ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/335445806

A User-Centred Methodology for the Development of Computer-Based Assistive Technologies for Individuals with Autism

Chapter · August 2019

DOI: 10.1007/978-3-030-29381-9_6

CITATION
1

READS
143

4 authors:

Raquel Hervás
Complutense University of Madrid
73 PUBLICATIONS **534** CITATIONS
SEE PROFILE

Virginia Francisco
Complutense University of Madrid
39 PUBLICATIONS **271** CITATIONS
SEE PROFILE

Gonzalo Méndez
Complutense University of Madrid
73 PUBLICATIONS **289** CITATIONS
SEE PROFILE

Susana Bautista
Complutense University of Madrid
25 PUBLICATIONS **104** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Idylico View project

The GUIDE project View project

All content following this page was uploaded by Gonzalo Méndez on 11 September 2019.

The user has requested enhancement of the downloaded file.

A User-Centred Methodology for the Development of Computer-Based Assistive Technologies for Individuals with Autism

Raquel Hervás[1], Virginia Francisco[1], Gonzalo Méndez[2], and Susana Bautista[3]

[1] Facultad de Informática, Universidad Complutense de Madrid, Spain
{raquelhb,virginia}@fdi.ucm.es
[2] Instituto de Tecnología del Conocimiento (UCM), Madrid, Spain
gmendez@fdi.ucm.es
[3] Escuela Politécnica, Universidad Francisco de Vitoria, Pozuelo de Alarcón, Spain
susana.bautista@ufv.es

Abstract. The design and development of computer assistive technologies must be tied to the needs and goals of end users and must take into account their capabilities and preferences. In this paper, we present MeDeC@, a Methodology for the Development of Computer Assistive Technologies for people with Autism Spectrum Disorders (ASD), which relies heavily in our experience working with end users with ASD. The aim of this methodology is not to design for a broad group of users, but to design highly customizable tools so that they can be easily adapted to specific situations and small user groups. We also present two applications developed using MeDeC@ in order to test its suitability: EmoTraductor, a web application for emotion recognition for people with Asperger Syndrome, and ReadIt, a web browser plug-in to help people with ASD with written language understanding difficulties to navigate the Internet. The results of our evaluation with end users show that the use of MeDeC@ helps developers to successfully design computer assistive technologies taking into account the special requirements and scenarios that arise when developing this kind of assistive applications.

Keywords: computer assistive technologies · interactive systems · Autism Spectrum Disorder · Asperger Syndrome · User-Centered Design

1 Introduction

The design and development of interactive technologies must be tied to the goals and requirements of end users. This is even more important when the system is aimed to users with cognitive impairments, and more specifically with Autism Spectrum Disorder (ASD), who present very specific goals and restrictions and find unexpected difficulties when using interactive systems. For example, whereas many applications are configured with general profiles that try to cover all the possibilities for a specific group of users, in fact capabilities and individual preferences are not the same for two different people even if they share the same impairment.

Although user-centered methodologies are the most appropriate for this kind of developments, developers usually find it difficult to work with users with ASD. The problem is not only that these users may face extra difficulties to express their needs or desires, but also that they may not be available to be involved during such complex and usually long processes. In the case of users with low functioning cognitive impairments, or even children, the burden of this involvement may be too big [11]. When talking about high functioning cognitive impairments, end users may be reluctant to be involved in research over and over again. In our experience, young adults with Asperger Syndrome, for example, state that they feel like "guinea pigs", and there are studies, such as [13], which describe similar motivations of these users to avoid taking part in research.

Therefore, it is important to employ user-centered methodologies that rely less on the availability of these users and can take advantage of other stakeholders, such as experts, caregivers or tutors. Therefore, these experts must be directly involved in the design process, not only because they can enhance the communication with end users, but because they provide a wider context of the needs and problems faced by a collective of people with impairments that can be important in the development process [2, 18, 19, 25]. Some authors have pointed out the difficulties of working with experts (e.g. [2]), but these difficulties must be faced when either end users are not available or not willing to take part in the study, or when, to some extent, the success of the project depends on the involvement of experts.

Taking this context into account, we present MeDeC@, a Methodology for the Development of Computer Assistive Technologies that adapts classic User-Centered Design approaches to the special requirements of applications for people with ASD. MeDeC@ comprises four main tasks (requirements elicitation, design, implementation and evaluation) that are subdivided in several subtasks. This methodology has been applied to the creation of two applications: *Emo-Traductor*, a web application for emotion recognition in texts aimed at people with Asperger Syndrome, and *ReadIt*, a web browser plug-in to foster digital inclusion of people with ASD. MeDeC@ has been evaluated on two aspects: the appropriateness of the applications for end users' scenarios and their impact on daily activities and the satisfaction of end users and experts with both the process and the results. Results suggest that MeDeC@ can make the development of computer assistive technologies more tied to the requirements and scenarios that arise when developing applications in such a complex and sensitive scope.

The rest of the paper is organized as follows. In section 2 we present a description of previous approaches to formalize the development of computer assistive technologies. Section 3 provides an overview of MeDeC@ and its driving principles. A detailed description of the methodology is then provided in section 4. After that, section 5 describes two case studies of applications developed using MeDeC@. Finally, section 6 includes the discussion of the main contributions of this work, and section 7 depicts some conclusions and future worklines.

2 Related Work

The development of computer assistive technologies has been tackled in very different ways, but currently User-Centered Design (UCD) and Participatory Design (PD) approaches seem to be widely spread [5]. In general terms, UCD focuses on the needs of the user in an iterative process, and its aim is to optimize the final product taking into account the needs, wants, and limitations of end users. Designers not only analyze and predict how users are likely to use the product, but also test the validity of their assumptions through evaluations with end users. On the other hand, in PD users have a more active participation than in UCD, as they become a key group of stakeholders. The idea behind PD is that end users become co-designers by empowering them to propose and generate design alternatives.

A number of assistive systems for people with impairments have been developed using UCD approaches. A three-phase UCD methodology to rapidly develop connected health systems is described in [15]. In the first phase, the authors propose the construction of a document detailing the context of use of the system through the use of storyboarding, paper prototypes, and mock-ups along with user interviews to gather user feedback. In the second phase, they emphasize the use of expert usability inspections, such as heuristic evaluations and cognitive walkthroughs, with small multidisciplinary groups to review the developed prototypes. In the third phase, they propose to use classical testing with end users, using various metrics to measure the user experience and improve the final prototypes. However, the report on the application of this methodology to the development of only one connected health system makes it hard to evaluate its robustness or the adequacy to develop other kinds of systems.

