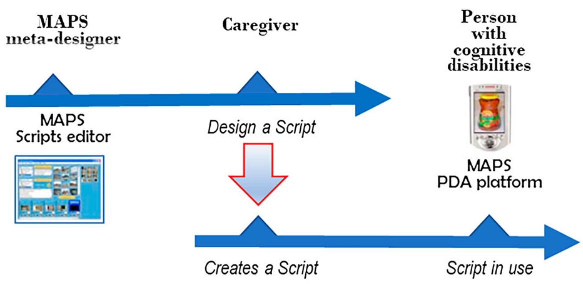


Figure 3. Meta-Design: empowering caregivers to act as designers.

The idea of MAPS was one of the most successful techniques for empowering the same population towards independence. It was based on existing work practices, namely occupational therapists' technique of teaching tasks using memorisation with repeated enactment. MAPS replaced memorising with a tool – instead of memorisation, the end user learned to use the PDA tool. Therefore, in the case of MAPS (Carmien and Fischer 2005), the trade-off is between

- *tools for living* grounded in a distributed cognition perspective (Hutchins 1996), in which intelligence is mediated by tools for achieving activities that would be error prone, challenging, or impossible to achieve and
- *tools for learning* grounded in a 'scaffolding with fading' perspective leading to autonomous performance by people without tools.

4.1.2 Case 2: empowering people to use public transportation – the ASSISTANT Project

The ASSISTANT project (ASSISTANT Project 2012) aimed to support seniors and persons with various disabilities (particularly diminished cognitive abilities) to use public transportation. The motivation underlining this three-year pan-European project derived from the observation that seniors might be new to the public transportation system (perhaps due to loss of ability to safely drive oneself due to age) and it is fundamental to support users when travelling goes awry.

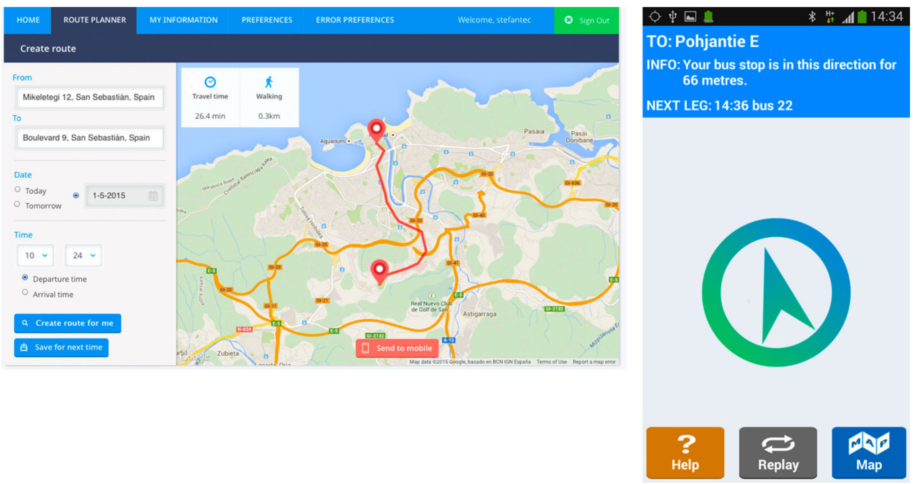
ASSISTANT consisted of three parts: (1) a web-based route and preference editor (Figure 4(a)); (2) a smartphone-based *Personal Navigation Device (PND)* (Figure 4(b)); and (3) a server coordinating the other two parts, processing real-time information for active routes being displayed on the PND, and implementing error detection and mitigation strategies.

A formal or informal caregiver was in charge of setting up the system for the end user in the route editor (the EUD environment of ASSISTANT), and creating routes in the editor, which were stored in the server and PND. At the scheduled time, the PND alerted the end user and guided him/her along the route while coordinating the monitoring of error states with the server to take appropriate actions according to the end-user profile (Carmien 2016).

ASSISTANT had several novel features: simplified route planning (in contrast to offering multiple confusing options), deep personalisation (both at the web browser accessibility and smartphone interface level), and error tracking and mitigation. Using real-time transit information, personalisation settings, and sensors on the smartphone, ASSISTANT monitored the progression of the trip following the planned route and providing dynamic mediation when the user got lost or confused, or the environment changed (e.g.: the bus was stalled on the road) (Siira and Heinonen 2015). This dynamic, real-time personalised error tracking and mitigation was based on an approach of 'designed for failure' (Carmien 2017), which takes into account that outages frequently occur and must be managed, ensuring that the passenger is always supported by the system.

These novel features are explained in detail in the following.

Personalisation features. The ASSISTANT system has two places to tailor the application: the PND preferences, which reflect typical ('shallow') personalisation (Figure



(a) (b)

Figure 4. (a) The web-based route and preference editor and (b) the Personal Navigation Device designed in the Assistant project.

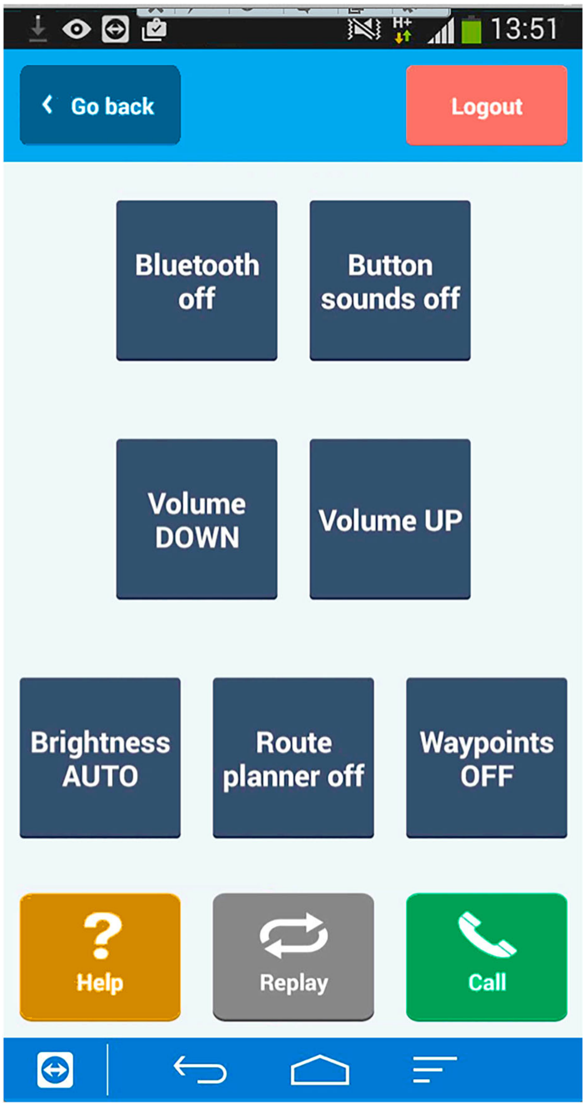


Figure 5. Tailoring assistant in the PND preferences.

5) and the route editor/setup tool, which supports also deep personalisation.

The three personalisation screens of the route editor/setup tool are shown in Figures 6–8 (corresponding to tabs 'My Information', 'Preferences' and 'Contacts' respectively).

In the initial personalisation screen (Figure 6), the left side is the typical personalisation input to create an account and set up the language for all screens. However, the right side does not just capture a primary caregiver for the person but creates a user with all the rights and privileges as the end user. A caregiver can make routes and examine exiting routes for the user, who nominates him/her as being in a caregiver role.

The second personalisation screen (Figure 7) is the core of the meta-design aspect of this system supporting deep personalisation. An initial glance at the number and description of the possible options might lead the novice user to become confused, so the system provides a low effort entry point through a set of user themes (in the upper left box). By selecting a theme, a set of checkboxes is activated, letting the user know how the system would be configured for them. This is a response to the problem of implementing *meta-design as a series of choices*, where some of them are quite specific and complex. By providing the user with templates, the cognitive difficulty of initial use can be significantly lowered, ensuring easy initial use but a high level of possibilities (*low floor high ceiling*).

Below the theme section, the user can select specific disabilities, in order to make the system configures itself to accommodate them. For instance, selecting 'Blind' turns on reading aloud all the guidance statements the PND displays en route. Selecting one of the mobility options changes the 'walking' rate for calculating the time needed for getting to transit stops, between stops en route, and from the last stop to the final goal.

