End-User Development, End-User Programming and End-User Software Engineering: a Systematic Mapping Study

Barbara Rita Barricelli[1], Fabio Cassano[2], Daniela Fogli[3], Antonio Piccinno[2]

[1]Dip. di Informatica, Università degli Studi di Milano, Italy, barricelli@di.unimi.it
[2]Dip. di Informatica, Università di Bari "Aldo Moro", Italy, {fabio.cassano1, antonio.piccinno}@uniba.it
[3]Dip. di Ingegneria dell'Informazione, Università degli Studi di Brescia, Italy, daniela.fogli@unibs.it (corresponding author)

Abstract

End-User Development (EUD), End-Programming (EUP) and End-User Software Engineering (EUSE) are three related research fields that study methods and techniques for empowering end users to modify and create digital artifacts. This paper presents a systematic mapping study aimed at identifying and classifying scientific literature about EUD, EUP and EUSE in the time range January 2000 - May 2017. We selected 165 papers found through a manual selection of papers from specific conferences, journal special issues, and books, integrated with an automatic search on the most important digital libraries. The answer to our research question was built through a classification of the selected papers on seven dimensions: type of approach, interaction technique, phase in which the approach is adopted, application domain, target use, class of users, and type of evaluation. Our findings suggest that EUD, EUP and EUSE are active research topics not only in Human-Computer Interaction, but also in other research communities. However, little cross-fertilization exists among the three themes, as well as unifying frameworks and approaches for guiding novice designers and practitioners. Other findings highlight trends and gaps related to the analysis' dimensions, which have implications on the design of future tools and suggest open issues for further investigations.

Keywords: systematic mapping study, end-user development, end-user programming, end-user software engineering.

1. Introduction

Since the last twenty years, researchers from all over the world have studied several different approaches on how end users can tailor software programs to their needs or create new software artifacts, to solve their professional or personal problems. Most of these techniques have been proposed in the Human-Computer Interaction (HCI) field, and in particular they refer to End-User Programming (EUP), End-User Development (EUD), and End-User Software Engineering (EUSE).

Burnett and Scaffidi consider EUP as the subset of EUD that is the most mature (Burnett & Scaffidi, 2013). Also on the basis of the definition provided in (Ko et al., 2011), they regard EUP as a set of techniques that enable end users to create their own programs. In this way, EUP aims at empowering end users to be able to write programs by adopting special-purpose programming languages, such as those included in spreadsheets or web authoring tools, or professional programming languages, such as C or Java (Myers et al., 2006). EUP also encompasses techniques such as programming by demonstration, visual programming, and scripting languages.

EUD, on the other hand, aims at empowering end users to develop and adapt systems at a level of complexity that is adequate to their practices, background and skills (Lieberman et al., 2006a). Therefore, it pays attention to system flexibility and modifiability, as well as it encompasses domain-specific environments for software creation. Therefore, EUD is not only concerned with programming activities, like EUP, but it spans the entire software development lifecycle. In other words, the goal of EUD is to make users able to participate in their own software artifacts design and development, not only at design time, but also during their actual use. This distinguishes EUD from participatory design that in turn foresees users' participation at design time only (Simonsen & Robertson, 2013). Moreover, thanks to the many possibilities provided by recent technology, such as Internet of Things and smart devices, the term EUD has acquired a broader meaning covering methods, situations, and socio-technical environments that allow and empower end users to express themselves and being independent of high-tech scribes (Fischer et al., 2017) in crafting both software artifacts and hardware technology (e.g., smart objects). For this reason, in the following we use the general term *digital artifacts.*

EUSE is another concept overlapping with EUD (Burnett & Scaffidi, 2013). It takes a different perspective with respect to EUP and EUD because it focuses on systematic and disciplined activities carried out throughout the system lifecycle to guarantee the quality of the code created by end users. In particular, EUSE proposes techniques derived from traditional software engineering, which are aimed at fostering reliability, efficiency, reuse, debugging support, maintainability, and version control (Burnett, 2009).

A few studies have been conducted in recent years aimed at scrutinizing and comparing different approaches to EUP, EUD and EUSE, e.g. Ko et al. (2011), Maceli, (2017), Paternò (2013), Tetteroo & Markopoulos (2015). However, a limited number of approaches and techniques have been found and analyzed, since paper retrieval was mainly guided by researchers' a priori knowledge of these fields. For example, the work of Ko et al. (2011) is mainly focused on studies on EUSE, whilst the other authors principally investigate EUD with reference to EUD-related conferences and journals only.

There is thus a need for a more systematic identification of papers describing and evaluating approaches and techniques related to the above-mentioned fields, in order to obtain an in-depth analysis and classification of the research. To this end, this paper presents a systematic mapping study that aims at answering this general Research Question (RQ):

RQ*: What approaches and techniques have been proposed in literature to support end users in tailoring, extending and creating digital artifacts?*

The final goal is to categorize and summarize the knowledge currently available in literature around the field of "development by end users", in order to identify gaps in current research for suggesting areas of investigation and for providing knowledge to novice research practitioners in this field.

The adopted methodology is inspired to the work reported in (Budgen et al., 2008; Fernandez et al., 2011; Kitchenham et al., 2009) and is based on a mixture of manual selection and automatic search. Our study pays attention to a variety of conferences and journal special issues on EUD, even though the automatic search considers also "end-user programming" and "end-user software engineering" as search keywords.

The paper is structured as follows: Section 2 presents related work; Section 3 describes the methodology adopted to carry out the systematic mapping study; Section 4 presents the results of the study; Section 5 discusses the main findings and the limitations of the work; Section 6 provides the conclusions of the paper.

2. Related Work

An interesting introduction to the motivations and concepts related to EUD can be found in (Burnett & Scaffidi, 2013). These authors underline how EUD is inherently different from traditional software development and that it is often not enough to support EUD by simply mimicking traditional software engineering approaches. Indeed, end users are usually expert in some particular domain, but they do not have training in programming languages and development processes, nor are interested in learning them. Consequently, to support EUD activities, one must provide end users with appropriate tools, communication infrastructures and development frameworks that are easy to use and to learn, and that are also easily integrated into the domain practice.

The introduction to the first book on EUD by Lieberman et al. (2006a) is one of the first attempts to classify different types of techniques for supporting end-user development, by exploiting the classification of end-user activities proposed by Costabile et al. (2003). EUD research is then analyzed by distinguishing between research on end-user participation at *design time* and end-user development at *use time*. As to the latter, the authors discuss the different modification levels with increasing complexity and power of expression that systems may support. Another aspect is concerned with the creation of languages that are suited to non-programmers, in particular domain-specific languages and graphical languages. Related to this theme, Kelleher and Pausch (2005) presents a survey of programming environments and languages for novice programmers, by creating a taxonomy that focuses largely on learning goals. The analysis of the systems is carried out by considering the programming constructs they support and how they make programming more accessible to novice programmers. Therefore, the perspective is not on end users and their work practices, but on the facilitation of teaching and learning programming languages. However, Paternò (2013) observes that the idea of simplifying development through, for example, visual programming has not always led to provide end users with the skills and capabilities necessary for developing their own applications.

As mentioned before, the survey presented in (Ko et al., 2011) is focused on EUSE methods. The authors start from the consideration that, beyond professional programmers, a huge number of people, not expert in programming, use spreadsheets and databases at work by writing formulas and queries that facilitate their daily activities (Scaffidi et al., 2005), as well as writing simulations in MATLAB (Gulley, 2006) or scripts to process scientific data (Fischer et al., 2009). In all these cases, end users often create programs containing errors that they are not aware of or that are difficult to discover. As a consequence, the software engineering community raises many doubts on the value of EUP, by underlining the role of skilled, professional software developers in guaranteeing software correctness, efficiency, maintainability, and security (Harrison, 2004). Because of these quality issues, EUSE has been proposed as an umbrella encompassing EUP practices and technologies that support end users to improve software quality (Ko et al., 2011). Therefore, the survey of Ko et al. presents an analysis based on the software lifecycle, by organizing more than one decade of research on end-user oriented methods for requirements specification, design, testing, verification, and debugging. Several issues related to EUSE are also discussed, such as the role of risk and reward on end users' decision making, as well as individual factors like self-efficacy and gender (Ko et al., 2011).

The first survey specifically focused on end-user development is proposed by Paternò (2013). This paper presents the different motivations for EUD, a brief history of EUD, and the key concepts related to EUD including, among others, domain-oriented design environments (Eisenberg & Fischer, 1994), mutual development

(Andersen & Mørch, 2009), co-development (Costabile et al., 2009) and participatory design (Muller & Kuhn, 1993). The survey by Paternò then addresses recent developments in EUD, such as EUD for web and mobile applications, which are not considered in (Ko et al., 2011). Today, new scenarios are opening up for EUD, from Ambient Intelligence (Cabitza et al., 2017; Desolda et al., 2017; Fogli et al., 2016; Martín et al., 2015; van Doorn et al., 2008) to the Internet of Things (Akiki et al., 2017; Barricelli & Valtolina, 2015; Hafidh et al., 2017; Kim et al., 2017) from tangible and ubiquitous computing (Bellucci et al., 2017; Garzotto & Gonella, 2011; Lee et al., 2013; Turchi et al., 2017) to the Do-It-Yourself movement (Anderson, 2012; Sas & Neustaedter, 2017). In our systematic mapping study, we consider the most current trends of EUD and novel domains where it is applied. In addition, we extend the analysis provided in (Paternò, 2013) by defining several dimensions to explore the general RQ and map the works according to such dimensions.