The use of UCD *Personas* [6] to design Augmentative and Alternative Communication (AAC) devices for individuals with speech impairments due to amyotrophic lateral sclerosis is studied in [26]. The authors report on the used method to validate the representativeness and utility of the designed *Personas*. Overall, although the authors present favourable results, they also report on the difficulty to create enough *Personas* for all the different user profiles or to represent users with changing symptoms.

USERfit [1, 24] is a UCD methodology for assistive technology design that provides a framework to ensure that human issues are adequately considered during the design process. This methodology covers the design of the product and the understanding of the context in which a product will be used, and aspects of the support environment (e.g. intended documentation and training). Design activities involve matching requirements to the technological alternatives, and then developing a functional specification for the product to satisfy requirements. Subsequent steps involve design, test and redesign until the design is complete. Once a fully working product is available, USERfit assists in the selection of methods for evaluation as well as establishing evaluation criteria. The USERfit methodology is too broad, as it is aimed at a wide spectrum of assistive technologies based on the Design for All paradigm, and that it does not address in detail the role of experts in the design process, as it sets its main focus on the end users. In addition, USERfit is a summarizing methodology that advices on tools and techniques that can be used in the different steps of the methodology, but it does not describe in detail how to take users and experts into account.

From the point of view of PD approaches, the authors of [23] present the PD process followed to develop interactive technologies for people with ASD to support collaboration and social conversation skills. During the process, ASD children regularly participated in design and evaluation activities to inform the development of the prototypes. The response of ASD children and teachers involved in the pilot test was very positive: children enjoyed the initiative and the teachers found the process useful. According to the authors, one of the main reasons for the good feedback obtained in the pilot test is the significant participation of teachers and children throughout the development of the prototypes. They believe that this involvement has shaped the content, the usability and the suitability of the technologies used.

Cooperative Inquiry (CI) is a PD method where developers, designers and end users come together to explore issues of concern and interest. In CI, everyone must decide which questions should be addressed and what ideas may be of help, eliminating the split between designers, developers and end users. CI is applied to the design of a sports video game with children with learning challenges in [8]. The authors concluded that CI was effective but they recognize that there were limitations, as only ten children participated.

Although PD has been used frequently, some difficulties have been encountered when using it. Sometimes end users are unable or unwilling to collaborate [16, 3] and sometimes there are difficulties that hinder their full integration into the process [11]. As explained in [22], children with ASD are characterized by deficiencies in social interaction (difficulties understanding non-verbal behavior or social cues, difficulties to establish and maintain an adequate peer relationship...), in communication (inability to initiate and maintain conversations, tendency to interpret idiomatic expressions literally...) and by a rigidity of thought and behaviour (need of structured data and routines, inability to generalize information...). Nevertheless, it is important to note that ASD is very heterogeneous, so the difficulties encountered when working with this impairment can be very different depending on the specific users.

Different solutions have been tried to overcome these last limitations. In [12], the CI method is extended to develop IDEAS [4], where drawing templates are used to help children in prototyping and a visual timeline is used in the sessions to clearly define beginning and end. The authors of [20] suggest ways to integrate users with impairments in the PD process. The authors of [10] present a tool to support children with ASD in design critique, an activity in which users and designers discuss the qualities of a design, allowing participants to express their criticisms and contribute to the design in a creative way. This tool was used with seven children to criticize a prototype of the ECHOES system. The tool not only helped children with ASD to annotate the prototype but also facilitated the interaction with the facilitator.

3 Overview of MeDeC@

Over the last few years, we have implemented a number of computer assistive applications for people with ASD, such as a predictive editor for pictogram messages, a text-to-pictogram translator, an application for emotion recognition and an assistant for web navigation. The results and lessons learned in those experiences have been used to formalize MeDeC@ (Methodology for the Development of Computer Assistive Technologies), a UCD methodology that covers all the tasks necessary to create this kind of applications from requirements elicitation to evaluation. We decided to adopt a UCD approach instead of a PD one as we did not have access to end users during all the stages of the development process. The schools and associations we worked with considered that our presence, the interruption of the users' routines or their participation in the design process would affect them quite negatively. As stated previously, PD approaches can only be successful if some access to end users is granted, but when users are not reachable, it is necessary to draw on UCD approaches that rely less on end users and take advantage of other stakeholders, such as experts, caregivers or tutors.

MeDeC@ is built around three basic principles: (1) Personalization, (2) Interaction with Users and Experts, and (3) Service-Oriented Implementation.

3.1 *Principle #1.* Personalization-based Approach

Solutions for people with ASD must be highly configurable, since each person has specific capabilities and needs that do not always coincide with those of others, even within the same impairment. MeDeC@ proposes, following the approach advocated by Harper [14], a design-for-one approach instead of a design-for-all one. Therefore, MeDeC@ is driven by the importance of personalization not only of general functionalities and controls, but of their usage depending on the specific user needs. We propose the customization of the developed applications to the specific needs of associations and educational centers where they will be deployed. At the same time, applications must present mechanisms for their configuration according to the specific requirements of each user or situation in which they will be used.

3.2 *Principle #2.* Interaction with Users and Experts

Although the main end user of assistive technologies is the person with impairments, our experience has shown us that end users are not always available, or they may not be able or willing to collaborate in the design process. In those cases, experts become a valuable resource for obtaining the required information, and will become a collateral user as well. Therefore, MeDeC@ aims to integrate not only end users in the design and implementation processes, but also their tutors, caregivers and/or relatives as experts in their needs and capabilities. Consequently, experts, together with end users, will be consulted and integrated in the tasks related to requirements elicitation, system design, implementation and evaluation. Both actors will provide differentiated and complementary information in each stage: end users are of course authorities in living with their condition, whereas experts do have a wider context of the problems and difficulties faced by a collective, even when these problems may not be perceived by the end users themselves.

3.3 *Principle #3.* Service-Oriented Implementation

MeDeC@ proposes to implement computer assistive applications following a Service-Oriented Architecture (SOA). This kind of software architectures organize the implementation in units of functionality called services that can be accessed remotely and work in an individual fashion. In this way, each service solves a small problem, and the main application uses service composition to perform its complete functionality. We consider that SOA is specially appropriate for the implementation of assistive technologies aimed at people with ASD for two fundamental reasons. First, this approach allows these users to have tools that are adaptable to their personal needs, since the application will help them to decide what services they need and how they should be configured. Second, the designers or programmers of new computer assistive applications will be able to integrate the already created services in their implementations, thus saving an enormous cost in research and development of these accessible technologies.

4 Detailed Description of Tasks in MeDeC@

This section provides insights on the tasks and subtasks proposed by the methodology in a reproducible manner so that other researchers can apply it.