The middle and right sections of 'Preferences' tab allow fine-tuning of the route editor to precisely fit the users' requirements. The user can select any combination of the three modalities of alerting him/her for giving next instructions about the route presented on the PND. Below them, there is a series of finer grained setting of PND display and behaviours. For instance, alerting the user with a beep and a vibration may not work for this user, so one might decide to continue the alert till the PND screen is touched. Similarly, some travellers might need to be told in an active way when their connectivity is lost. One problem many elders have with small screens and maps is that inadvertent interaction with the touchscreen may make the map unreadable, whereas other, more capable and experienced users, want the ability to move around in the map. Therefore, the default setting for 'Active Map Zoom' controls is, for all themes, not selected, so that the user has to really want this functionality (a typical opt-in functionality according to *nudge* theory (Thaler and Sunstein 2009)).

Beyond playing the alert till the screen is touched, the user can, in the right side of the screen, specify the number of repeat alerts to be played and the spacing between them. Some older and disabled users may have problems with using touch screens, due to palsy or a nervous perseveration tapping the screen, so the touch screen delay allows the system to only accept the first screen touch, disabling it for a user-settable time. The 'Walking distance' option collects the maximum distance that the route would make the user walk between two stops: if the distance is exceeded, then another route is presented. The application only chooses and presents the best route

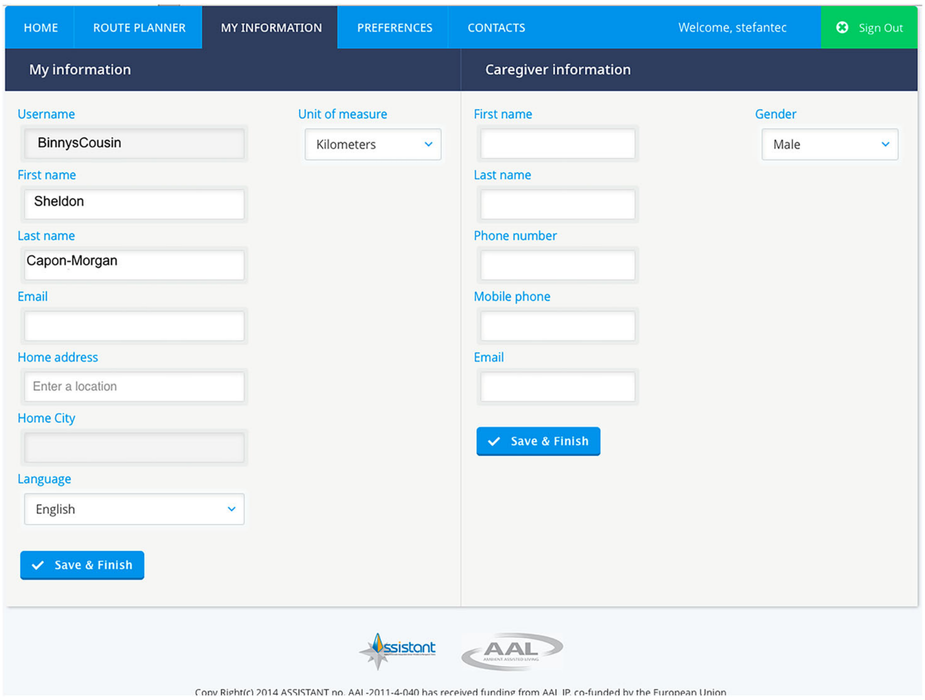


Figure 6. Personalisation through the 'My information' tab.

based on the source and destination entered in the route editor; the logic here is that, unlike other planning applications, it might be too complex for elders to compare and choose amongst multiple routes. The last option ('Time to exit home'), accommodates the wide range of 'getting ready to go' activities and times to perform these activities by adding the appropriate offset to the start of the route time.

The last screen (see Figure 8) allows the creation of a list of contacts for possible intervention, thus providing the support for both the user and caregiver to feel safe in using the system (see 'Error trapping and mitigation' below).

This detailed explanation of the personalisation features in ASSISTANT should provide some insights for understanding the boundaries between standard application design and meta-design. While 'shallow' personalisation makes using the application more comfortable, deep personalisation and functional options are able to cope with the problems and users considered in the project. A characteristic of meta-design is that the application, as used by *this* user, is completely specific to him/her: from the route planning, the way the PND presents the route instructions, and the fashion errors and breakdowns are mitigated for *this* user on this route (and possibly at this time of day), the personalised version of ASSISTANT is deeply different from ASSISTANT 'out of the box'.

Difficult activities in the meta-design were therefore finding either the most critical changes in behaviour that would make a big difference to the user or those additions to the behaviour that would circumvent adoption. A less difficult but critical issue was working out how to communicate these options to the users so that they could envision them in practice.

Error trapping and mitigation. The implementation of error trapping functionality in ASSISTANT aimed to cope with the practical problems of automatic adaptation in meta-design. Automatic adaptation of a system requires a sufficient log of user interactions and a measurable result to optimise. However, emulating this adaptivity in meta-design is problematic because of both of these requirements. In practice, the infrequency of use of the system may only produce a handful of data over months, and measuring the optimisable result is difficult in a complex system. This is magnified by the potentially high negative consequences in the process of use, both in the confusion matrix (false positives can lead to abandonment of the system, while false negatives can lead to harm) and in the very large error space of any act in the real world. This trade-off between capturing every problem and not raising too many false alarms is an important consideration in adoption or abandonment of high-functioning critical systems. Therefore, in designing ASSISTANT, adaptation was replaced by making the system iteratively (over many routes taken) adaptable, and

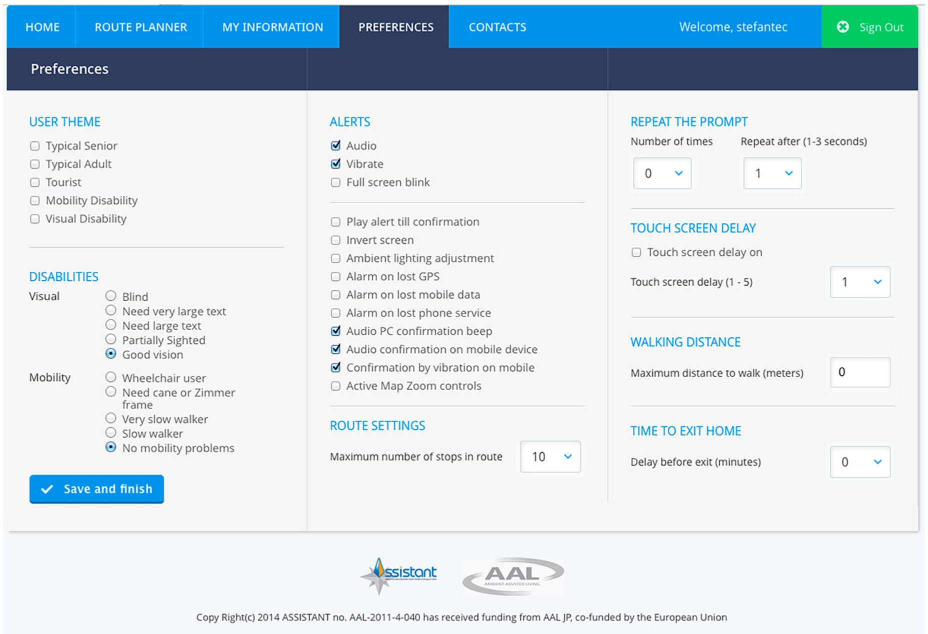


Figure 7. Personalisation through the 'Preferences' tab.

by making both error capture and mitigation part of the adaptation process.

ASSISTANT captured many simple errors, such as no battery power, missing connectivity (phone, network, GPS), and not being on the expected route. However, there were further error conditions than just not being on the route, which the system, doing constant monitoring, could detect easily. To these error conditions, the designers applied the 2-basket principle (Carmien 2016, 2017): enumerate the errors that one can anticipate and for all others get contextual information and go to a human to evaluate.