Our study mainly aims at analyzing and classifying the approaches and techniques proposed in literature for supporting end users in developing activities. Therefore, it complements the review presented in (Tetteroo & Markopoulos, 2015), which instead focuses on the *research methods* applied in the fields of EUD, EUP, EUSE and meta-design in the period 2004-2013. That review includes *meta-design*, which is one of the most influential frameworks for supporting EUD (Fischer & Giaccardi, 2006): it is aimed at creating the socio-technical conditions that allow the owners of problems (end users) to be actively engaged in the *continuous development* of personally meaningful socio-technical systems and act as designers at use time (Fischer & Giaccardi, 2006; Fischer et al., 2004). Tetteroo and Markopoulos's review analyses a total of 93 papers manually retrieved from relevant conferences and journals. To this end, it adopts the classification scheme for HCI research proposed in (Kjeldskov & Graham, 2003), which is articulated according to two dimensions: research methods and research purposes. Eight research methods are considered, namely, case studies, field studies, action research, laboratory experiments, applied research, basic research, and normative writings; whilst, research purposes include understanding, engineering, re-engineering, evaluating, and describing. Findings of the review suggest that research is dominated by the engineering of systems and laboratory evaluations, whilst there is an evident lack of action research and basic research. The authors underline how action research could be particularly suitable to the evaluation of EUD systems in natural environments (Tetteroo & Markopoulos, 2015); therefore, it points to a lack of understanding the very nature of EUD activities in end users' personal and work life. Some of our RQ dimensions aim at investigating this issue more deeply.

The work that is most similar to ours, but at a more preliminary stage, is that one presented by Maceli (2017). Maceli's survey focuses on technologies proposed in EUD literature and to this purpose it analyses in detail 73 papers from 2004 to 2016 that fall in the categories "engineering" and "re-engineering" discussed in (Tetteroo & Markopoulos, 2015). The list of selected papers is derived from four EUD-related conference proceedings and five journals, by considering authors' keywords "end-user development", "end-user programming", "end-user software engineering" and/or "meta-design". Of the 73 papers, 48 were considered also in the survey of Tetteroo and Markopoulos (2015), whilst other 25 were added in Maceli's study. Each paper has been analyzed by the author and her graduate assistant, in order to identify the purpose of the work, the tool proposed, the general category of the tool and the means of its evaluation (if any). Thirteen broad categories of technology tools have thus been obtained through inductive qualitative analysis; among them, mashup tools, programming environments and frameworks, spreadsheet tools, and web authoring tools represent the clear areas of research that emerge from paper analysis, since most of the tools fall in these categories. The findings also suggest that the types of tools do not vary significantly over time and that there is relative little research on novel user interface paradigms, such as tangible or voice interfaces, and social and crowdsourcing applications. An interesting hypothesis by Maceli is that researchers publish EUD-related papers in different venues, known to the wider HCI community and also outside this community. Such important literature work is not considered in Maceli's survey and thus represents a limitation that we aim to overcome with our systematic mapping study.

Finally, Fischer et al. (2004) have mainly explored the concept of meta-design for socio-technical environments, which require to be open, flexible and capable to evolve at the hands of the end users. In this way, meta-design extends the traditional notion of system design to encompass support for system evolution at use time, namely social and technical mechanisms for EUD. Fischer and Giaccardi (2006) describe a set of conceptual frameworks and environments, including the Seeding, Evolutionary growth and Reseeding (SER) process model (Fischer & Ostwald, 2002) and the Domain-Oriented Design Environments (DODEs) (Eisenberg & Fischer, 1994), as well as some interesting applications of meta-design in different domains, such as interactive art, social creativity and learning communities. The concept of meta-design has been recently revisited and broadened in (Fischer et al., 2017): here meta-design is conceived not only as the study and development of enabling technologies for EUD, but also as all mechanisms that sustain a cultural transformation (Fischer, 2013). Therefore, the primary objective of meta-design is to allow and support end users to become end-user developers of all software and hardware systems that pervade their everyday life, such as smartphones, smart watches, interactive displays, and all other smart devices that they may find in their houses or offices (Fischer et al., 2017). These recent trends in the application of the EUD idea are considered in the present systematic mapping study.

3. Methodology

This study has been carried out by following the guidelines for systematic mapping studies proposed in (Budgen et al., 2008; Kitchenham et al., 2009). A systematic mapping study provides an objective procedure for identifying and classifying the papers published in a given research field. Therefore, after defining the research question of the study and the search process, exclusion criteria are defined for data selection. Selected papers are then analyzed for the sake of classification and findings about trends and gaps are finally discussed.

3.1 Research Question

Given the general RQ: “What approaches and techniques have been proposed in literature to support end users in tailoring, extending and creating digital artifacts?”, introduced in Section 1, the following dimensions have been considered for organizing the mapping study:

D1: *Types of approaches proposed.*

This dimension would like to find which approach, among EUD, EUP and EUSE, has been proposed to support end users in tailoring, modifying or creating digital artifacts.

D2: *Types of techniques proposed.*

This dimension would like to discover the types of techniques proposed and applied in the frame of the approaches identified with D1. By “technique” we mean the way made available to the user to carry out the system shaping activity: it may be a particular interaction style, or a metaphor adopted in the user interface.

D3: *Phase in which the shaping activities are carried out.*

This dimension aims to analyze when the shaping activities are most frequently carried out by end users, i.e., design time, use time or both.

D4: *Application domain in which the shaping activities are employed.*

This dimension is related to discovering the different application domains where the shaping activities have been supported.

D5: *Target use of the shaping activities*.

This dimension would like to discover the addressed target uses of the shaping activities.

D6: *Classes of end users to which the proposed solutions are addressed.*

This dimension aims to analyze which are the classes of users that have been supported by the proposed solutions.

D7: *Empirical validation of proposed solutions*.

This final dimension is aimed to find out which kinds of empirical evaluation have been adopted to validate the proposed solutions.

3.2 Search Process

The study has been carried out by selecting the papers of interest through a manual search on important related venues and then through an automatic search performed on the main digital libraries. In the following the two searches are reported.

3.2.1 Manual Search

To perform the manual search, we first considered the following sources (Table 1):

- Proceedings of International conferences specifically focused on EUD, EUP, and EUSE themes;
- Special issues of scientific journals on EUD, EUP, and EUSE;
- Books (collections of multi-authored chapters) on EUD, EUP, and EUSE.

230 papers were manually extracted from the above sources.

Table 1. Conferences Proceedings, Journal (special issues) and Books (collections of multi-authored chapters) on EUD, EUP, and EUSE.

Source	Year	Type
Special Issue, Computer-Supported Cooperative Work, 9(1), Springer	2000	Journal
Special Issue, Communications of the ACM, 47(9), ACM Press	2004	Journal

End-User Development, Book Series "Human-Computer Interaction", Springer	2006	Book
Special Issue, Journal of Organization and End-User Computing, 18(4), IGI Global	2006	Journal
Workshop on End-User Software Engineering (WEUSE 2008), Leipzig, Germany, ACM Press	2008	Proceedings
Int. Symposium on End-User Development (IS-EUD 2009), Siegen, Germany, LNCS 5435, Springer	2009	Proceedings
Special Issue, Journal of Organization and End-User Computing, 22(2), IGI Global	2010	Journal
Int. Symposium on End-User Development (IS-EUD 2011), Torre Canne, Italy, LNCS 6654, Springer	2011	Proceedings
Int. Symposium on End-User Development (IS-EUD 2013), Copenhagen, Denmark, LNCS 7897, Springer	2013	Proceedings
Int. Symposium on End-User Development (IS-EUD 2015), Madrid, Spain, LNCS 9083, Springer	2015	Proceedings
Int. Symposium on End-User Development (IS-EUD 2017), Eindhoven, Netherlands, LNCS 10303, Springer	2017	Proceedings
Special Issue, Journal of Visual Languages and Computing, 40, Elsevier	2017	Journal
Special issue, ACM Transactions on Human-Computer Interaction, 24(2), ACM Press	2017	Journal

3.2.2 Automatic Search

We used the web application SEOBook Keyword Density Analyzer[1] to detect the most recurrent words, 2-word phrases and 3-word phrases in titles and abstract of the papers selected through the manual search.