4.1 Requirements elicitation

As a previous step to the design and implementation of an application, developers must obtain and comprehend some basic information about end users and their intended use of the application. This task must be based on the importance of personalization (Principle #1), so the subsequent design phase can be oriented not only to a collective of users but to individual users inside this group. MeDeC@ proposes to collaborate in this task with end users and also with professionals, tutors and/or relatives who work with end users on a daily basis, as we consider them as experts with respect to end users goals, needs and capabilities (Principle #2). End users will therefore provide information about their goals and how the application can help to achieve them, and experts will present a wider context taking into account problems and needs that end users may not be aware of. This collaborative analysis process is essential to achieve the goal of developing tools that are useful for end users, and it is also one of the most challenging ones due to the diverse fields in which the different actors perform. Therefore, it is of utmost importance that developers identify clearly the environment and needs of end users and why a certain assistive tool can be useful for them, whereas experts and end users must understand the affordances and limitations of available technologies. The main output of this task is a document with formalized answers to the following aspects:

- **User goals with respect to the application**: The initial step of this task is to obtain information about users' goals and expectations with respect to the use of the application, that is, what they are going to use the tool for and why it will be useful for them. It is also important to take into account how the use of the application will affect their daily activities.
- **Description of scenarios, environment and activities in which the tool will be used**: The scenarios of use typically describe how and when the tool will be used, as well as the people or other actors that will interact with the users during their usage of the application. These scenarios must include information related to the physical and digital environment and the specific activities that will be performed in each scenario.
- **Technological capabilities and limitations of end users**: An important aspect to be considered from the point of view of end users is to understand their technological capabilities and limitations. For example, whereas some users might be able to use the mouse or the keyboard, other users may need alternative input options such as speech recognition interfaces.

4.2 Design

Once the main purpose of the application has been agreed by experts, end users and developers, this information must be transformed into a detailed specification of both the general behaviour of the tool and its interface and interactions. Although related, these two issues are not the same and must be treated separately. On the one hand, the behaviour of the tool comprises the tasks that can be performed in it and their relative importance to the general operation. On the other hand, the same tasks can usually be accomplished by using different types of controls and interactions (mouse or keyboard, contextual menu or keyboard shortcuts, etc.). These decisions must always take into account Principle #1 (Personalization). As the developers are not the ones that can decide which options are better in each case, they will rely on prototypes and mock-ups to discuss different alternatives with end users and experts, and find the best designs for each aspect of the final tool. This interaction between developers, end users and experts is based on Principle #2. The application design can therefore be subdivided into three subtasks:

- **Design of the behavior of the application**: The high-level description of the application goals and scenarios of use must be used to decide its general behavior. General aspects like platforms and devices where the application will be used must be defined at this point, always having the scenarios, environment and activities already defined as a starting point. For example, if the application will help users with autism to communicate out of school or at home, a mobile app will surely be the best option. At this point, designers must also decide the main functionalities that will be available.

- **Design of the interaction with the application**: The expected behavior and functionalities of the application must be translated into controls and widgets from the point of view of both interactions and interfaces. Questions like how data will be presented, how the users will interact with these data or how they will activate the controls must be answered at this point. Designers will study different answers to these questions, capturing them in a series of mock-ups and prototypes, always taking into account that they can not make the right decisions without the help of end user and experts.
- **Evaluation of the different alternatives with the help of experts and end users**: For both the behavior and interactions of the application, designers must study different alternatives (in the form of mock-ups or prototypes) so they can discuss them with experts and end users. In this subtask, different solutions must be discussed from the point of view of end users capabilities, limitations and expectations. The output of this subtask must be an agreement about how to face the implementation of the working prototypes that will be tested afterwards.

The main output of the design task is a document with sufficient information to start the implementation of different prototypes.

4.3 Implementation

Following Principle #3, we propose the implementation of computer assistive applications following a Service-Oriented Architecture (SOA). Therefore, the design outputs must be translated into development decisions not only from the point of view of the final tool, but also of the small pieces of functionality that will be implemented as services and then used to compose the application. The implementation phase receives the design outputs produced in the previous task as its main input, and it is composed of two subtasks:

- **Implementation of services**: An important part of the implementation task consists in deciding the services that will encapsulate the functionalities that conform the application. For example, in a text simplification application we could find services for the lexical simplification of a word, the syntactical simplification of a sentence, or the summarization of a text. Then, all these pieces could be integrated as different functionalities in the final tool.
- **Implementation of application prototypes**: Different application prototypes will be implemented using the services that have been defined in the previous subtask as basic pieces. When composing the intended application, developers must bear in mind the desired personalization of this kind of applications (Principle #1). Following this kind of architecture, it would be easy to configure the application differently for each user just by selecting which services must be activated in each case, for example.

The main outputs of this task are one or more working prototypes that can be evaluated with users, and an API of fully functional services that are used to compose the developed prototypes.

4.4 Evaluation

In an assistive application, two main aspects must be evaluated: the appropriateness of the application from the point of view of end users goals and scenarios, and the satisfaction of end users and experts with both the process and results. To study these aspects, we propose a user-centered evaluation where experts and end users can explore the application prototypes (Principle #2), so they can try out possible options and find bugs, technical errors or usability flaws. When possible, these evaluation sessions should be organized around the following phases:

- **Prototype presentation**: The prototypes must be presented and explained to end users before they can use them. Depending on their cognitive level or specific impairments, the tools can be presented just before the evaluation, or they can be introduced previously by professionals or tutors in separate sessions prior to the evaluation.
- **Task-oriented testing**: We propose the first part of the evaluation sessions to be task-oriented. This means that the evaluators will prepare a set of predefined tasks that must be completed by end users with the application. This task-oriented testing allows to cover the most important functionalities and scenarios so they can be evaluated in the first place. This approach allows the research team to control the difficulty of the tasks to be carried out, starting with easy tasks and then continuing with a more complex use of the tool during the evaluation session.
- **Free testing**: At some point of the evaluation, preferably after the task-oriented testing, users must be allowed to explore the tool freely. This kind of testing not only allows the research team to study how end users interact with the application when they are not told what to do, but also serves to discover unexpected ways to use it or assumptions about the users that had not been considered before. This step is specially useful when it was possible to involve end users in previous stages of the design process.
- **Acquisition of users opinions and findings**: During the whole evaluation session, developers will rely on different techniques to collect user opinions and findings. Not only questionnaires and interviews are useful, but also techniques like user observation can be very helpful to extract the desired information. All these data will later be analyzed and changes and improvements will be proposed for the next prototypes of the application.