ASSISTANT's functional personalisation (see Figure 8, 'Contacts' tab) supported incremental intervention. Contacts added in the list could be relatives or caregivers or emergency personnel. When the system detected a serious error (e.g.: the smartphone was not responding to repeated pings, could not connect to the network, etc.), the server sent SMS messages, which identified itself and the user and contained the latest contextual information and route plans, to the contact at the top of the contact list. If an accepting SMS reply was not received, then the server moved up the list to the next entry in the contact list. Thus, by the user putting

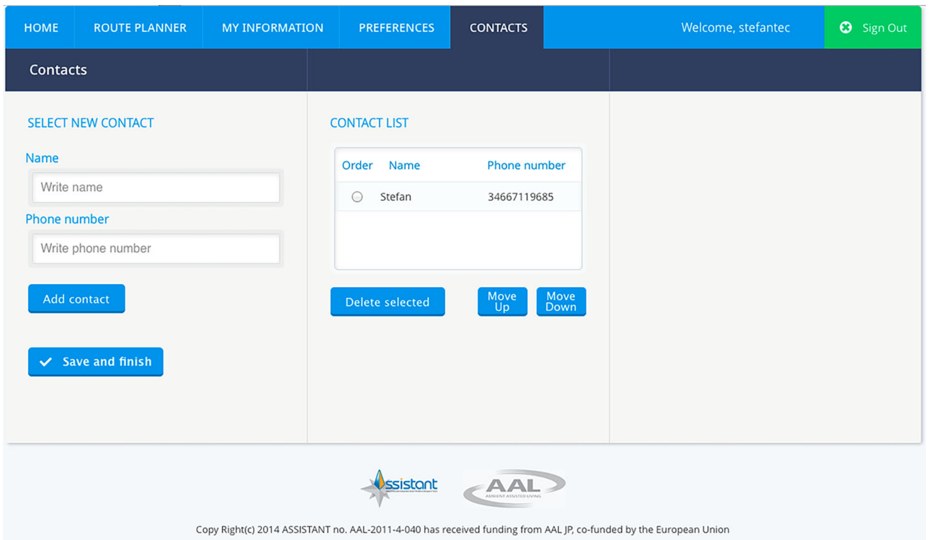


Figure 8. Personalisation through the 'Contacts' tab.

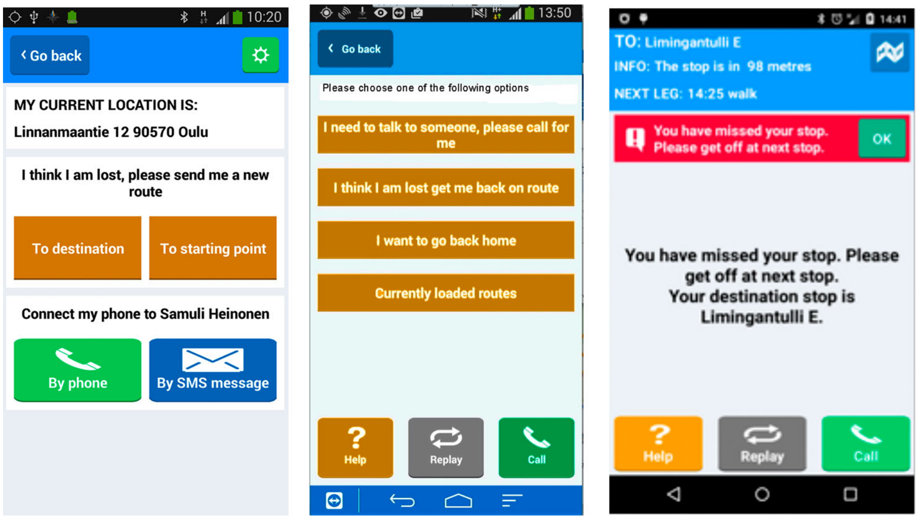


Figure 9. Request of intervention through the PND.

emergency personnel (such as the police) at the top of the contacts, the ASSISTANT traveller was guaranteed support no matter what situation occured. This same contact list was accessible by the user when he/she was informed by the PND of an error or when he/she felt there might be an error (Figure 9).

For each anticipated problem, an appropriate mitigation was applied. For instance, there were two versions of ASSISTANT with respect to its server: for some transit system, particularly the larger ones, there was an application programming interface (API) that allowed the server to read real-time location through GPS and current schedules; for other, typically smaller and more rural, regional transit administrations, which provided schedules and routes in GPRS data format, all route guidance was predicative rather than in real-time. In case of lost connectivity or offline server, the system switched to the last, most current schedule, and based its guidance on this, by informing the user so that he/she could apply his/her own intelligence and knowledge to the route. This *distributed cognition* approach (Rogers 1997) allowed the users to make their own decisions about what to do.

Finally, it is important to note that all the features described above resulted in a tightly fitted tool for a specific person using public navigation, a tool that, once used with satisfaction, did not require further effort beyond choosing a destination. Without the initial meta-design approach, this would have required endless fiddling and interpretation by the end user.

4.1.3 Case 3: Customized and interactive pill dispenser — the EUDroid Project

As observed in the ASSISTANT case, elderly people often need a specific support for carrying out daily activities, like taking public transportation or taking prescribed medicines. Several electronic devices that help old patients take pills have been proposed in the literature (Ahadani et al. 2012; Crema et al. 2015; Minaam and Abd-Elfattah 2018). However, most of the existing works do not allow users to customise the behaviour of the device. In the EUDroid project, we regarded this case as a multi-tiered proxy design problem in which the dyad was composed by an old patient and his/her caregiver who was called on to perform device personalisation.

The EUDroid system encompassed a modular pill dispenser customised by formal and informal caregivers according to the patient's specific therapy needs (Buono et al. 2018b). Using a smartphone, users could activate a buzzer and LEDs related to pills, according to the type of pill, the day and time of activation and some other properties. An underlying formal language was defined for device configuration and to provide users with the conditions to build complex rules for therapy reminders (Buono et al. 2018a). The end users were seniors that needed to follow a therapy by taking one or more pills at different times a day. The end-user developers were informal caregivers, usually relatives (often sons or daughters) or formal caregivers that were in charge of monitoring and checking that the therapy was correctly followed.

The overall system was composed by (see Figure 10):

- A pill dispenser with the actuators (LEDs, buzzer, button), which allowed the interaction with the patient;
- A web server that stored the information and checked if a specific event had occurred;
- An Android app that allowed the caregiver to specify the therapy and the pill dispenser behaviour and to get feedback about the ongoing therapy.

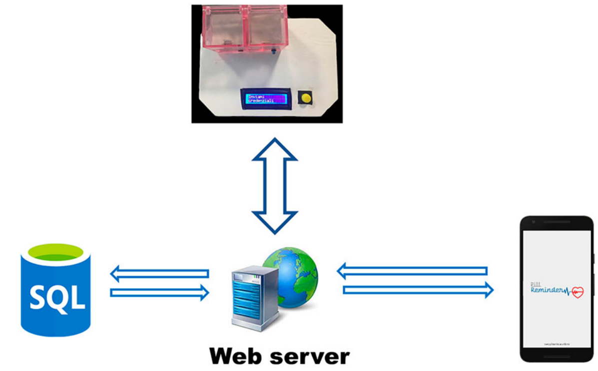


Figure 10. The architecture of the EUDroid prototype.

The pill dispenser was a modular physical device (see Figure 11). The modularity of the device allowed managing any number of pill types and multiple therapies at once, according to the specific needs. Physical modularity was ensured by a number of modular pill boxes (see Figure 12) that could be added to the pill dispenser according to the therapy. Once the therapy time occurred, a buzzer was activated to remind that one or more pills needed to be taken. At the bottom right of the pill dispenser, there was the button that the patient had to push to reset the device status when the pill was (or the pills were) taken. This button was the only user input to the pill dispenser and, when it was pressed, it sent to 'LOW' the status of the device, deactivating all the actuators.

The remote web server hosted a mySQL database that stored the ID of the pill reminder and the available commands for the device. The web server stored the information sent through the app, listened to occurrences of specific events to send a trigger to the pill reminder, and received (and processed) the button push event to notify when a pill had been taken. In case of more than one pill to be taken together, the user had first to take all the pills and then to press the button.

Figure 11. The pill dispenser prototype with modular pill boxes.

Figure 12. The modular pill box.