On the basis of the results of the keywords extraction, we composed the following query:

((End-User Development *OR* End User Development *OR* EUD)

OR

(End-User Programming *OR* End User Programming *OR* EUP)

OR

(End-User Software Engineering *OR* End User Software Engineering *OR* EUSE)

We used this search string to perform automatic searches on the following digital libraries:

- ACM Digital Library[2]
- IEEE Xplore Digital Library[3]
- Springer Link[4]

[1] SEOBook Keyword Density Analyzer: http://tools.seobook.com/general/keyword-density/
[2] ACM Digital Library: dl.acm.org
[3] IEEE Xplore Digital Library: http://ieeexplore.ieee.org/Xplore/home.jsp

- ScienceDirect[5]

We restricted the research on a specific period of time: January 2000 - May 2017.

The automatic search provided 2487 results. Therefore, as a whole, 2717 papers (230 manually selected papers plus 2487 resulted from the automatic search) were selected for subsequent analysis.

3.3 Paper Selection

We considered each of the 2717 papers in order to decide whether or not it was to be included in this study. The paper selection phase consisted in the application of two sets of exclusion criteria we defined at the beginning of the study.

An exclusion process like the one we decided to apply in this study may suffer from a bias due to the presence of researchers with different expertise in this domain. To avoid the most of inaccuracy in data extraction and mis-classification of papers, in both exclusion stages we crosschecked the obtained results as follows: each one of the four involved researchers was assigned with 1/4 of the papers to be analyzed, and once the analysis was concluded, another researcher revised the results. The set of papers assigned to each researcher during the first stage of the selection was then changed during the second stage. Moreover, every time a researcher felt unsure about the evaluation of a paper, the discussion was opened to one or two other researchers.

The first stage of the selection was based on the analysis of title, abstract, keywords, source, and type of each paper. At this stage, we excluded the papers that met at least one of the following exclusion criteria:

- Duplicated results
- Off-topic papers
- Monographs
- Papers published as Technical Reports
- PhD dissertations and Master theses
- Papers published in doctoral consortium venues
- Introductory papers for special issues, books and workshops
- Papers included in conference proceedings but classified as posters, demos or panels
- Papers republished (as-is) on other journals or as book chapters or in collections of papers

---

[4] Springer Link: https://link.springer.com/

[5] ScienceDirect: https://www.sciencedirect.com/

- Papers not written in English
- Invited papers.

Discrepancies in the exclusion decision were solved by searching for consensus among the researchers with a shallow reading of the papers under analysis.

At the end of the first stage of selection we excluded 964 papers, thus 1753 papers out of 2717 were still under consideration.

Then, a second stage of the selection was applied in order to select papers that most probably could answer our research question. It therefore consisted in the application of a second set of exclusion criteria:

- Papers presenting only recommendations, guidelines, or principles for end-user development, end-user programming, end-user software engineering.
- Papers presenting only conceptual models, comparative studies or surveys.
- Papers presenting preliminary ideas on tools or interaction techniques.
- Papers in which the addressed end users are professional developers.
- Papers presenting studies without any evaluations.
- Papers reporting the same approaches or techniques in different venues (only the most important venue was taken into account, possibly considering the most recent publication).
- Papers reporting the same approaches or techniques in different stages (only the most complete one was kept).

Specifically, in order to successfully apply the last two criteria, we developed a script that analyzed the list of papers and automatically highlighted all the papers written by the same authors. In this way, it was easier to check if the same author contributed to the publication of the same approach/technique in more than a venue or if the same approaches and techniques were published in different stages.

After the second stage, **165** papers out of 2717 were selected for our mapping study (we excluded other 1588 papers). Table 2 reports the distribution of such papers according to type of source (conference proceedings, book or journal).

Table 2. Distribution of selected papers.

Publication type	# papers
Papers in Conference proceedings	101
Book chapters	4

Papers in Journals (special issues)	60
Total	**165**

Fig. 1 summarizes the search strategy and the selection carried out at the different stages of evaluation.

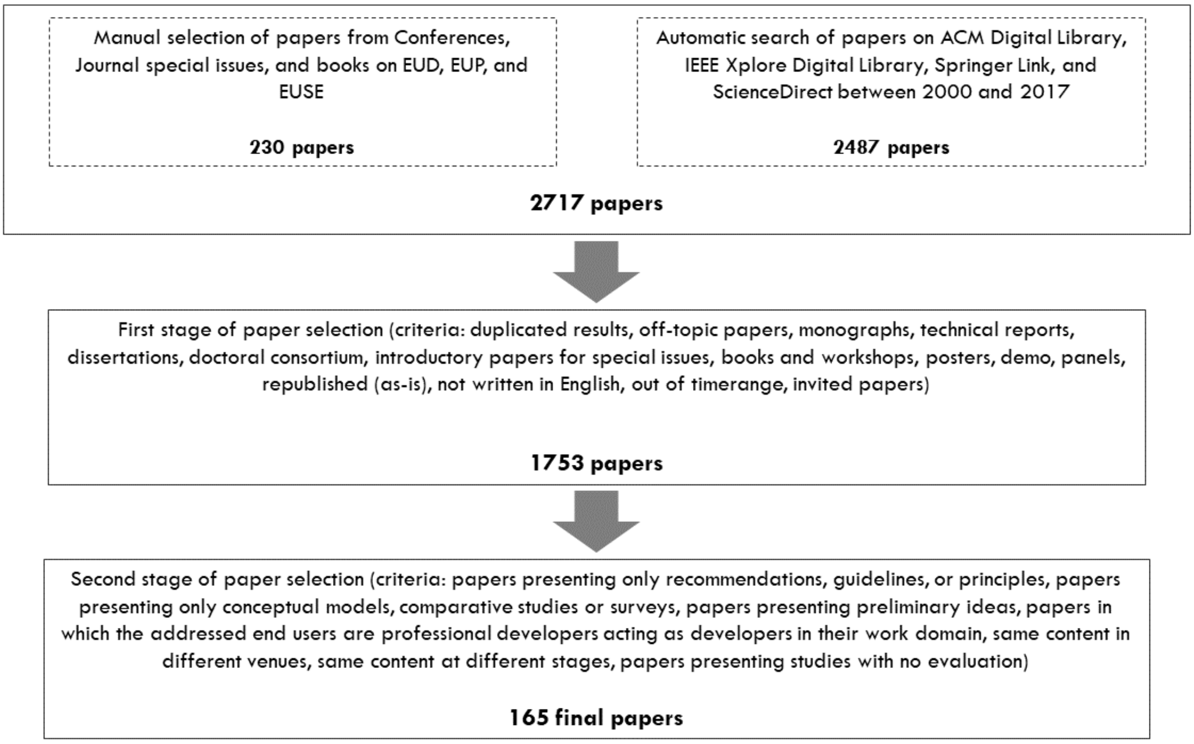

Fig. 1. Search and selection process.

3.4 Data Extraction Strategy

The 165 papers selected in the selection phase represent the set of documents to be analyzed for finding an answer to the RQ. They have been read and analyzed in detail. Specifically, we analyzed and classified each paper according to the seven RQ dimensions:

D1. *Type of approach*: each paper could present one or more approaches among End-User Development, End-User Programming, and End-User Software Engineering. We based this classification on what the papers' author(s) declared in the papers, i.e. we did not decide about the assignment of a specific approach – we trusted the author(s) judgement/opinion. This choice reflects our intention to estimate also the perception that the researchers in this field have about EUD, EUP and EUSE potentials and their application in practice.

D2. *Type of technique*: we identified one or more techniques proposed in the paper for enabling end users to modify and/or create digital artifacts.

D3. *Phase*: we classified the paper according to the time in which EUD, EUP, or EUSE was adopted, i.e., design time, use time, or both.

D4. *Application domain*: we identified the application domain in which the proposed solution was adopted.

D5. *Target use*: for this classification, we identified the purpose of the proposed solution, for instance, whether it was for personal use or educational purpose, by possibly considering the examples provided by the authors or, if missing, the applicability of the solution according to our interpretation.

D6. *Class of users*: with this classification, we specified which community of users was addressed by the proposed solution.

D7. *Empirical validation*: according to our exclusion criteria, we considered only papers presenting empirical validation of the research. Then, we classified each of them considering the following three categories: formal experiment with users, informal experiment with users, and pilot study. In particular, "formal experiments" are rigorous user studies with several participants, which provide qualitative and quantitative results. With "informal experiment" we mean a workshop, or a user study aimed at gathering qualitative data, namely feedback on pros and cons of the application, or an empirical investigation carried out by simulating users' behavior. Pilot studies are instead preliminary evaluations with few users aimed at measuring some usability attributes (e.g., efficiency, effectiveness, etc.); this type of validation was explicitly indicated in the considered papers.

To carry out the analysis, papers have been equally divided among the four authors. The classification process was conducted by each researcher on the assigned subset of 41 or 42 papers and reported by filling in Excel sheets with all data related to the seven dimensions. First, each researcher independently analyzed her/his subset of papers and proposed a classification. Then, the subsets were exchanged among the researchers and analyzed again by a different researcher. Double-scoring was conducted on the results of the classification process yielding an initial value of the inter-rater reliability superior to .85. Doubtful situations have been solved through videoconference or face-to-face meetings and all differences were solved by discussion. A unique Excel sheet was finally created.