Materials for end users must be adapted to their limitations and needs. To carry out this adaptation, the help of experts is essential as the vocabulary, punctuation scales, possible answers or devices to be used must be adapted to the target group. The output of this task is a set of proposed modifications to any of the sub-products of the other tasks (requirements and design documents and/or prototypes).

5 MeDeC@ in Practice: Case Studies

This section explains in detail how MeDeC@ has been applied to the creation of two computer assistive applications: EmoTraductor, a web application for emotion recognition in written texts aimed at people with Asperger Syndrome, and ReadIt, a web browser plug-in to promote the digital inclusion of people with written language understanding difficulties.

5.1 EmoTraductor: A Web Application for Emotional Text Analysis

EmoTraductor [7] is a web application that automatically detects the presence of each of the five basic emotions (joy, sadness, disgust, fear and anger) in a given text. Not only does the application show the main emotions transmitted in a text using emoticons and colors, but it also highlights the words in the text that are considered emotional and therefore lead to the presented emotions. EmoTraductor was designed and implemented following the MeDeC@ methodology. In this case study, end users did not participate in the preliminary stages of requirements elicitation and design, but only in the evaluation stage. As expressed by the experts involved in the development of EmoTraductor, these users were quite reluctant to participate in experiments over and over again, so they only agreed to be involved in the tool evaluation. Therefore, only experts with a global understanding of the needs of this collective were involved throughout the whole process.

Requirements Elicitation. EmoTraductor was designed and implemented with the help of experts from *Asociación Asperger Madrid*[4], a local association aimed to help people with Asperger Syndrome (AS). We met with a group of experts from the association, from psychologists to social workers, who introduced us to the capabilities and daily life of people with this syndrome. Experts let us know that one of the main difficulties that people with AS face is the detection of the emotions of others, as well as the difficulty to express their own. However, people with AS find themselves in many situations where the correct identification of emotions is fundamental for their social integration. For example, when they publish or comment on a blog, they may express in a way that does not comply with social conventions: if what they read has annoyed them, they will post a comment with a disproportionate tone of anger. This comment could then start a chain of unpleasant responses transmitting different emotions that could be difficult for them to handle. They can also find difficulties in more common situations, such as answering an email: they may misinterpret the tone of an email from their boss or teacher and answer in an inappropriate way.

In all these cases, it would be very useful for them to have an "emotional translator" capable of suggesting them the emotions that are transmitted by the text they are reading or writing. With a tool like this, they could check if the emotion they perceive in a text they are reading is correct or if the emotion transmitted by a text they have written matches the emotion they wanted to convey. Although language is very subjective, and a machine is far from being able to perfectly interpret the emotions of a text, any aid in this path can be very helpful for this collective. Experts also indicated that the application should

[4] http://www.aspergermadrid.org

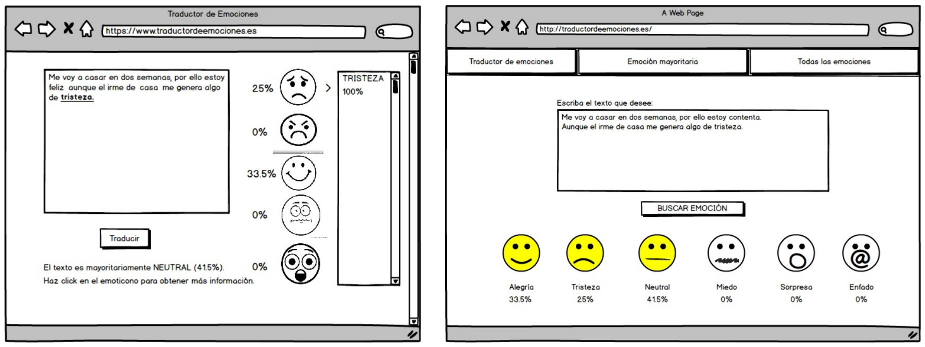
not only express the overall emotion of a text, but also give clues about which elements of the text influence this global emotion. These clues are vital if the user finds that the emotion conveyed by a text is not the desired one, as they will help him to identify what must be changed in order to transmit the desired emotion correctly. Although end users were not directly present in these meetings, experts provided us with real examples of problems experienced by people with AS that could be avoided using a tool with the described functionalities.

From the point of view of the scenarios of use and activities, most members of the association are young adults in their twenties who spend lots of time in social networks and the Internet. They mostly find problems with emotions transmitted in WhatsApp and Telegram groups, entries in Facebook or blogs where they read, write and comment, or communications through email. Conversely, people with AS are usually very capable from the point of view of technology.

Design. Once the main purpose of the application was agreed by experts and developers, we had to transform it into a detailed specification. From the point of view of the general behavior of the application, we decided to implement the translator as a web application. Thus, the application could be used on any device with Internet access. Regarding the emotions detected by our application, we decided to use the five basic emotions: joy, sadness, fear, anger and disgust.

The design of the application was discussed with experts in two different sessions. First, we analyzed with them different options for representing emotions, which was probably the most important decision from the point of view of the design. Three different possibilities were considered: to use emoticons to represent each of the emotions, to use a scale representing the polarity (positive or negative) of the emotion or to use different colors for the words in the text according to the emotion they represent. The experts discarded the polarity scale and the colors in the text from the start, as they thought they could confuse end users, and they suggested that the best choice was to associate each emotion with an emoticon that could be accompanied by a color and a textual label such as *anger* or *joy*.

Then, in order to design the rest of the interface and the interactions with the web application, we employed a parallel design process: each developer designed a mock-up of the application independently. The aim was to explore a series of ideas without developers influencing each other, so there would be more variety of functionalities as well as different designs for the same functionality. Once the individual prototypes were ready, they were studied and compared by the developers, and a final mock-up of the application was built to be analyzed by experts. This mock-up had, in some cases, different options for the same data to be represented or the same functionality to be used (as shown in Fig. 1), so these options could easily be discussed with the experts. Different alternatives for the main decision points of the design were discussed in a new meeting with the group of experts. The main issues that were discussed were the following: how to represent the emotions and its values, where to position the results, and what was the appropriate granularity for this representation.



Me voy a casar en dos semanas, por ello estoy feliz aunque el irme de casa me genera algo de **tristeza.**

El texto es mayoritariamente NEUTRAL (41.5%).
Haz click en el emoticono para obtener más información.

Me voy a casar en dos semanas, por ello estoy contenta.
Aunque el irme de casa me genera algo de tristeza.