Different behaviours, represented as elementary event-condition-action rules, could be set by the caregiver through the Android mobile app (the EUD environment).

Figure 13 shows two screenshots of the mobile app. On the left a new therapy is being added, then a specific pill ('farmaco' in Italian) is chosen, and a set of event-condition-action rules are created (right) to define therapy scheduling.

Furthermore, the app provided the caregivers with a 'notification' feature in case a pill had not been taken at the right time; and it finally reported the history of pill taking (see Figure 14). In this way, the caregiver could be sure that the patient was taking the pills correctly.

One peculiarity of the pill dispenser was that the number of physical boxes was decided (and programmed) by

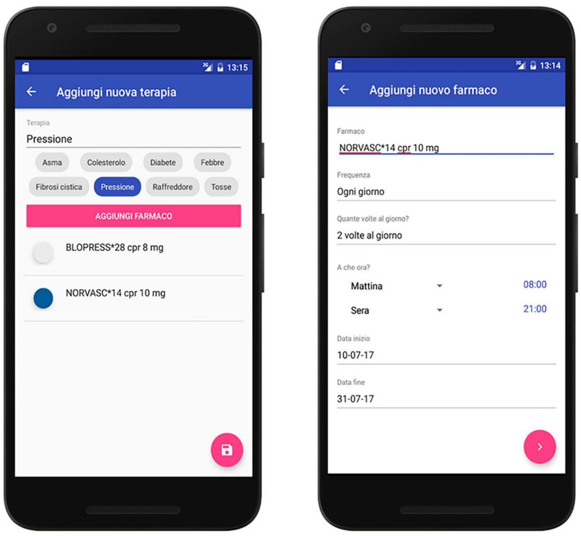


Figure 13. The Android mobile app: adding a new therapy (left), adding a new pill to be taken (right).

the caregiver and could be changed at any time to best fit the specific therapy.

The final goal of this project was to help elderly people take advantages of new technologies to improve their quality of life. However, the main sociological limitation of the proposal was related to the acceptability by the old people of a technological device that should replace paper. In one of our first user studies, we discovered that people felt better with a silent, always working, not proactive and discreet item, namely the schedule written on a paper sheet. On the other hand, this traditional approach needs the caregiver to be co-located in the same place where the patient lives.

4.2. Lessons learnt from case studies

While the digital divide (Schön, Sanyal, and Mitchell 1998) has often been discussed as the difference between having access to modern information technology or not, in the three case studies discussed above, IT has been exploited to design socio-technical environments that increase social inclusion and improve people's quality of life, going beyond the mere accessibility to IT.

This objective has been achieved through a meta-design approach supported by EUD environments that empower stakeholders to tailor and evolve digital artefacts for specific tasks and specific users. Meta-design supports the development of socio-technical systems that includes people at a small scale (in dyads), at a medium scale (by sharing artefacts within a specific community, e.g.: the community of caregivers), and at a large scale (by being broadly in society, e.g.: in a job or in daily life).

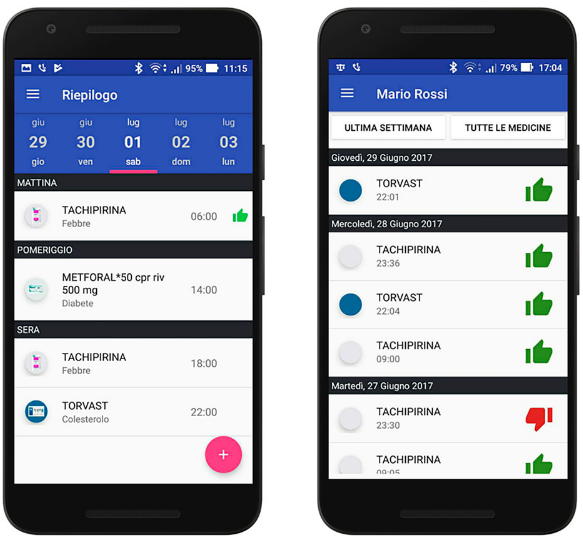


Figure 14. The Android mobile app: screenshot showing the today's therapy (left) and the overall status (right).

The adoption of assistive technology requires having an ecologically valid approach (for example, using ethnographic methods), in order to study the use of the developed systems. In other terms, integrating theory, system development, practice, and assessment plays a fundamental role. The assessment of a number of dyads to be studied in detail in their daily situations is necessary to cope with 'universe of one' problems, but this may bring high cost and effort both for developers and users.

The three case studies suggest that addressing the 'universe of one' world of people with specific needs requires highly personalised systems. For instance, ASSISTANT demonstrates the need for shallow and deep personalisation, which, on the other hand, may require a significant effort on behalf of caregivers. Also in the case of the pill dispenser, the high variability of therapies and the number of types of pills to be taken requires a high participation of caregivers in personalising the system and in checking if it is correctly working with the patient. Meta-design could support participation by making available all relevant possibilities, but a huge effort is needed to figure out all critical changes in behaviour and properly design related options in the EUD environment.

In the AT domain, error trapping and mitigation is crucial for avoiding potentially dangerous situations for specific users (who may get lost in a town or take an overdose of pills). This requires that all problems are anticipated, and related solutions designed, but the very large error space and the small usage datasets can make this activity rather difficult. On the other hand, perceiving the solution as not safe enough may prevent adoption or lead to abandon the technology. This issue is also related to the availability and reliability of real-time use data for automatic adaptation: performing inferences from these data can make a digital solution really powerful, but in case things go in an unexpected way (data are not available, connectivity is lost, etc.), a default minimal solution by either reducing functionality and/or usage solution space must always be foreseen in the meta-design phase.

Last but not least, sophisticated technological solutions may be rejected by end users, who can feel not comfortable with them and prefer traditional tools and practices. As in the EUDroid case, acceptability of technology is not only a matter of providing a more valid and secure daily practice, but psychological and social issues should be considered during meta-design.

5. Survey study

This section presents a survey study we performed with the help of research scholars in the field of EUD who have been involved or are currently involved in projects aimed at solving multi-tiered design problems. A questionnaire including 20 questions was used to carry out the survey study. A first version of the questionnaire was prepared by two authors of this paper, and, before submitting it to the participants in the study, it was validated with the help of the other two authors. The validation led to reformulate one question, to drop two questions that turned out to be unimportant and to substitute them with two new questions necessary to collect more interesting feedback. Time and effort required to fill in the questionnaire were considered adequate.

Nine questions of the final questionnaire were pure open questions, whilst the others encompassed both a closed choice and the request of explaining the given answer. Seven colleagues have been invited to participate in the survey study by email; all of them have accepted the invitation and filled in the submitted questionnaire. The participants live in four countries (Italy, Spain, The Netherlands and US), and are professors or researchers in seven different universities and research institutes.

An approach taken from qualitative analysis (Strauss and Corbin 1990), based on coding and iterative refinement of categories and themes, has been applied to process the answers provided in the questionnaire.

5.1 Issues explored with the survey study

The aim of the survey study was collecting additional information concerning the development of solutions for multi-tiered design problems through EUD and meta-design, with a particular attention to technical and social issues emerging during the specification, design and deployment phases of the related projects.

The submitted questionnaire was organised in three parts:

- The first part included an introduction to the definition of 'multi-tiered design problem' and to the terms 'end users' and 'end-user developers', in order to be sure that the participant understood the questions correctly. An example referring to the MAPS project was presented to give a concrete account of the terminology adopted in the questionnaire.
- The second part of the questionnaire referred to a specific project carried out by the participant, which description was available in literature (but that the participant could change with a different one) and included questions related to the involved stakeholders (especially, end users and end-user developers). The questions addressed issues related to stakeholders' needs, skills, roles, motivations, and perceived benefits and weaknesses of the proposed solution.
- The third part was aimed at better investigating the EUD solution proposed in the participant's project. It was asked its relationship with users' daily or work practice, its applicability to other application domains, its sociological and technological limitations, and its long-term sustainability. The last question asked for a comment on the adopted meta-design approach (if any), in order to gather the participant's opinion about the advantages brought to the project by such an approach.