Then, a videoconference meeting involving the four researchers was taken with the aim of making terminology coherent; in particular, limited sets of classes were defined for techniques (D2), application domains (D4), target use (D5) and classes of users (D6) (see Table 3). This activity was performed a posteriori, because it was guided by the preliminary classification individually carried out by each researcher.

Table 3. Classes of techniques, application domains, target uses and target users.

Techniques	Application domains	Target uses	Classes of users
Annotation-based Assertion-based Component-based Digital sketching Gesture-based Model-based Natural language Programming by demonstration/example Rule-based Spreadsheet-based Template-based Text-based Wizard-based Workflow and dataflow diagrams	Business and data management Education and teaching Games and Entertainment Healthcare and wellness Interaction Design Mobile applications Robotics Smart objects and environments Web applications and mashups	Personal Utilitarian Educational Playful Assistive technology	Generic users Domain experts Students

In order to identify representative studies useful to provide examples during the analysis of results, papers have been also qualitatively assessed. In particular, we classified journal papers according to the quartile (Q1, Q2, Q3, and Q4 quartiles) assigned by Scimago Journal & Country Ranking (SJR) database[6], by choosing the highest quartile assigned to the considered journal in the year of the article among the possible different subject areas. As far as conference papers are concerned, we used the GII-GRIN-SCIE (GGS) Conference Rating: Class 1, Class 2, Class 3, and Work in progress conferences[7]. This is an initiative sponsored by GII (Group of Italian Professors of Computer Engineering), GRIN (Group of Italian Professors of Computer Science), and SCIE (Spanish Computer-Science Society) to develop a unified rating of computer science conferences. The proposed ranking is based on three international and well-known rankings/ratings of computer science conferences: the Australian CORE Conference Rating, Microsoft Academic and LiveSHINE (that is the successor to the SHINE Google-Scholar-Based Conference Ranking). For each paper published on a journal, conference proceedings or another venue, the number of citations on Google Scholar (accessed in the last week of September 2017) was also retrieved.

[6] http://www.scimagojr.com
[7] http://gii-grin-scie-rating.scie.es

At the end, we preferably selected as example papers to be discussed in the following sections those ones with at least 10 citations on Google Scholar, which have been published on Q1/Q2 ranked journals and Class 1/2 conferences. However, in some cases, we also considered papers published recently (in 2016 and 2017) on a high-ranking journal or conference, even though, due to their recent publication date, they had less than 10 citations.

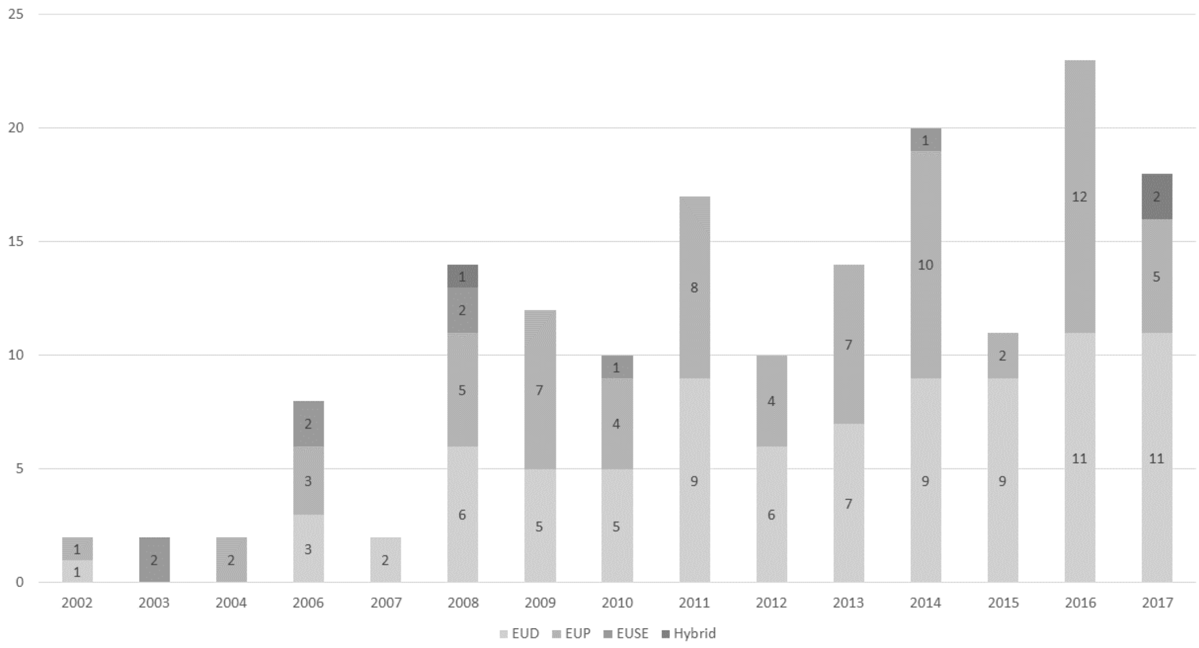

Fig. 2. Number of papers selected in the period 2000-2017.

4. Results

This section discusses in detail the analysis carried out on the 165 selected papers. Fig. 2 shows the distribution of the papers over the time range 2000-2017: as one may observe, no paper has been selected before 2002, and there is a general increase in volume, especially in the last two years (also considering that our analysis ended in May 2017). One may observe that there is a first peak in 2006, which is related to the publication of the Springer book on EUD (see Table 1); then, there are three peaks in correspondence to dedicated events (WEUSE 2008, IS-EUD 2009 and IS-EUD 2011 respectively). The peaks in 2014 and 2016 are neither related to specific conferences nor to special issues.

In the following sub-sections, the results of the analysis conducted according to the seven dimensions are reported.

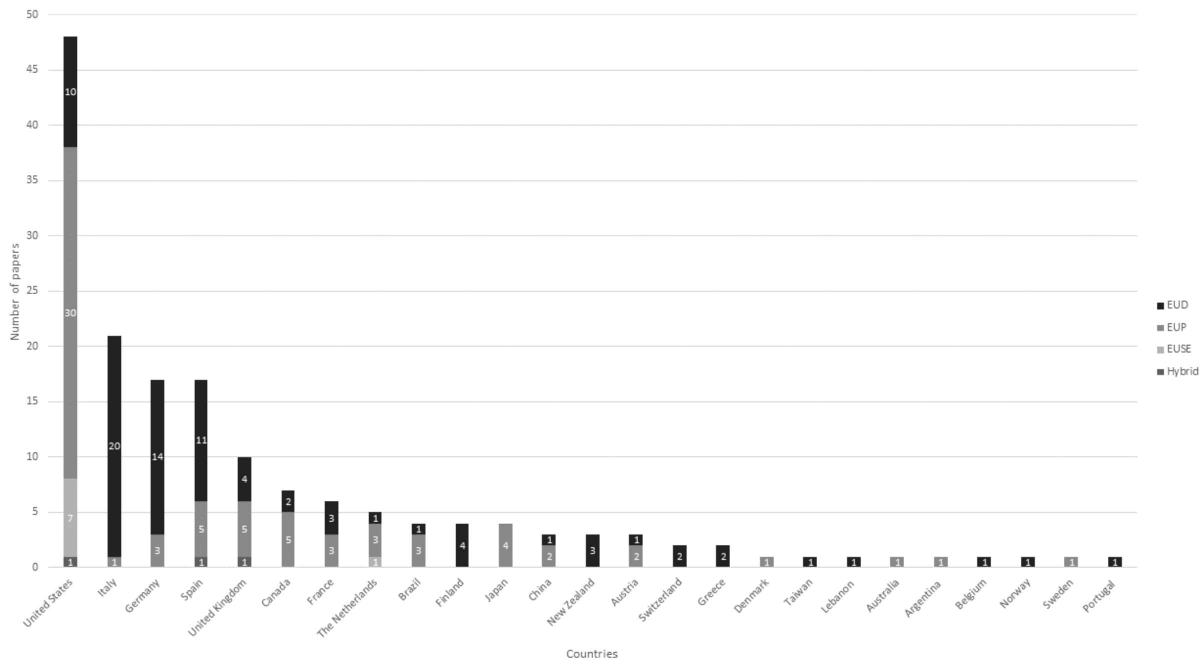

Fig. 3. Number of papers according to the geographical distribution based on the first author's affiliation.

4.1 Types of approaches

As underlined in the Introduction, approaches proposed in this research field are usually classified by their authors as EUD, EUP, or EUSE. We refer to such terms exclusively according to what is reported by the authors of each analyzed article, regardless of the actual meaning or if the term has been used properly or not.

Fig. 2 shows the temporal distribution of papers and their related classification according to the type of approach. As one may observe, EUP and EUD are almost equally distributed with respect to time (except for 2015); interestingly enough, even though EUP was born before EUD, it keeps on being a widely adopted term to denote this kind of support for creation and modification of digital artifacts offered to end users.