(a) (b)

Fig. 1. Two of the mock-ups created during the parallel design process

When asked if the intensity of the emotions could be represented with percentages next to the emoticons, the experts considered that percentages could mislead potential users of the application. They recommended a more visual way with a bar divided by emotions, associating each emotion with a color, and a division of this bar proportionally among the emotions according to the results obtained from the emotional analysis of the text. Taking into account the textual labels accompanying the emoticons, the specific use of the word "neutral" to express the absence of emotions was considered ambiguous. In this case, the experts recommended the use of a message like "this text has no emotions". As emoticons, they recommended the use of the ones provided by ARASAAC[5], since end users were already familiar with them. When discussing the best position for the obtained results (to the right of the text or below the text box), the experts considered that both solutions could work, but they preferred the one with the results below the text. When asked if this should be an option that could be configured, they thought that personalizing this aspect would not add any benefit to the application and could mislead the user. From the point of view of the granularity of the results to be shown, the experts were asked if it would be better to show all the emotions that were present in the text or only the predominant emotion, for which they preferred the former.

In general, the experts considered that the functionalities included in the prototype were adequate for the application to be useful for the daily activities of the end users. They also indicated that personalization is very important in this type of applications. For example, they pointed out that it would be very useful to allow the change of the colors associated with the emotions, since people with AS are more likely to have synesthesia[6] than other population groups. The experts also indicated that it would be very useful to be able to change the

[5] ARASAAC is an extensive database of pictograms for an ample vocabulary in Spanish. More information in: http://www.arasaac.org/

[6] Condition by which a mixture of the senses is experienced. For example, colors are seen when sounds are heard or emotions are felt.

emoticons associated with each emotion, as this could help some users to better interpret the results.

Implementation EmoTraductor was implemented following a service-oriented architecture. The main interface had to be created following the decisions and mock-ups obtained in the design phase, and the functionality had to be decomposed in smaller pieces that could be implemented as web services. The idea was to design these web services so that they could also be used as pieces in future developments and other applications.

A complete API of the emotional web services[7] was implemented considering three levels of text granularity: word, sentence and multi-sentence text. Although there are more auxiliary services, the most important ones are the following:

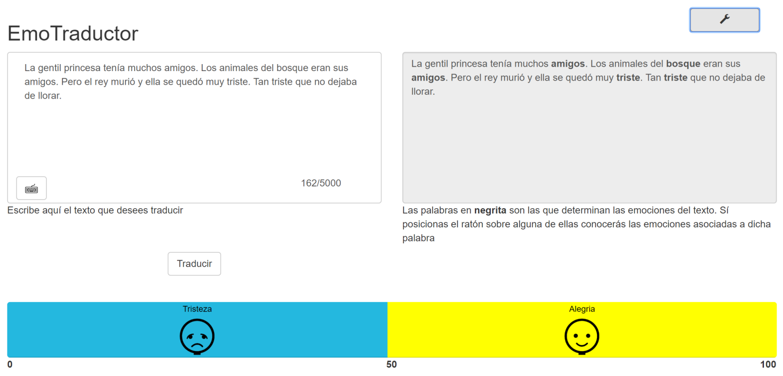
– *Web service to obtain the emotions expressed by a word.* Given a word, this service returns information about the degree of intensity of each emotion in the word, expressed with a value between 1 (lack of emotion) and 5 (full of emotion). This information is extracted from a series of pre-existent emotional dictionaries that associate the basic emotions and a very extensive vocabulary in Spanish. For example, given the word *"enfermedad"* (*illness*), the service returns the following values for each emotion: <sadness: 4.13; fear:3.96; joy:1.0; anger:2.93; disgust:2.43>.
– *Web service to obtain the emotions expressed by a sentence.* This service is based on the previous web service, and returns the emotional values of the sentence along with the list of words that have led to those results. For example, for the sentence *"Estoy alegre y feliz"* (*I'm glad and happy*), the service returns the following values for each emotion: <sadness:1.1; fear:1.21; joy:4.73; anger:1.05; disgust:1.02>, and the following list of emotional words: *<alegre (glad), feliz (happy)>*.
– *Web service to obtain the emotions expressed by a multi-sentence text.* In order to obtain the emotions of a text, this service splits the whole text into sentences and obtains the emotional degrees of each sentence using the previous web service. Enunciative, interrogative and admiration sentences are given different weights when calculating the final emotional values.

A complete functional application[8] was implemented using these web services in a responsive web application, always following the design decisions from the previous phase. The interface of the application can be seen in Figure 2. More technical and detailed information about the implementation of the web services API and the application can be found in [7].

Evaluation Thanks to *Asociación Asperger Madrid*, it was possible to evaluate the EmoTraductor with end users and experts. The main goal of this evaluation was to figure out the suitability of the application for different users, and

[7] The emotional web services API is publicly available in http://sesat.fdi.ucm.es/apiEmoTraductor/

[8] EmoTraductor is publicly available in http://sesat.fdi.ucm.es/traductor/


EmoTraductor

La gentil princesa tenía muchos amigos. Los animales del bosque eran sus amigos. Pero el rey murió y ella se quedó muy triste. Tan triste que no dejaba de llorar.

162/5000

Escribe aquí el texto que desees traducir

La gentil princesa tenía muchos **amigos**. Los animales del **bosque** eran sus **amigos**. Pero el rey murió y ella se quedó muy **triste**. Tan **triste** que no dejaba de llorar.

Las palabras en **negrita** son las que determinan las emociones del texto. Sí posicionas el ratón sobre alguna de ellas conocerás las emociones asociadas a dicha palabra

Fig. 2. Final version of the EmoTraductor interface.

the satisfaction of both end users and experts with the process and the results obtained with the application. The evaluation of EmoTraductor was performed with a total of 9 users with AS and 2 experts from the association.

First, the application was briefly presented to the evaluators. Then, during a task-oriented testing phase, the evaluators were provided with different texts to test in the application (texts lacking emotion, texts expressing a single emotion and texts with several emotions). For each text, evaluators had to indicate in a questionnaire which was the information presented by the application: the emotions in the text, the predominant emotion and the emotional words. The aim of this phase was to verify if users could interact with the application and comprehend the information presented in it. The evaluators were not asked which was the emotion they thought the text conveyed, but which was the emotion the application assigned to the text and words. The aim was to check if the application was intuitive and offered the information in an understandable way, and if the users could understand the results.

Next, the evaluators were asked to explore the application freely in the free testing phase. In this way, they could use their own texts and test if the results would help them to interpret the emotions. At the end of this phase, the evaluators completed a form to assess the usability of the application. All the questionnaires were designed in collaboration with experts. Most of the questions were answered with a 5 point Likert scale, the granularity recommended by the experts, and we made sure that potentially emotion-based subjective responses were not used. In addition to the questionnaires, the developers of the application were observing the evaluators, taking notes of their suggestions and comments during the whole session.