5.2 Lessons learnt from the survey study

The projects described in the survey study referred to several domains including (1) autonomous assisted living (Chesta et al. 2018), (2) physical rehabilitation (Tetteroo 2017), (3) cognitive therapy (Garzotto et al. 2017), (4) education (Repenning et al. 2015), (5) smart cities (Valtolina and Di Gaetano 2018), (6) Internet of Things (Ardito et al. 2018), and (7) cultural heritage (Romano, Aedo, and Díaz 2016).

'Universe of one' cases. In the first three domains focused on elderly people, patients in physical rehabilitation, and children with cognitive disability, respectively, end users constitute a 'universe of one' (meaning that each user requires a personalised solution). Contrary to the wide-spread assumption that these end users' special needs would prevent the use of digital technology, *digital technology became the medium* to support these end users and foster their inclusion in daily practices (the same evidence resulted from the three case studies described in Section 4).

In these contexts, the role of end-user developers (caregivers, family members, and therapists creating software artefacts for the sake of someone else) is fundamental, because they are the only ones who know the characteristics and preferences of each end user and may design the most appropriate solution.

Among the benefits brought about by the technology, survey participants underlined how domain experts felt empowered in creating innovative solutions to be used in their work, even though some of them encountered difficulties in understanding all the potentialities. Usually, domain experts' motivation to acquire skills for using an EUD tool was just improving some other's quality of life or quality of care. The EUD tool was usually designed to be easily integrated with the existing daily/work practice, often by defining a domain-specific visual language and providing personalisation features of increasing complexity (from customising contents, to creating new contents until defining new functionalities). Participants declared that the system could be kept 'alive' in the hands of end-user developers even after researchers have left them alone. A formal and long training was never required for using the EUD tools, even though in a project experimented in a real setting it emerged that other kinds of effort were required; in such a case, the time needed for carrying out the EUD activity was considered too high in comparison to the perceived value, since it was often only a portion of the daily activity of the domain expert.

Another recurrent aspect of these projects is the importance of sharing the created artefacts, or parts of them, within the community of domain experts, both to speed up the development process of new artefacts and to foster inclusion of other domain experts (often, those experts who are less technologically oriented). On the other side, solutions designed by non-IT experts are perceived as fragile and unsafe; domain experts often feared to lose control on the system, by underestimating and misinterpreting its potentialities; finally, one participant underlined the side-effects (not yet explored) that digital technologies may have on end users with special needs.

'Universe of communities' cases. The other four domains (education, smart cities, Internet of Things, and cultural heritage) considered in the participants' projects allowed us to expand the analysis of applications and users that could be involved in multi-tiered design problems.

All these projects were oriented to a 'universe of communities', with different target communities of end users having homogenous needs and goals. Survey participants underlined different types of motivations, benefits and socio-technical limitations with respect to the 'universe of one' cases. Acquiring new competences and exploring new work possibilities, and thus a professional reason, was often a motivation for domain experts to participate in these projects. Economic reward was also perceived as an important benefit in the long run. Technological limitations were usually related to the prototypical nature of the developed tool, whilst different social limitations emerged, due to some domain experts that were not interested in learning and using a new tool or complained about their daily workload, or to political reasons for not adopting the proposed solution.

Patterns in both cases. All survey participants underlined the generality of their EUD solutions, that is, their applicability to a variety of contexts. The need of a multi-disciplinary team, often including roles beyond IT people, domain experts and end users, was another recurrent aspect of multi-tiered proxy design problems. For example, sometimes managers might work for or against the initiative, by prescribing the formal training of end-user developers or allowing/negating the time for carrying out the EUD activity. In the education case, the involvement of school managers and public institutions resulted to be fundamental for the successful deployment of the system.

Finally, as to the adopted meta-design approach, five survey participants underlined, as main advantages, the possibility for domain experts to express themselves through technology, the promotion of creativity tools, the definition of co-creation spaces, and the capability to foster communication and collaboration among different stakeholders, with different expertise and language. Two participants declared they did not adopt a meta-design approach; interestingly enough, these ones are the most convinced that long-term sustainability is an issue in their projects, since they considered unfeasible leaving the EUD solution solely in the hands of end-user developers; they also claimed that researchers and IT experts might be needed to manage the correct operation of the system, extend it overtime and intervene in case of breakdowns.

6. Design trade-offs and meta-design guidelines

The reflection on the case studies and the feedback collected through the survey study led us to identify a series of trade-offs that may be encountered while designing environments for social inclusion through multi-tiered architectures. From the analysis of these trade-offs, new additional guidelines for meta-design refining those proposed in (Fischer, Nakakoji, and Ye 2009; Fischer, Fogli, and Piccinno 2017) are defined.

6.1 Design trade-offs

Creating socio-technical environments for social inclusion addresses wicked problems (Rittel and Webber 1984) and it is therefore not an easy and straightforward task but requires the exploration of *design trade-offs* (Fischer 2018). The problem domains described in this paper are wicked problems for which there are (1) no perfect designs (Simon 1996); (2) no decontextualised sweet spots (Fischer 2018); and (3) no silver bullets (Brooks Jr 1987).

Without a deep understanding of both the strengths and weaknesses of the technology (e.g.: when, where, why, how, for what, and for whom it is and isn't suitable), researchers and developers will not be able to act in the best interests of stakeholders and may therefore (despite the best intentions) *increase the digital divide rather than the social inclusion* (Schön, Sanyal, and

Mitchell 1998). Some of the major design trade-offs that we have explored in our objectives to move towards more inclusive societies are discussed in the following.

Universe of one versus universe of communities. In our case studies and in some projects described by survey participants, each user needs a specific, personalised solution that requires forming a dyad with a domain expert who must be facilitated in tailoring the system for his/her end user. This usually requires that domain experts have intrinsic motivations to participate, due to their strong bond with end users and the desire to improve their quality of life. In other situations, such as education, cultural heritage, and smart cities, the objective is instead designing environments to allow the creation of artefacts that may be used by different communities of users. This obviously guarantees general applicability and scalability, with significant economic rewards, even though the risk of social exclusion could be higher than in solutions for 'universe of one' problems.

Tools for living versus tools for learning. Projects proposing novel IT solutions are usually focused on developing tools for living, namely tools that facilitate the execution of activities in daily life and/or work. However, promoting social inclusion should consider the possibility of developing tools for learning (Carmien and Fischer 2005). Learning may include acquiring skills in doing something or knowledge and competencies about a specific domain, thus making people evolve and become more independent. However, even though this deeper understanding of the activity may create the possibility of an independence of the tool, this will often come at considerable costs, including time and effort in learning the activity, and then ultimately executing it in a possibly error-prone and time-consuming way. Finally, characterising a tool as a tool for living or as a tool for learning is not an attribute of the tool itself but is determined in many cases by the user's objectives and the use context (Fischer 2006). Therefore, a meta-design approach must contribute to frame the problem by taking into account these different possibilities.

Overreliance on external tools versus independence. Cognitive biases, due to our limited resources, affect our everyday interaction with real word, including IT technology. As a consequence, each of us could become dependent on external tools (e.g.: today we do not remember any more our relatives' phone numbers because they are in our cell phone). This brings about the risk of losing our capabilities to properly deal with unexpected situations or, worst, of being persuaded to do things that could be against our safety, interests and values. Designing socio-technical systems for social inclusion should pay attention to the weaknesses of people (especially the most fragile, like children, elderly and cognitive disabled), and foster distributed cognition to avoid overreliance on external tools.

Easy-to-use versus difficult-to-design. Designing IT solutions that are easy-to-use by everyone, independently of ability, age, culture, and so on, is a challenge that meta-design aims to address. However, this often leads to create EUD environments that require advanced skills for designing the right solution for the different users. 'Ease-of-use' along with the 'burden of learning something' are often used as arguments for why people will not engage in design. Building systems that support users to act as designers and not just as consumers is often less successful than the meta-designers have hoped for. Being able to cope with the (sometimes opposed) requirements of end users and end-user developers is an issue to be considered in meta-design.

Personalisation versus participation overload. IT solutions often require to be deeply personalised according to different users' needs, and personalisation could be less or more sophisticated. This may lead to a participation overload for end-user developers, who are called on to perform such personalisation for the benefit of other people. To address this trade-off, existing methods such as reuse, redesign, and remixing need to be further improved and extended. EUD environments should be conceived as construction kits and domain-oriented design environments providing high-level building blocks and allowing users to express themselves in their own language and notation (Fischer, Fogli, and Piccinno 2017).