To investigate the use of the EUP and EUD terms, we carried out a geographical analysis of selected papers on the basis of the authors' affiliation. **Fig. 3** shows that in US (the country with more selected papers) the EUP term is much more used than EUD, whilst in papers coming from European countries, such as Italy and Germany, EUD is the preferred term. The former result could depend on the fact that US is the country where the research on EUP is born, dating back to the works by Bonnie Nardi (1993) and Alan Cypher (1993). The latter probably derives from the experience gathered in the frame of the European Network of Excellence on End-User

Development (EUD-Net) funded by the European Commission in 2003; among the members of this network, there were different Italian and German research institutions, and their scholars have kept on working on EUD till now. Regardless of terminology preferences, by analyzing the papers' content we discovered that proposals discussing EUP and EUD approaches generally pursue distinct goals, as it will be illustrated next in this section and in the mapping results (Section 4.8).

More precisely, in the analysis of the selected papers according to D1, we identified just 8 papers that propose a pure EUSE approach (around 5% of the papers). This limited number of papers might be due to the fact that almost only the community from the US universities that participated in the EUSES Consortium[8] is actually working on this theme. The main outcomes of this community is reported in the already mentioned survey (Ko et al., 2011). These papers recognize the need to provide end users with better support for software development activities, in order to improve the dependability of their software products. One of the first attempts in this direction is proposed in [a45], where the authors applied to spreadsheets (and to end-user programmers) the method adopted in professional programming of adding assertions in the form of preconditions, post-conditions, and invariants for making explicit the properties expected for a program. The HCI aspects of this method are better explained in [a125], which presents a curiosity-centered approach to eliciting assertions from end users, built on a surprise-explain-reward strategy; in this work, the authors also demonstrate the effectiveness of this approach in encouraging end users to enter assertions that help them find errors. The paper [a144] investigates fault localization methods through an empirical study that examines the impact of two specific factors on the effectiveness of the proposed methods. The obtained results provide suggestions to researchers and practitioners on the design and evaluation of fault localization methods. The paper [a46], instead, presents an empirical study aimed at demonstrating the usefulness of versioning support in web mashups: the authors have integrated the popular mashup environment Yahoo! Pipes with versioning support and carried out experiments that provide evidence on how this advanced feature may help pipe developers create and debug mashups.

In another 42% of the selected papers authors refer to their approach as EUP, usually proposing visual languages suitable to a programming activity carried out by non-professional programmers. For instance, the proposal in [a38] aims to support EUP of robot technologies to be used in therapies for training social skills to autistic children. With the developed tool, therapists with general computer skills may create training or behavioral scenarios by connecting existing behavioral blocks and by typing textual robot commands. Other EUP approaches aim

[8] http://eusesconsortium.org

at making programming activity easier by providing simplified programming languages. For instance, the paper [a50] explores the use of a Domain-Specific Language (DSL), based on JavaScript, to carry out web augmentation, namely layering relevant content, layout or navigation features over a website. The proposed language targets hobby programmers and computer literates by allowing them to easily create scripts in a more domain-oriented and declarative way than using JavaScript. The paper [a82] proposes the use of natural language to support end users in performing tasks – e.g., filtering, reducing, and joining - over tabular spreadsheet data. The key component of this system, implemented as an Excel add-in, is a translation algorithm for converting a natural language specification to a ranked set of programs in a DSL proposed by the authors, which is expressive enough to represent the desired categories of tasks.

In the 51% of papers the authors characterize the approach as EUD. Most of EUD approaches are specifically oriented to empower end users to create new digital artifacts (forms, mashups, simulations, video games, etc.) separated from the EUD environment they are using. For instance, the paper [a49] proposes a mashup development framework characterized by a WYSIWYG (What You See Is What You Get) interface that facilitates data integration and service orchestration, by hiding the technology and implementation complexity of mashups. Marchiori et al. presents a methodology and a EUD tool for game authoring aimed at educators who can thus collaborate in the development process of educational games [a58]. The paper [a27] proposes a framework based on the jigsaw metaphor, named Puzzle, which allows end users to create, modify and execute mobile applications able to interact with smart things and web services.

In principle, approaches can be applied in conjunction with others: for instance, a EUD approach could integrate techniques to guarantee the quality of the created digital artifact and thus implement EUSE techniques. However, among the papers analyzed, only two papers [a91][a108] mention the fact that both EUD and EUP are implemented, whilst only the paper [a154] declared to adopt all the three approaches. Thus, the papers proposing a hybrid approach are only 2% of all papers.

4.2 Types of techniques

To investigate dimension D2, the analysis of papers revealed that several different types of techniques are proposed in the frame of the approaches identified with D1. After various discussions within the research group, we propose here a classification of such techniques.

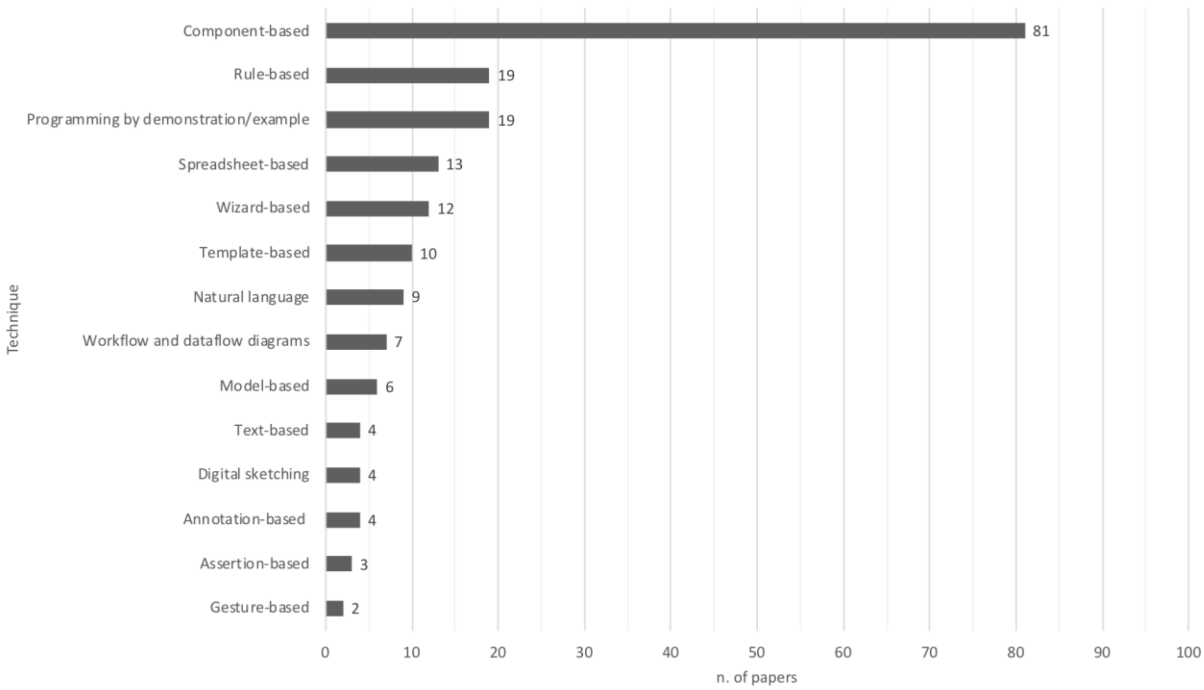

Fig. 4. Classification of techniques proposed in the selected papers (D2).

Fig. 4 illustrates the identified classes of techniques and their frequencies in the selected papers (please note that the total number of techniques found in the papers is more than 165 since some proposals adopt more than one technique). In the following subsections the 14 techniques identified in the selected papers are presented, ordered by their frequencies in the papers.

Component-based

This technique is the most frequent in the selected papers (48%). This is probably because software designers are used to address the design of modern complex systems by following the modularity principle of software engineering, which allows achieving extendibility, reusability and compatibility. Systems are divided into modules based on functionality and developers may use prewritten code and program new functionalities in separate modules. A component is a module with additional restrictions of substitutability using a specific interface. Reasoning in terms of components that can be easily assembled to create a program is thus transferred to end users. This technique encompasses composition of digital artifacts by means of visual programming environments based on domain-specific concepts represented either as 2D or 3D objects. The composition activity is usually performed through direct manipulation and the interaction metaphor may vary from jigsaw puzzle [a20] to boxes-and-wires [a147], from pipes [a17] to 3D blocks [a165]. For instance, the paper by Dörner and colleagues presents a EUD environment that supports business process experts to model and adapt business processes of an Enterprise Resource Planning system [a147]. To this purpose, its graphical user interface employs the box-and-wire interaction metaphor that, by means of visual composition, allows creating visualizations and automations of business processes, and calculations using data from different systems and sources. Another solution that exploits a component-based interaction is that proposed in [a156]: it presents a platform that allows end users to extract contents from heterogeneous sources and compose Personal Information Spaces that satisfy their information needs. The composition paradigm is, on the one hand, suitable to non-technical users for its capability to abstract from technical details, and, on the other hand, amenable to customization with respect to the requirements of specific domains. Similarly, Zhong and Liu present an environment that provides chemistry educators with domain-oriented building blocks, which can be assembled to create 3D virtual chemistry experiments [a165].