All these data were later analyzed and several conclusions were obtained. The application was positively valued by both experts and end users, including the colors and emoticons used by default and the available personalization options. The emotional bar that shows the emotions related to the text and their degrees was considered intuitive. However, the emotional results obtained for some texts were not accurate in those cases where there were words with ambiguous meanings or negations. Slang language was also problematic for the application. From the point of view of the process, the experts specially valued the effort made during the design and implementation process for taking into account the information about end users, their necessities and requirements.

5.2 ReadIt: A Browser Plug-in to Enhance Web Navigation

ReadIt [17] is a web browser plug-in that supports and enhances web navigation for people with written language understanding difficulties. Considering the characteristics of these users, who may have problems when reading or interpreting a text, this application has been developed in order to facilitate web navigation. Through the use of web services, ReadIt provides a simple set of tools to help these users understand the texts they encounter in web pages. ReadIt was designed and implemented following the MeDeC@ methodology. In this case, end users participated mainly in the evaluation stage because the experts who work with them considered that our presence, the interruption of their routines or their participation in the design process would affect them in a rather negative way. Discussions about requirements elicitation and design were held only with the experts, who gave us a global view of the problems that this collective usually faces.

Requirements Elicitation. ReadIt was designed and implemented with the help of experts from *Estudio 3 Afanias School*[9], a school aimed to attend the special needs of children with learning difficulties that cannot study in a regular school, including children with ASD and other cognitive impairments. We met with teachers and psychologists who helped us to obtain and comprehend some basic information about the end users and their use of Internet technologies, as they usually encounter problems reading, writing or understanding text from web pages. Although there are certain guidelines to make the web more accessible, like choosing simple vocabulary and sentence structures to make the text easier to read, these guidelines are not fulfilled in most web pages. Despite the fact that the end users did not actively participate in the conversations, part of our meetings consisted in observing them working on the computer, so we could grasp some of the difficulties they usually face.

Estudio 3 Afanias students are teenagers under 21 who are educated in both regular studies and adult life according to their intellectual capabilities. The experts selected a specific class of students from 16 to 21 years old who could benefit the most from a tool for enhancing and facilitating web navigation, as they spend many hours consulting the web pages of their favorites singers or actors, or use the Internet as part of their studies. Considering the characteristics of this specific class, who were mostly capable of reading and writing but usually found problems when interpreting a text, the goal of ReadIt was set to providing a set of tools to help them understand the texts they encounter in the web.

[9] https://afanias.org/que-hacemos/educacion/colegio-estudio-3/

From the point of view of the scenarios of use, these users could benefit from tools that allowed them to obtain additional textual information for a word, to generate a summary of a text, or access YouTube videos and Wikipedia articles corresponding to the words they do not understand. It was important to take into account that these students, who were in general capable of using a computer and a web browser, could have problems for certain interactions such us selecting text or using the keyboard. Therefore, the experts emphasized the importance of being able to customize the tool for each of them.

Design. The information obtained from the first task was transformed into a detailed specification of both the general behavior and interactions. As the application was intended as an assistant to browse the web, we decided to implement it as an extension for Google Chrome, since it is currently one of the most popular web browsers. The tool was designed as a series of functionalities that can be activated at any time, such as looking up synonyms or definitions of a word or creating a summary of the content of a web page.

In order to discuss different design options with the experts, we created two mock-ups with alternative solutions to activate the available functionalities, selecting the text that is considered difficult, and the general layout:

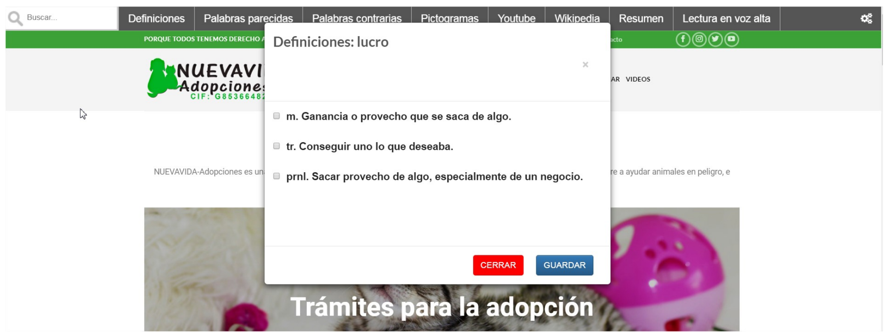
- *Mock-up 1: Right button click.* After activating the extension in the browser, there was no special interface to be shown. When the user selects a difficult word or a short part of the text, the list of functionalities can be activated by using the right mouse button. The response is shown in a modal window.
- *Mock-up 2: Toolbar.* After activating the extension, a new toolbar appears, displaying the different functionalities offered by the application. The users can then select the text as in mock-up 1 and then click on the toolbar button that corresponds to the desired function, or they can write the text in a search field instead. The response is also shown in a modal window.

Experts preferred mock-up 2 without any doubt, as they pointed out that a user with ASD would be confused if after activating the plug-in there was no visual feedback in the browser window. In addition, although *Estudio 3 Afanias* students who are going to use ReadIt are able to work with the keyboard and mouse, the right click interaction is usually difficult for them. Experts considered as a good idea to make it possible to introduce the difficult text by selecting or writing it, so users can decide depending on their capabilities or the situation. They also remarked that it was very important to be able to configure the available functionalities for each user, so they are not confused by the options they are not going to use. In addition, they recommended the use of simple vocabulary for the texts in the plug-in. For example, these users do not understand what a synonym is, but will understand an option called "similar words".

Implementation. Taking into account the conclusions obtained in the design phase, a completely functional instance of the plug-in was implemented[10]. Fig-

[10] ReadIt is publicly available in the Chrome Web Store.

ure 3 shows the ReadIt toolbar with all the functionalities activated and the additional option of writing text in a search space. The toolbar is visible in all websites during the navigation, and the options can be configured easily.



m. Ganancia o provecho que se saca de algo.

tr. Conseguir uno lo que deseaba.

prnl. Sacar provecho de algo, especialmente de un negocio.

Fig. 3. ReadIt plug-in interface with the search of definitions for *"lucro"* (*profit*).

This implementation follows a service-oriented architecture, with eight web services encapsulating the implementation of each functionality available in the plug-in: (1) obtaining the definitions of a word using the Royal Academy of the Spanish Language (RAE) dictionary[11]; (2) getting the synonyms and (3) antonyms of a word; (4) obtaining a pictogram translation of a text; (5) accessing the Youtube video for a word or text; (6) accessing the Wikipedia article for a word or text; (7) generating an automatic summary of the selected text; and (8) listening to a conversion of the written text into spoken words. More technical and detailed information about the implementation of the plug-in can be found in [17].