Personalisation versus scalability. Personalisation brings with it also the problem of managing all the created extensions in the long run. While end-user developers might be called on to manage different extensions for different end users, developers must be able to acquire control on all the extensions and possibly integrate them for the sake of developing an improved and more general version of the EUD environment, in order to foster scalability. The answer to this challenging situation may be in the development of social structures around these systems such as *collaborative work practices* (Nardi 1993).

Deployment in real settings versus fragility of the system. Deployment of IT solutions in real settings often results in unexpected situations and/or emergent user behaviours that may influence users' rejection or acceptance of the solutions themselves. Openness fostered by meta-design may go in conflict with the need of 'designing for failure' (Carmien 2017), and thus avoiding to cause harm to people unable to promptly react to unknown situations. Especially in the multi-tiered proxy design problems discussed in this paper, responsibility of designing a specific personalisation or extension is assigned to the end-user developer, someone who does not have the necessary competence and knowledge to foresee all failure causes and cope with them appropriately. Therefore, this trade-off underlines how meta-design must define all the socio-technical conditions necessary to address the errors occurring at run time, even those that cannot be anticipated and for which human intervention is needed.

6.2 Meta-design guidelines

General guidelines for meta-design were originally proposed in (Fischer, Nakakoji, and Ye 2009), and slightly revised in (Fischer, Fogli, and Piccinno 2017). Based on the research activities documented in this paper, we will propose a new and updated list specifically oriented to the design of socio-technical systems for social inclusion:

(1) *Adopt an ecologically valid approach*. Designing socio-technical systems addressing multi-tiered design problems require an in-depth analysis of daily/work practice to obtain a digital artefact that could be easily integrated with such practice. In 'universe of one' contexts, this means the assessment of a number of dyads to understand needs, preferences, idiosyncrasies, values, strengths and weaknesses of the involved stakeholders.

(2) *Define spaces for co-creation*. Beyond the end user, end-user developer and software developer, several other roles are often involved in wicked problems. Meta-design should foster multi-disciplinarity by creating tools and conditions for communication and collaboration both at design time and use time. Meta-design should provide spaces for expressing one's own creativity and promoting inclusion at different scales: small scale (the dyad), medium scale (the community), and large scale (the society).

(3) *Support design rather than programming*. To address multi-tiered design problems, meta-design should support high-level EUD activities allowing domain experts to easily design and personalise IT solutions for other people, rather than requiring them to learn some (possibly easy-to-use) programming environment or language. A 3-layer approach is proposed in literature to address this problem: (1) the meta-design layer, where there are tools and practices to design the EUD environment; (2) the design layer, where there is the EUD environment devoted to domain experts; and (3) the use layer, where the digital artefact created by the domain expert is used by end users (Ardito et al. 2018; Costabile et al. 2007).

(4) *Foster sharing and collaboration*. In most of the analysed cases and projects, artefact sharing is required to increase re-use of solutions and improve domain experts' efficiency. Thus, creating technical mechanisms and social enablers for collaboration might make domain experts' work easier and contribute to stimulate participation and promote inclusion of those domain experts that are less technologically oriented or not interested in technology at all.

(5) *Accommodate different stakeholders' motivation*. Domain experts are triggered by different motivations to participate in solving wicked problems. In the 'universe of one' case, they are usually moved by intrinsic motivations, such as the possibility of improving some others', and sometimes their own, quality of life or work. If a 'universe of communities' is the target of the designed technology, motivations may be more related to professional career or economic rewards, thus in general to extrinsic aspects. A meta-design project should consider these issues from the beginning to create the socio-technical conditions that are most suitable to the different stakeholders' motivations.

(6) *Work for scalability and sustainability*. To promote social inclusion, IT solutions may be designed to address as many end users as possible. Furthermore, it is important to consider the different abilities and experiences of domain experts who may be willing to participate in shallow or deep personalisation, or in more sophisticated function extension. This should be accompanied by proper mechanisms to support long-term sustainability of projects, even after the software developers have left them.

(7) *Favour a distributed cognition approach*. Ensuring usability and accessibility of the artefacts created by end-user developers is a fundamental objective of multi-tiered proxy design solutions. However, as observed in the ASSISTANT project, error trapping and mitigation in real settings is often difficult due to the width of the error space. Safety-critical situations may occur, which require human intervention. This suggests studying solutions that, beyond pursuing usability and accessibility, adopt a distributed cognition approach to avoid overreliance on technology.

7. Limitations of the research and future work

The findings reported in this paper are mainly derived from the three case studies of the authors, and thus might have a limited scope and be affected by biases. We tried to alleviate this problem by carrying out a survey involving seven research scholars of other universities, in order to integrate further perspectives.

Only the 'universe of one' problem has been really deepened in this paper, whilst 'universe of communities' cases have been commented only with reference to others' research activities. To address this limitation, a more systematic study that examines several design solutions should be carried out, as well as performing a larger survey with a higher number of research scholars.

Despite the fact that some EUD environments and their supporting research have been around for years (Lieberman, Paternò, and Wulf 2006), and some success models exist (as discussed in the paper), the impact of academic research efforts in this area has been limited. Substantially more experience and assessment is required to determine whether the *advantages* of meta-design (putting owner of problems in charge, empowering end-users to act as designers and not only as consumers, personalising systems to the needs of specific users, evolving systems to correspond to a changing world) will outweigh the *disadvantages* (participation overload in personally irrelevant activities, lack of relevant technical skills, propagation of incoherent voices, and incompatible versions of systems). Such a determination will depend on creating a deeper understanding of the associated design trade-offs for achieving social inclusion in multi-tiered design problems.

Exploring design trade-offs is a broad and important topic to explore for the future of our societies in the information age. The wicked problems facing us have no correct solutions or right answers; the rightness or wrongness of a design is not a question of fact (as it is the case in the natural sciences), but a question of value and interest of the involved stakeholders requiring careful choices between trade-offs. A more elaborate framework for design trade-offs (transcending the exploration of the design trade-offs identified and analysed in this paper) is required to achieve a deeper understanding about how design trade-offs can contribute (1) to avoid oversimplified solutions ignoring important facets of complex problems; (2) to uncover unknown alternatives and identify the truly limiting factors that underlie problems; (3) to transcend one-sided views and group think; and (4) to move beyond binary choices by identifying interesting syntheses and meaningful compromises between specific design trade-offs.

Achieving the sustainability of IT-based solutions in the context of a research project is a challenge, irrespective of the efforts made to this goal, as most of the effects end when a project is over. This is a timely and pressing topic, as research in applied computing today requires IT researchers to deeply engage with practitioners in order to achieve innovative results that are also useful in practice. Hence, critical to successful IT research is the ability to develop solutions that can be adapted and further developed by users with a view onto changing their practices.

8. Conclusions

This paper explored the design trade-offs that emerge when addressing multi-tiered proxy design problems for social inclusion. These are wicked problems that can be encountered in several domains, especially those ones involving people with special needs. Despite the great need and the unique opportunities to improve the quality of life for people with special needs, few research activities have been focused to solve them.

Our research efforts and projects have been grounded in exploring meta-design as a promising framework to fill this gap. By analysing related research activities, and reflecting on our own case studies, this paper identified design trade-offs and derived design guidelines for meta-design specifically oriented to support the creation of socio-technical environments for social inclusion addressing multi-tiered proxy design problems.

In particular, we observed how modern technologies (e.g.: the pervasiveness of mobile devices, smart objects, and Internet of Things) have the potential to support and enrich people's life in multiple contexts. We identified unique opportunities to improve the social inclusion not only of people with special needs but of all of us. Even though people with cognitive disabilities might seem to be a special case for HCI research and practice, our research is grounded in the assumption that many important issues can be learnt by focusing on these communities. All human beings have limited cognitive abilities (if we had perfect memories, there would be no need to write things down), and advances in human cognition and intelligence are made possible by powerful socio-technical environments.

Acknowledgments

The authors would like to thank the scholars who participated in the survey study for their insightful comments and suggestions.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Abras, Chadia, Diane Maloney-Krichmar, and Jennifer Preece. 2004. "User-centered Design." In *Berkshire Encyclopedia of Human-Computer Interaction*, edited by William Sims Bainbridge, 763–769. Great Barrington, MA: Berkshire Publishing Group.