Several component-based solutions exploit domain-specific visual languages (DSVLs) suitable to domain experts, who may consider intuitive composing virtual objects that resemble those ones they use in daily activities. To cope with the development of visual design environments that support the implementation of DSVLs, the paper [a153] proposes Pounamu, a meta-tool aimed to support end users to rapidly design, prototype and modify tools supporting a wide range of visual notations.

In recent years, component-based techniques are adopted in most of those environments that empower end users to create mashups, such as [a46][a49][a55][a133] [a151]; mashups are web applications that result from the combination of more than one web services. The availability of several open application programming interfaces and data sources nowadays contribute to the success of mashup technologies.

Rule-based

The rule-based technique is proposed by 19 papers (about 12%) and is mostly aimed to support end users to easily and autonomously personalize the behavior of smart devices, Ambient Intelligence (AmI) systems and Internet of Things (IoT) applications, i.e. artifacts that encompass both hardware and software. Among the selected papers, [a22][a24][a57][a61][a87][a115] present methods and tools that allow end users without programming experience to customize the behavior of devices/environments/applications through the specification of trigger-action rules. For example, Ghiani and colleagues propose a EUD environment that allows end users to create complex trigger-action rules for different types of IoT applications [a22]. The paper [a61] presents an ontology-based framework for the development of context-aware applications, which provides personalized programming support for users with different technical skills; furthermore, several cooperation modalities are enabled, as well as resource sharing and reuse. Houben et al. propose a system for “human-data design” for IoT applications based on rule definition; this system includes cubical interconnected artifacts, called PhysiCubes, to provide physical and embedded ambient data visualizations, and a web-based EUP tool, to add, remove, or change rules that define such visualizations [a87]. Barricelli and Valtolina, instead, propose the SmartFit Rule Editor, a tool specifically designed for coaches and trainers of non-professional teams of athletes to monitor and analyze fitness and wellness data streams; with this tool, end users may create rules that support them in detecting relevant events and performing specific actions [a115].

Programming by demonstration/example

This technique, proposed in 19 papers, is one of the first proposals for enabling end users to program in an easy way: indeed, the user performs actions on concrete examples, the system records these actions and infers a generalized program that can be used on new examples. For instance, the paper [a136] is considered a seminal work in the field of context-aware computing: it presents a context-aware prototyping environment targeted to end users, which allows them to program a desired context-aware behavior in situ, by demonstrating it to the system and by annotating portions of the demonstration. More recently, [a89] presents a new programming-by-demonstration system that allows end users to create automations on their smartphone; the user may give verbal commands and then demonstrate them by directly manipulating Android apps’ user interface. On the basis of the verbal instructions, the demonstrated procedures, and the user interface hierarchical structure of the apps, the system is able to generalize a script than can successively be used with different variations and parameters.

Spreadsheet-based

The spreadsheet-based technique has been proposed in 13 papers (about 8%). Spreadsheets are a common business tool and are considered the most widely used EUP environment (Burnett et al., 2004). For example, considering its ubiquity in business landscape, Saldivar et al. presents a spreadsheet-based technique for business process model analysis [a146]. Here, the problem of business process performance analysis and verification is mapped into the problem of configuring and analyzing data in common spreadsheets; the generation of spreadsheets from business process models is thus supported, as well as the possibility of defining analysis reports and simulations of business process executions. The paper [a74] proposes instead a EUP environment for mashups, which offers, also in this case, a spreadsheet-like programming experience. It is based on an expressive data structure that takes advantage of nested table and mashup operators visualized with contextual menus and formula bars. A spreadsheet tool with features for using and exploring hierarchical data is presented in [a86]. It supports the user in creating different kinds of visualizations by just moving spreadsheet columns through drag-and-drop. It also allows the user to define a variety of data summaries without having to learn new programming-oriented concepts such as SQL queries or pivot tables.

Wizard-based

Wizard-based interaction, found in 12 papers (7%), is usually adopted in those situations where a task can be naturally split into a limited sequence of simple operations, in order to guide the user throughout the overall activity without requiring too much cognitive effort. In EUD, this technique has been adopted for example in the e-government domain to drive civil servants throughout the definition of web pages implementing e-government services for citizens [a39], to allow house inhabitants to customize smart home data visualization through a guided step-by-step creation mechanism [a90], or to support physicians to develop mobile data applications (e.g., for clinical trials or online surveys) in an intuitive way [a41]. Usually, wizard-based interaction is combined with form-based interaction to help end users customize and create digital artifacts by means of fill-in forms, through which they may provide the parameters necessary to the system to automatically generate code or interface customizations.

Template-based

The template-based technique is proposed in 10 papers (6%). Templates are considered an effective way to empower end users to customize application features and tailor the content to their personal interests. For example, the paper [a95] proposes a tool that allows users to tailor information awareness content: layout templates are made available in the tool and the user can create her/his own collages of channels (information services) by simple drag-and-drop, thus associating each template region with a piece of data. When the collage must be displayed, the system inspects the data to identify its type and automatically determines the most suitable rendering method to depict it to the screen. The paper [a104], instead, uses the concept of template to allow users to edit spreadsheets that do not contain some kinds of errors, and proposes an automatic system to infer such templates.

Natural language

The idea of programming in natural language was proposed more than fifty years ago (Sammet, 1966), but only recently significant advances in natural language understanding have been made, see for example IBM Watson, Apple Siri, and Google Assistant. However, giving the possibility to end users to program using natural language remains an open challenge. The selected 9 papers (about 5%) aim at addressing this challenge in specific and narrow domains. For instance, the paper [a48] proposes this technique for debugging the behaviors of intelligent assistants, i.e., software assisting end users with email, shopping, and other tasks. A “Why-oriented” approach is here proposed: users may ask questions about how the assistant made its predictions, and consequently change the answers to their questions, in order to debug the assistant’s current and future predictions. In [a76] the authors describe a novel personal information assistance engine, called Atomate, that supports end users in automating a variety of simple tasks and reminders, by managing several information sources, including the user’s online calendar, web-based e-mail client, news feeds and messaging services. To this end, the system provides end users with a constrained-input natural language interface for behavior specification. The paper [a82] describes a natural language-based interface for spreadsheet programming, which capitalizes on the design of a typed domain-specific language appropriate for end users. Another form of natural language programming is based on *keyword commands*: the system proposed in [a75], for example, allows the user to enter a set of keywords expressing a command in a given domain (e.g. web browsing) and then directly translates such keyword commands into executable code.

Workflow and Dataflow diagrams

Easy-to-use visual programming languages are often based on the idea of representing complex computations (activities) through diagrams that encompass nodes (single computations or activities) and wires to indicate dependencies among computations (activities). This leads to notations called dataflow diagrams or workflow diagrams, according to the emphasis put on data transformation or activity execution.

This technique has been proposed in 7 papers (4%). For example, the paper [a85] presents LondonTube, a visual language for enabling end users to build cloud-mobile-web apps in an intuitive way. As any other dataflow notation, it represents computational logic with graphical icons connected by wires; in addition, a node may represent either a cloud, a mobile or a web device, and thus may contain a nested dataflow diagram, giving rise to a structured dataflow. Catala et al., on the other hand, explores the use of a visual dataflow language for creating trigger-action rules able to control the behavior of smart environments [a109]. The recent paper of Turchi et al.

describes an approach to adapting the user experience with pervasive displays to heterogeneous usage contexts; the approach allows end users to create workflows to satisfy their specific usage needs, by means of tangible interaction and puzzle metaphor [a159].

Model-based

This technique, found in 6 papers, usually comprises a language for describing a model of the system to be created and an automatic code generator taking the model as an input. In this way, the user does not need to learn any programming aspects but may easily express the requirements of the system at a high level of abstraction. For example, [a67] proposes discourse modeling, a way of modeling interaction design as a dialogue between human and computer (the so-called 'discourse'). Some model-based approaches adopt the idea of *design patterns* to drive interaction modeling: for instance, [a9] describes a recommendation tool embedded in a visual environment, which aims at suggesting design patterns to support end users to create their designs; similarly, in [a64], a model-based and pattern-based web application framework is proposed to help educators, teachers and therapists create and customize tangible learning experiences for disabled children.

Text-based

Even though visual programming environments and direct manipulation have demonstrated to be easy to use by end users, some works, like [a130][a133], observe that graphical creation and manipulation of complex formulas or applications can become overly cumbersome. Therefore, they propose hybrid approaches that, beyond employing visual manipulation, also encompass textual representation of formulas, markups and scripts. Only 4 papers present this kind of technique.