Evaluation. A user-centered evaluation where experts and end users could explore the application prototype was carried out in *Estudio 3 Afanias School*. We tested the implemented plug-in with 10 students and 2 teachers in their computer classroom. First, a presentation was made to students to show them the application and its functionalities. Then, during the task-oriented testing, we proposed a series of guided activities to use all the options and functionalities in the plug-in. As the class was about to go on a trip to Tarragona, we oriented the activities to different ways of looking for information about what they were going to see, and how to use the plug-in to better understand the web pages they were navigating. Next, users could use the plug-in freely during one hour, always with the help of developers and teachers. Finally, we asked both students and teachers to complete a questionnaire with different questions about the application and their experience. In the case of the students, we had to adapt a typical questionnaire so that people with ASD could understand and complete

[11] http://www.rae.es/

it. The solution was to use a semaphore scale, where they could choose the red color if they did not agree with a statement, yellow if they were not sure, and green if they did. In addition to this final questionnaires, the developers were observing the session and taking notes. After the evaluation, a short debriefing session was performed with the teachers. They were very satisfied about the outcome of the collaboration, and valued the effort of adapting the development of the tool to the special needs of the students in their school.

All the compiled data was later analyzed. The importance of this work was immediately apparent, as we were able to observe first-hand how the application assisted people with different reading comprehension skills. Students felt impressed by the things they could do with the application, and eager to use it both at school and at home. All the functionalities available in the plug-in were considered very helpful by both students and teachers, specially some like the summary or the definitions. The interface was also considered simple and practical, although teachers highlighted that some vocabulary in the menus and options could be simplified. An interesting finding was that ReadIt not only made it easier for users to understand the texts in a web page, but also promoted and encouraged them to improve their learning and independence. For example, a student found an unknown word (*"sarcophagus"*) and was very happy when she discovered the meaning by herself using the tool. The same happened to another student who did not understand the word *"bovine"*.

6 Discussion

The methodology presented in this work relies heavily in our experience. Throughout these years of experience we have learned that organizations that work with people with ASD are very willing to collaborate, but it is usually difficult to have an appropriate access to end users. Therefore, one of the main contributions of our methodology is to recommend the inclusion of experts (as other authors have previously done, such as [9, 27, 28]), along with end users, in all the stages of the development process. This makes it necessary to adapt phases like requirements elicitation or evaluation to the presence of two different actors that provide different and complementary information for the task at hand.

However, we acknowledge that the MeDeC@ methodology relies mainly in the involvement of experts throughout the design process, which may imply several drawbacks according to other authors. For example, the authors of [2, 21] have analyzed some of these drawbacks in the development of two assistive handheld applications. One of the drawbacks has to do with the experts' interests, perspectives and expectations, as they may differ from the researchers' ones or even the users' ones. This has been the case in our experience, although in the two presented case studies we have managed to take their needs into account in the resulting product, as they did not clash with our own. Another drawback that the same authors point out is the possible interference between the roles of domain experts. Again, we have experienced this problem as our domain experts acted both as representatives of and liaisons with the end users. Therefore, we have had to take their needs into account so that we made sure that they would eventually grant us access to the end users to carry out the necessary evaluations of our applications. In addition, in their roles of teachers and caregivers, these experts have been considered as side users of our applications. Finally, the authors of [2, 21] also point out how beneficial it can be to work with existing support organizations. Although it is true that they have provided us with valuable information related to the users' needs, we cannot deny that they have also imposed some limitations that have affected our working plans. First, these organizations have been quite reluctant to let us work with the end users, which is reflected in the MeDeC@ methodology in the prominent role that experts have over end users. Subsequently, this is one of the reasons why we have favoured User-Centered Design over Participatory Design, as a purely PD approach has not been possible without a bigger involvement of the end users. In addition, different organizations had conflicting expectations about the functionality of the final applications, which has made us design different solutions for each of them, as a design-for-all strategy would not have lead us to a satisfactory solution.

Another contribution of our methodology is not to design for a wide collective of users with a specific impairment, but to adapt UCD to specific situations and smaller groups. The applied MeDeC@ approach has the potential to assist developers to rethink the development of computer assistive applications by listening to, and focusing on, the needs and aims of potential users of those applications. The involvement of end users or experts throughout the development process can lead to a more effective design and the development of more accessible applications that obtain a high degree of satisfaction for the end users.

Finally, although more tied to the implementation than to the design, the idea of using a Service-Oriented Architecture with small pieces of reusable code has demonstrated to be highly flexible for the configuration and personalization of computerized assistive tools. The availability of services that solve small accessibility problems simplifies the design process, because once the operations are given, the choices that remain are mostly about interface and interaction. That makes it easier and faster to develop new tools that solve similar problems but must be personalized for a different group of users. Harper [14] expresses a similar idea: the separation between user interface and functionality allows universal access to applications, since the interface can be adapted to the user without modifying the code that implements the functionality. Therefore, we consider that the exploration of this kind of implementation architectures can greatly enhance the development of computerized assistive technologies, with a lower cost and effort and shorter development times.

7 Conclusions and Future Work

This paper has presented a Methodology for the Development of Computer Assistive Technologies (MeDeC@) that adapts the classic UCD approaches to the design and implementation of applications for people with ASD. The MeDeC@ methodology, along with the two applications developed using it, are some of the results obtained in the IDiLyCo (*Digital Inclusion, Language and Communication*) resesearch project. The main objective of the two applications has been to assist users with their difficulties in the use of language. Interestingly enough, the main limitations of both applications come from the Natural Language Processing (NLP) research field. In the case of EmoTraductor, advances in the detection of negation or in emotional analysis are likely to improve the obtained results. In the case of ReadIt, advances in summarization and machine translation are also likely to improve the quality of the texts presented to the users. Even so, the results of both evaluations show that the users are satisfied with their use and that the experts are willing to keep collaborating using MeDeC@.

The use of a service-oriented architecture in MeDeC@ results in services that solve small problems that facilitate their reuse in other applications. We are currently integrating some of the emotional services created for the EmoTraductor emotional translator into the ReadIt plug-in to emotionally mark words, sentences or texts. This integration would have a higher cost with other kind of software architectures. In addition, the architectural approach using web services will allow us to incorporate NLP improvements to the applications without bothering the users, which is likely to help improve their quality of life.