Ahadani, Mohammad Auzi, Liyanage C. De Silva, Iskandar Petra, Mohammad Faisal A. Hameed, and Teck Sion Wong. 2012. "Low Cost Robotic Medicine Dispenser." *Procedia Engineering* 41: 202–209. doi:10.1016/j.proeng.2012.07.163.

Ardito, Carmelo, Paolo Buono, Giuseppe Desolda, and Maristella Matera. 2018. "From Smart Objects to Smart Experiences: An end-User Development Approach." *International Journal of Human-Computer Studies* 114: 51–68. doi:10.1016/j.ijhcs.2017.12.002.

Ardito, Carmelo, Maria Francesca Costabile, Giuseppe Desolda, Maristella Matera, Antonio Piccinno, and Matteo Picozzi. 2012. "Composition of Situational Interactive Spaces by End Users: A Case for Cultural Heritage." In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense through Design (NordiCHI)*, 79–88. New York, NY: ACM. doi: 10.1145/2399016.2399029.

ASSISTANT Project. 2012. "Description of Work (DOW)." In *AAL Program Funding.*

Barakova, Emila I., Jan C. C. Gillesen, Bibi E. B. M. Huskens, and Tino Lourens. 2013. "End-user Programming Architecture Facilitates the Uptake of Robots in Social Therapies." *Robotics and Autonomous Systems* 61 (7): 704–713. doi:10.1016/j.robot.2012.08.001.

Barricelli, Barbara Rita, Fabio Cassano, Daniela Fogli, and Antonio Piccinno. 2019. "End-user Development, end-User Programming and end-User Software Engineering: A Systematic Mapping Study." *Journal of Systems and Software* 149: 101–137. doi:10.1016/j.jss.2018.11.041.

Binder, Thomas, Giorgio De Michelis, Pelle Ehn, Gulio Jacucci, Per Linde, and Ina Wagner. 2011. *Design Things*. Cambridge, MA: MIT Press.

Bødker, Susanne, and Kaj Grønbæk. 1991. "Cooperative Prototyping: Users and Designers in Mutual Activity." *International Journal of Man-Machine Studies* 34 (3):453–478. doi:10.1016/0020-7373(91)90030-B.

Brooks Jr, Frederick P. 1987. "No Silver Bullet Essence and Accidents of Software Engineering." *Computer* 20 (4): 10–19. doi:10.1109/MC.1987.1663532.

Buchina, Nina, Sherin Kamel, and Emilia Barakova. 2016. "Design and Evaluation of an End-user Friendly Tool for Robot Programming." In *Proceedings of the 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2016)*, 185–191. New York, NY: IEEE. doi: 10.1109/ROMAN.2016.7745109

Buono, Paolo, Fabio Cassano, Alessandra Legretto, and Antonio Piccinno. 2018a. "EUDroid: a Formal Language Specifying the Behaviour of IoT Devices." *IET Software* 12 (5): 425–429. doi:10.1049/iet-sen.2017.0347.

Buono, Paolo, Fabio Cassano, Alessandra Legretto, and Antonio Piccinno. 2018b. "A Homemade Pill Dispenser Prototype Supporting Elderly." In *Current Trends in Web Engineering (ICWE)*, edited by Irene Garrigós, and Manuel Wimmer, 120–124. Cham: Springer International Publishing.

Cabitza, Federico, and Carla Simone. 2015. "Building Socially Embedded Technologies: Implications About Design." In *Designing Socially Embedded Technologies in the Real-World*, edited by V. Wulf, K. Schmidt, and D. Randall, 217–270. London: Springer.

Carmien, Stefan. 2016. "Assistive Technology Design for Intelligence Augmentation." In *Synthesis Lectures on Assistive, Rehabilitative, and Health-Preserving Technologies*, edited by Ron Baecker, i–171. San Rafael, CA: Morgan and Claypool Publishers.

Carmien, Stefan. 2017. "Assume the Worst: Design for Failure." Paper presented at the Workshop Designing for Uncertainty in HCI: When Does Uncertainty Help?, Thirty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2017), Denver, CO, May 7, 2017.

Carmien, Stefan, Melissa Dawe, Gerhard Fischer, Andrew Gorman, Anja Kintsch, and James F. Sullivan Jr. 2005. "Socio-technical Environments Supporting People with Cognitive Disabilities Using Public Transportation." *ACM Transactions on Computer-Human Interaction* 12 (2): 233–262. doi:10.1145/1067860.1067865.

Carmien, Stefan, and Gerhard Fischer. 2005. "Tools for Living and Tools for Learning." In *Proceedings of the HCI International Conference (HCII)*, July 2005. Las Vegas. (published on a CD).

Carmien, Stefan, and Gerhard Fischer. 2008. "Design, Adoption, and Assessment of a Socio-Technical Environment Supporting Independence for Persons with Cognitive Disabilities." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 08)*, 597–606. New York, NY: ACM. doi: 10.1145/1357054.1357151.

Carmien, Stefan, and Gerhard Fischer. 2010. "Beyond Human-Computer Interaction: Meta-Design in Support of Human Problem-Domain Interaction." In *Generalization of Knowledge: Multidisciplinary Perspectives*, edited by M. Banich, and D. Caccamise, 331–349. New York: Psychology Press.

Celentano, Augusto, and Marek Maurizio. 2011. "An End-User Oriented Building Pattern for Interactive Art Guides." In *End-User Development*, edited by Maria Francesca Costabile, Yvonne Dittrich, Gerhard Fischer, and Antonio Piccinno, 187–202. Berlin Heidelberg: Springer.

Chesta, Cristina, Luca Corcella, Marco Manca, Fabio Paternò, and Carmen Santoro. 2018. "Trigger-action Programming for Context-aware Elderly Support in Practice." In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI 2018)*, 58:1–58:3. New York, NY: ACM. doi: 10.1145/3206505.3206582.

Cole, Elliot. 2006. "Patient-Centered Design as a Research Strategy for Cognitive Prosthetics: Lessons Learned from Working with Patients and Clinicians for 2 Decades." In *Proceedings of the CHI 2006 Workshop on Designing Technology for People with Cognitive Impairments*, Montreal, Canada.

Costabile, Maria Francesca, Daniela Fogli, Rosa Lanzilotti, Piero Mussio, and Antonio Piccinno. 2006. "Supporting Work Practice through End-User Development

Environments." *Journal of Organizational and End User Computing* 18 (4): 43–65. doi:10.4018/joeuc.2006100103.

Costabile, Maria Francesca, Daniela Fogli, Piero Mussio, and Antonio Piccinno. 2007. "Visual Interactive Systems for End-User Development: a Model-Based Design Methodology." *IEEE Transactions on System Man and Cybernetics Part A-Systems and Humans* 37 (6): 1029–1046. doi:10.1109/TSMCA.2007.904776.

Crema, Claudio, Alessandro Depari, Alessandra Flammini, Mirko Lavarini, Emiliano Sisinni, and Angelo Vezzoli. 2015. "A Smartphone-Enhanced Pill-dispenser Providing Patient Identification and In-take Recognition." In *Proceedings IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. New York: IEEE. doi: 10.1109/MeMeA.2015.7145252.

Davis, Fred D. 1989. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology." *MIS Quarterly* 13 (3): 319–340. doi:10.2307/249008.

Fischer, G. 2001. "User Modeling in Human-Computer Interaction." *User Modeling and User-Adapted Interaction* 11 (1): 65–86. doi:10.1023/A:1011145532042.

Fischer, Gerhard. 2006. "Distributed Intelligence: Extending the Power of the Unaided, Individual Human Mind." In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 06)*, 7–14. New York, NY: ACM. doi: 10.1145/1133265.1133268.

Fischer, Gerhard. 2018. "Design Trade-Offs for Quality of Life." *ACM Interactions* 25 (1): 26–33. doi:10.1145/3170706.

Fischer, Gerhard, Daniela Fogli, and Antonio Piccinno. 2017. "Revisiting and Broadening the Meta-Design Framework for End-User Development." In *New Perspectives in End-User Development*, edited by Fabio Paternò, and Volker Wulf, 61–97. Cham: Springer International Publishing.