Digital sketching

This technique is based on digital painting for creating interface layouts and behaviors, which are then processed by automatic systems able to generate the corresponding code. It has been proposed in 4 papers. The paper [a101] employs this technique to enable game developers to create interactive computer-controlled characters by digitally painting storyboards; an algorithm based on machine learning then analyzes each storyboard and generates a behavior suitable to the situations provided in the storyboard. This technique can be combined with gesture-based [a113] (see below) or rule-based development techniques [a155]. Obrenovic and Martens propose instead to extend the concept of sketching to the more generic concept of manipulation of interactive materials, which include any piece of software or hardware that can take part of the interactive user experience, such as input from sensors, output in different forms (sound, video, or image), or interaction with web services [a47]. The proposed tool provides designers with the possibility of combining elements of traditional pen-and-paper sketching with end-user programming tools, such as spreadsheets and scripting.

Annotation-based

This technique, found in 4 papers, is based on the idea of empowering the user to create new artifacts (e.g., new web pages) by simply annotating (or similarly augmenting) one or more existing artifacts. As an example, the paper by Firmenich et al. [a150] describes a way for creating new collaborative procedures, such as planning a trip over the Web, by means of augmentation tools that allow extending the set of elementary tasks users can do while navigating the Web. The approach is based on the use of so-called "augmenters", able to automatic extract information and Document Object Model (DOM) elements from different websites for creating new distributed user interfaces.

Assertion-based

Similarly to its use in software engineering, the assertion-based technique is proposed to support error prevention [a73][a125] and debugging [a45] (3 papers). The focus is mainly on the quality of spreadsheets and the approach appears suitable to end users, who can easily write assertions to reason about their spreadsheets, by improving correctness and efficiency [a45].

Gesture-based

A few scholars have recently explored the use of physical gestures as a novel paradigm to compose digital artifacts in specific domains, such as to support video game creation [a113] and music composition [a120].

4.3 Phases

To explore dimension D3 we analyzed the development activities presented in the papers with reference to the time in which such activities are carried out. Specifically, we classify as "design phase" all those situations where the end user is called on to design and develop a new digital artifact, and which can therefore be used also by other users; the term "use phase" is instead adopted whenever the user carries out a system customization or develops an extension of the system, by creating new functionalities (e.g., by writing macros in a spreadsheet) or new contents at use time.

The results for D3 analysis reveal that development activities discussed in the selected papers are carried out more often in the design phase (57%) than in the use phase (41%), whilst only 4 works (2%) propose approaches for both phases.

As to the design phase, there are for example proposals that aim at facilitating the creation of mobile applications [a27][a68], video games [a28][a58], mashups [a46][a49][a70][a74] and web applications [a135][a148][a149], personal information spaces [a156], assistive technologies [a107], rehabilitation exercises [a103], e-government services [a39], and chemistry experiments [a165].

The adoption of tools at use time usually enables debugging activities (of spreadsheets, macros, and other types of user-generated code) as shown in [a45][a73][a104][a125][a144], system customization and tailoring [a95], or business process management in an easier way [a82][a97]; in addition, the most recent approaches promote the use of development features in the use phase for personalizing and controlling the behavior of smart spaces and smart devices (see for instance, [a23][a92][a129][a141][a142][a143]).

The results of this analysis show that there are two distinct needs in this field: 1) creating new digital artifacts, possibly capable of evolving at use time in the end users' hands; 2) tailoring and customizing one's own application, smart environment or object. These two different needs should be considered and analyzed since the beginning of a project, in order to apply proper design methodologies.

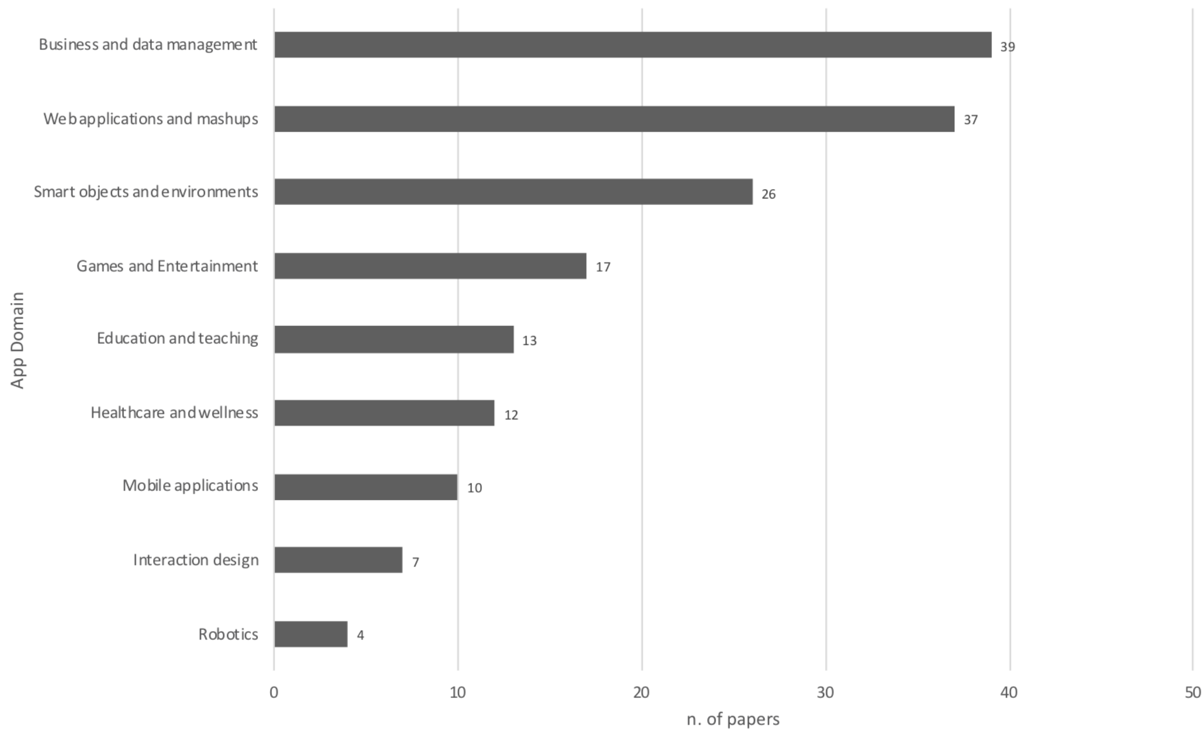

Fig. 5. Application domains identified in the select papers (D4).

4.4 Application domains

Dimension D4 is useful to discover the application domains considered by the publications analyzed in this work. The results suggest that they are highly variable, even though three application domains are the most frequent (see **Fig. 5**): *business and data management* (24%), *web applications and mashups* (23%), and *smart objects* and *smart environments* (16%). The first one is the historical domain where the idea of shaping digital artifacts by end users at run time was born; the other two domains are gaining momentum since users are more and more interested in contributing to the Web (also by creating their web applications) and in defining the behavior of personal devices and environments.

As it will be better noticed in mapping results presented in Section 4.8, *business and data management* is an important application domain for spreadsheet programming; for instance, the paper [a97] presents an approach to the extraction of information from spreadsheets, in order to present it with leveled dataflow diagrams, suitable to users in industrial settings. It is also worth noticing that most of the research in this application domain concentrates on spreadsheet debugging [a36][a45][a99][a125][a144].

The most cited papers in the second application domain are those that propose approaches for facilitating mashup creation by end users. As an example, the paper by Wang et al. [a74] presents Mashroom, a mashup tool that offers a spreadsheet-like programming experience by adopting the nested table as the data structure and a set of visual mashup operators; the authors demonstrate through different case studies that Mashroom provides end users with an effective and efficient way to build the common mashups.

Ambient Intelligence, IoT and any kind of "smart" application may require the intervention of end users for configuration and solution customization over time; therefore, several works published in 2016 and 2017 propose for example tools to be applied in the case of *smart objects and environments* (e.g. the smart home) [a20][a21][a22][a87][a90][a141][a142][a143].

Other application domains that received attention in our selected papers are *games* and *entertainment* (10%), *education and teaching* (8%), *healthcare and wellness* (7%), *mobile applications* (6%), *interaction design* (4%), and *robotics* (2%). As an example, cultural heritage is a field that we included in the *games and entertainment* category: here, museum and exhibit curators often would like to create virtual guides or related websites, as discussed in [a91][a106]. The adoption of techniques for system shaping is advocated for instance in [a59] to support teaching cognitive modeling, or to allow creating educational games in an easy way, see for example [a58]. As far as *healthcare and wellness*, we may cite here the variety of proposals in the field of rehabilitation and assistive technologies for disabled and elderly people. As an example, the paper by Carmien and Fischer describes the Memory Aiding Prompting System (MAPS), an environment in which caregivers can create scripts for people with cognitive disabilities to support them in carrying out daily tasks [a107]. Similarly, Tetteroo et al. propose TagTrainer, a physical rehabilitation technology that supports physiotherapists in the creation of customized rehabilitation exercises for people with neurological impairments [a103]. *Mobile applications* represent another important domain addressed by some recent papers, such as [a41] that proposes a model-driven framework to empower domain experts, such as physicians and psychologists, in creating mobile data collection applications in an intuitive way, or [a158], which describes an iconic language, called MicroApp, for specifying pervasive mobile applications directly on the mobile device. *Interaction design* is often in the hand of designers that are not experienced programmers; for this reason, the authors of Sketchify have connected their tool for sketching interactive systems with spreadsheets and scripting languages to allow designers to quickly outline sketch behaviors [a47]. Díaz et al. created a recommendation tool embedded in a visual environment that suggests novice designers which design patterns should be used in their development project [a9]. Similarly, as an example of works in the *robotics* application domain, [a88] presents Code3, a system that enables non-roboticist programmers to program a mobile manipulator to perform complex tasks.