In the near future, our work is planned to evolve in two different directions. In the case of MeDeC@, we are using the methodology to develop other applications, such as a tool that helps people with ASD to understand complex words by comparing them with easier ones or an editor of pictogram boards. In addition, we are getting in touch with more organizations in order to test the suitability of MeDeC@ with a wider range of cognitive impairments. Regarding the applications, our aim is to create a repository of web services that can be composed to build configurable applications in a fast and simple way, helping developers and users to satisfy their needs as effortlessly as possible. We are also preparing an observational longitudinal study in collaboration with schools in order to better understand the benefits that the developed applications may imply for the end users.

Acknowledgements

The work presented in this paper has been partially funded by the projects IDiLyCo: Digital Inclusion, Language and Communication, Grant. No. TIN2015-66655-R (MINECO/FEDER) and InVITAR-IA: Infraestructuras para la Visibilización, Integración y Transferencia de Aplicaciones y Resultados de Inteligencia Artificial, UCM Grant. No. FEI-EU-17-23. The authors want to thank Lorena Jiménez, Paloma Gutiérrez, Elena Kaloyanova and Gema Eugercios for their work in the implementation of the presented case studies.

References

1. Abascal, J., Nicolle, C.: The application of userfit methodology to teach usability guidelines. In: Tools for Working with Guidelines, pp. 209–216. Springer (2001)

2. Allen, M., Leung, R., McGrenere, J., Purves, B.: Involving domain experts in assistive technology research. Universal Access in the Information Society **7**(3), 145–154 (2008)
3. Allsop, M.: Involving children in the design of healthcare technology. Ph.D. thesis, University of Leeds (2010)
4. Benton, L., Johnson, H., Ashwin, E., Brosnan, M.J., Grawemeyer, B.: Developing IDEAS: supporting children with autism within a participatory design team. In: CHI'12: The 2012 ACM annual conference on Human Factors in Computing Systems (2012)
5. Börjesson, P., Barendregt, W., Eriksson, E., Torgersson, O.: Designing technology for and with developmentally diverse children: A systematic literature review. In: Proceedings of the 14th International Conference on Interaction Design and Children. pp. 79–88. ACM, New York, NY, USA (2015)
6. Cooper, A., Reimann, R., Cronin, D., Noessel, C.: About Face: The Essentials of Interaction Design. Wiley Publishing, 4th edn. (2014)
7. Eugercios, G., Gutiérrez, P., Kaloyanova, E.: Análisis emocional para la inclusión digital (2018), https://eprints.ucm.es/48834/
8. Foss, E., Guha, M.L., Papadatos, P., Clegg, T., Yip, J., Walsh, G.: Cooperative inquiry extended: Creating technology with middle school students with learning differences. Journal of Special Education Technology **28**(3), 33–46 (2013)
9. Frank Lopresti, E., Mihailidis, A., Kirsch, N.: Assistive technology for cognitive rehabilitation: State of the art. Neuropsychological rehabilitation **14**(1-2), 5–39 (2004)
10. Frauenberger, C., Good, J., Alcorn, A., Pain, H.: Supporting the design contributions of children with autism spectrum conditions. In: Proceedings of the 11th International Conference on Interaction Design and Children. pp. 134–143. ACM (2012)
11. Frauenberger, C., Good, J., Keay-Bright, W.: Designing technology for children with special needs: bridging perspectives through participatory design. CoDesign **7**(1), 1–28 (2011)
12. Guha, M.L., Druin, A., Fails, J.A.: Designing with and for children with special needs: An inclusionary model. In: Proceedings of the 7th International Conference on Interaction Design and Children. pp. 61–64. IDC '08, ACM, New York, NY, USA (2008)
13. Haas, K., Costley, D., Falkmer, M., Richdale, A., Sofronoff, K., Falkmer, T.: Factors influencing the research participation of adults with autism spectrum disorders. Journal of autism and developmental disorders **46**(5), 1793–1805 (2016)
14. Harper, S.: Is there design-for-all? Universal Access in the Information Society **6**(1), 111–113 (2007)
15. Harte, R., Glynn, L., Rodríguez-Molinero, A., Baker, P.M., Scharf, T., Quinlan, L.R., ÓLaighin, G.: A human-centered design methodology to enhance the usability, human factors, and user experience of connected health systems: a three-phase methodology. JMIR human factors **4**(1) (2017)
16. Holone, H., Herstad, J.: Three tensions in participatory design for inclusion. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2903–2906. CHI '13, ACM, New York, NY, USA (2013)
17. Jiménez-Corta, L.: Herramienta de apoyo a la navegación web para personas con discapacidad (2018), https://eprints.ucm.es/48863/
18. Keay-Bright, W.: The reactive colours project: demonstrating participatory and collaborative design methods for the creation of software for autistic children. Design – Principles & Practices **1**(2), 7–15 (2007)

19. Kientz, J.A., Hayes, G.R., Westeyn, T.L., Starner, T., Abowd, G.D.: Pervasive computing and autism: Assisting caregivers of children with special needs. IEEE Pervasive Computing **6**(1), 28–35 (2007)
20. Lazar, J., Feng, J.H., Hochheiser, H.: Research methods in human-computer interaction. Morgan Kaufmann (2017)
21. Leung, R., Lumsden, J.: Designing mobile technologies for individuals with disabilities. In: Handbook of Research on User Interface Design and Evaluation for Mobile Technology, pp. 609–623. IGI Global (2008)
22. Murray, D., Lesser, M., Lawson, W.: Attention, monotropism and the diagnostic criteria for autism. Autism **9**(2), 139–156 (2005). https://doi.org/10.1177/1362361305051398
23. Parsons, S., Millen, L., Garib-Penna, S., Cobb, S.: Participatory design in the development of innovative technologies for children and young people on the autism spectrum: the cospatial project. Journal of Assistive Technologies **5**(1), 29–34 (2011)
24. Poulson, D., Richardson, S.: USERfit – a framework for user centred design in assistive technology. Technology and Disability **9**(3), 163–171 (1998)
25. van Rijn, H., Stappers, P.J.: Expressions of ownership: motivating users in a co-design process. In: Proceedings of the Tenth Anniversary Conference on Participatory Design 2008. pp. 178–181. Indiana University (2008)
26. Subrahmaniyan, N., Higginbotham, D.J., Bisantz, A.M.: Using personas to support augmentative alternative communication device design: A validation and evaluation study. International Journal of Human–Computer Interaction **34**(1), 84–97 (2018)
27. Sullivan, J., Fischer, G.: Mobile architectures and prototypes to assist persons with cognitive disabilities using public transportation. In: 26th International Conference on Technology and Rehabilitation (2003)
28. Tee, K., Moffatt, K., Findlater, L., MacGregor, E., McGrenere, J., Purves, B., Fels, S.S.: A visual recipe book for persons with language impairments. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 501–510. ACM (2005)

View publication stats