Fischer, Gerhard, and Elisa Giaccardi. 2006. "Meta-Design: A Framework for the Future of End User Development." In *End User Development*, edited by H. Lieberman, F. Paternò, and V. Wulf, 427–457. Dordrecht: Springer.

Fischer, Gerhard, Kumiyo Nakakoji, and Yunwen Ye. 2009. "Metadesign: Guidelines for Supporting Domain Experts in Software Development." *IEEE Software* 26 (5): 37–44. doi:10.1109/MS.2009.134.

Fogli, Daniela, and Antonio Piccinno. 2013. "Co-evolution of End-User Developers and Systems in Multi-tiered Proxy Design Problems." In *End-User Development*, edited by Yvonne Dittrich, Margaret Burnett, Anders Mørch, and David Redmiles, 153–168. Berlin Heidelberg: Springer.

Fogli, Daniela, Davide Sansoni, Emanuele Trivella, Alberto Arenghi, and Ivana Passamani. 2018. "Advanced Interaction Technologies for Accessible and Engaging Cultural Heritage." In *Smart Objects and Technologies for Social Good (GoodTechs 2017)*, edited by B. Guidi, L. Ricci, C. Calafate, O. Gaggi, and J. Marquez-Barja, 364–373. Cham: Springer.

Gantt, Michelle, and Bonnie A. Nardi. 1992. "Gardeners and Gurus: Patterns of Cooperation among CAD Users." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*, 107–117. New York, NY: ACM. doi: 10.1145/142750.142767.

Garzotto, Franca, Mirko Gelsomini, Vito Matarazzo, Nicolò Messina, and Daniele Occhiuto. 2017. "XOOM: An End-User Development Tool for Web-Based Wearable Immersive Virtual Tours." In *Web Engineering (ICWE 2017)*, edited by J. Cabot, R. De Virgilio, and R. Torlone, 507–519. Cham: Springer.

Hermans, Felienne, Martin Pinzger, and Arie van Deursen. 2011. "Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams." In *Proceedings of the 33rd International Conference on Software Engineering (ICSE '11)*, 451–460. New York, NY: ACM. doi: 10.1145/1985793.1985855.

Hutchins, Edwin. 1996. *Cognition in the Wild*. Cambridge, MA: MIT Press.

Lieberman, Henry, Fabio Paternò, and Volker Wulf. 2006. *End User Development*. Dordrecht: Springer.

Marchiori, Eugenio J., Javier Torrente, Ángel del Blanco, Pablo Moreno-Ger, Pilar Sancho, and Baltasar Fernández-Manjón. 2012. "A Narrative Metaphor to Facilitate Educational Game Authoring." *Computers & Education* 58 (1): 590–599. doi:10.1016/j.compedu.2011.09.017.

Minaam, Abdul Diaa Salama, and Mohamed Abd-Elfattah. 2018. "Smart Drugs: Improving Healthcare Using Smart Pill Box for Medicine Reminder and Monitoring System." *Future Computing and Informatics Journal* 3 (2): 443–456. doi:10.1016/j.fcij.2018.11.008.

Nardi, Bonnie. 1993. *A Small Matter of Programming: Perspectives on End User Computing*. Cambridge, MA: The MIT Press.

Norman, Donald A., and Stephen W. Draper. 1986. *User Centered System Design; New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: L. Erlbaum Associates Inc.

Paterno, Fabio, and Volker Wulf. 2017. *New Perspectives in End-User Development*. Cham: Springer International Publishing.

Repenning, Alexander, and James Ambach. 1997. "The Agentsheets Behavior Exchange: Supporting Social Behavior Processing." In *Proceedings of CHI 97, Extended Abstracts on Human Factors in Computing Systems (CHI EA '97)*, 26–27. New York, NY: ACM. doi: 10.1145/1120212.1120231.

Repenning, Alexander, David C. Webb, Kyu Han Koh, Hilarie Nickerson, Susan B. Miller, Catharine Brand, Ian Her Many Horses, et al. 2015. "Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation." *ACM Transactions on Computing Education (TOCE)* 15 (2):1–31. doi:10.1145/2700517.

RESNA. 2019. Rehabilitation Engineering and Assistive Technology Society of North America. "Resources and Definitions." Accessed 14 April 2019. https://www.resna.org/resources-definitions.

Rittel, Horst W. J., and Melvin M. Webber. 1984. "Planning Problems are Wicked Problems." In *Developments in Design Methodology*, edited by N. Cross, 135–144. New York: John Wiley & Sons.

Rogers, Yvonne. 1997. "A Brief Introduction to Distributed Cognition." Accessed 14 April 2019. http://yvonnerogers.com/wp-content/uploads/2014/07/dcog-brief-intro.pdf

Romano, Marco, Ignacio Aedo, and Paloma Díaz. 2016. "Designing an End-User Augmented Reality Editor for Cultural Practitioners." In *Ubiquitous Computing and Ambient Intelligence (UCAmI 2016)*, edited by C. García, P. Caballero-Gil, M. Burmester, and A. Quesada-Arencibia, 365–371. Cham: Springer.

Rosson, Mary Beth, Hansa Sinha, Mithu Bhattacharya, and Dejin Zhao. 2007. "Design Planning in End-User Web

Development." In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, 189–196. New York: IEEE. doi: 10.1109/VLHCC.2007.45.
Scaffidi, Christopher. 2017. "Potential Financial Payoffs to End-User Developers." In *End-User Development (IS-EUD 2017)*, edited by S. Barbosa, P. Markopoulos, F. Paternò, S. Stumpf. S. Valtolina, 183-190. Cham: Springer.
Schön, Donald Alan. 1983. *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.
Schön, Donald A., Bishwapriya Sanyal, and William J. Mitchell. 1998. *High Technology and Low-Income Communities: Prospects for the Positive Use of Advanced Information Technology*. Cambridge, MA: MIT Press.
Schuler, Douglas, and Aki Namioka. 1993. *Participatory Design: Principles and Practices*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
Siira, Erkki, and Samuli Heinonen. 2015. "Enabling Mobility for the Elderly: Design and Implementation of Assistant Navigation Service." In *Proceedings of the 14th International Conference on Mobility and Transport for Elderly and Disabled Persons (TRANSED 2015)*. Lisbon, July 2015.
Simon, Herbert A. 1996. *The Sciences of the Artificial* (3rd ed.). Cambridge, MA: MIT Press.
Strauss, Anselm, and Juliet M. Corbin. 1990. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Thousand Oaks, CA: Sage Publications, Inc.
Tetteroo, Daniel. 2017. "Tagtrainer: end-user adaptable technology for physical rehabilitation." In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth '17)*, 452–454. New York, NY: ACM. doi: 10.1145/3154862.3154901.
Tetteroo, Daniel, Henk Seelen, Annick Timmermans, and Panos Markopoulos. 2014. "Rehabilitation Therapists as Software Creators? Introducing End-User Development in a Healthcare Setting." *International Journal of Sociotechnology and Knowledge Development (IJSKD)* 6 (1): 36–50. doi:10.4018/ijskd.2014010103.
Tetteroo, Daniel, Paul Vreugdenhil, Ivor Grisel, Marc Michielsen, Els Kuppens, Diana Vanmulken, and Panos Markopoulos. 2015. "Lessons Learnt from Deploying an End-User Development Platform for Physical Rehabilitation." *In Proceedings of the* 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15), 4133–4142. New York, NY: ACM. doi: 10.1145/2702123.2702504.
Thaler, Richard H., and Cass R. Sunstein. 2009. *Nudge – Improving Decisions about Health, Wealth, an Happiness*. London: Penguin Books.
Valtolina, Stefano, and Serena Di Gaetano. 2018. "ICT-based Methodology for Fostering ADHD Students Inclusion in Classrooms." *International Journal of Sociotechnology and Knowledge Development (IJSKD)* 10 (1): 1–20. doi:10.4018/IJSKD.2018010101.
Won, Markus, Oliver Stiemerling, and Volker Wulf. 2006. "Component-Based Approaches to Tailorable Systems." In *End User Development*, edited by Henry Lieberman, Fabio Paternò, and Volker Wulf, 115–141. Dordrecht: Springer.