We can finally observe that the classification of the papers according to this dimension is strictly related to the type of outcome of the EUD/EUP/EUSE activities: indeed, a good percentage of works (24%) propose solutions that deal with hardware artifacts, such as smart objects, robots, healthcare devices or physical games. In these papers, the user activity mainly consists in using a software artifact to change the behavior of a physical artifact; for this reason, we have considered this activity as one aimed at shaping hardware artifacts. Interestingly enough, there are also a few solutions, such as GALLAG Strip [a80] or TAPAS [a159], which propose tangible programming, that is, the adoption of physical manipulation, to define the behavior of software or hardware artifacts.

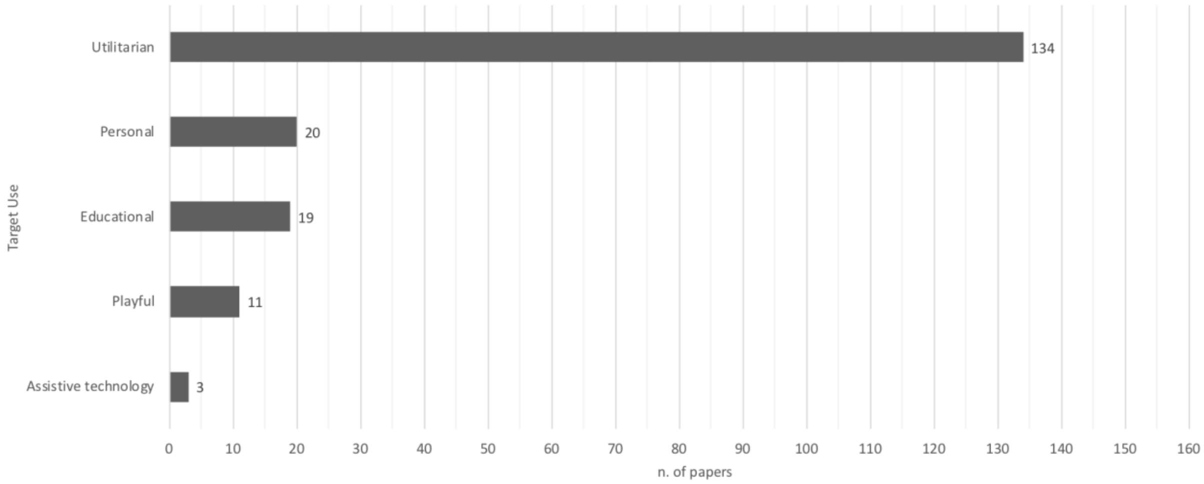

Fig. 6. Classification of the selected papers according to the target use (D5).

4.5 Target use

The analysis of papers with respect to D5 reveals that target use of the proposed approaches is mainly *utilitarian*, with 134 papers, as shown in Fig. 6. This highlights the importance, for end users, of creating customized systems or being able to adapt them, in order to cope with their daily work practice.

More precisely, 117 of them describe tools and methods to support different kinds of domain experts in customizing and/or creating digital artifacts used in their workplace. Other 14 papers propose approaches that could be used both for *utilitarian* and *personal* objectives (where 'personal objective' is intended for all those situations in which users perform activities for personal reasons and not for work); these papers are concerned with smart environment control (e.g., [a44][a53][a109]), mashup creation (e.g. [a49][a55][a70][a111]), and personal task management (e.g. [a84][a94]). Moreover, there are other 3 papers whose target use might be both *utilitarian* and *playful*. As an example, Smith and Graham present Raptor, a tool for sketching games through tabletop and mixed-reality interaction; they show how this advanced interaction mode effectively supports the ideation phase and the collaboration among designers and testers [a113].

Personal use only is envisaged in 3 papers (out of the 20 reported in Fig. 6). They discuss how to create smartphone automations [a72], Personal Information Management spaces [a76] or how to enable system customization [a95]; whilst, 3 papers consider both a *personal* and *playful* use. For instance, in the last category we found [a80], which describes GALLAG Strip, a mobile and tangible tool that allows users to create context-aware applications through programming by physical demonstration of envisioned interactions with sensors and objects; the aim of this work is to demonstrate how EUP tools may empower end users to create applications tailored to their own needs and lives, for breaking bad habits by behavior change; a tangible and playful interaction is here advocated to better engage end users. These papers bring to 20 the total number of papers in the "personal" class (including the above mentioned 14 papers that propose approaches for both *utilitarian* and *personal* target use).

Seventeen papers deal with pure *educational* objectives. Two other papers have *educational* target use combined with *playful* [a87] and *assistive technology* purposes [a64], thus bringing to 19 the total number in Fig. 6. Interestingly enough, the *educational* target use is drawing attention in recent years, probably because teachers are more and more called on to adopt novel ways of teaching to engage and motivate students in learning activities: 13 out of the 19 papers have been published in the last five years. For instance [a58] presents WEEV (Writing Environment for Educational Video games), a methodology for educational adventure game authoring; in particular, through a narrative metaphor and a novel visual language to represent the flow of the story, WEEV allows educators to collaborate in the educational game development process. Similarly, the paper by Zhong and Liu proposes a domain-oriented end-user design environment, called iVirtualWorld, to support educators to create 3D virtual chemistry experiments, by simply assembling three-dimensional building blocks [a165]. In [a64], Garzotto and Gonella describe a EUD environment that supports the creation and customization of tangible learning experiences for disabled children learning.

Empowering end users in the design of video games or other *playful* applications is proposed in 4 selected papers [a28][a32][a79][a112], which, together with the other ones addressing a mixture of target use, bring to 11 the total number of papers pursuing *playful* objectives (see Fig. 6). For instance, [a28] describes AgentCubes, a

3D-game authoring environment for middle school students, through which the pedagogical focus can be shifted from programming (far from trivial for this kind of users) to design.

Finally, 3 papers can be classified in the *assistive technology* target use: the already mentioned [a64], and papers [a35] and [a38].

4.6 Classes of users

Dimension D6 aims to discover which are the classes of users to whom the approaches and techniques previously discussed are addressed. In particular, we classified users in *generic users*, *domain experts* and *students*.

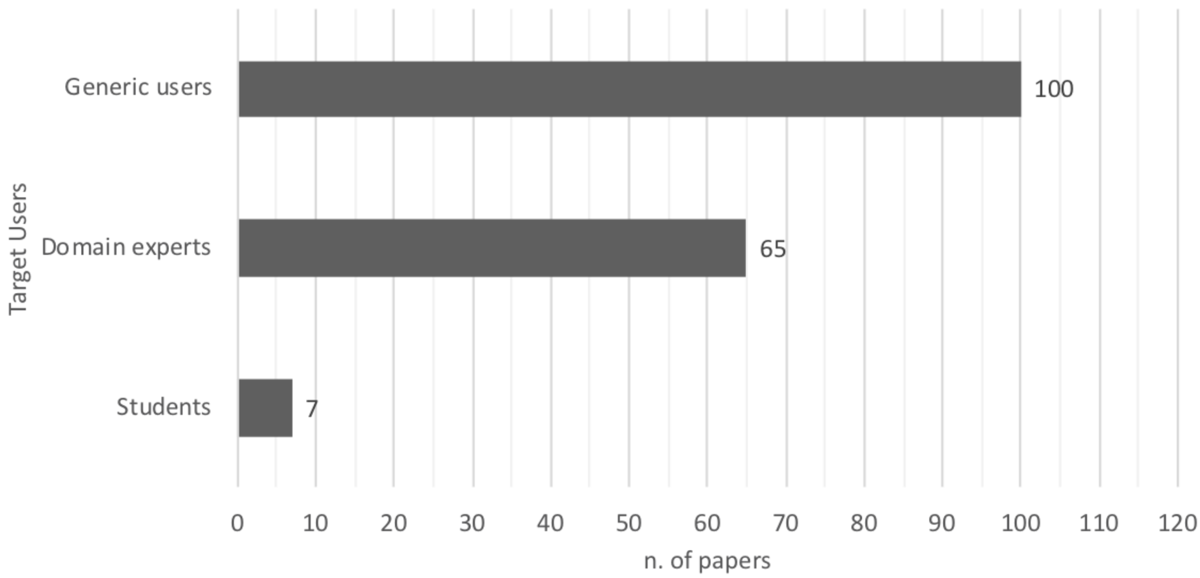

Fig. 7. Classes of users considered in the selected papers (D6).

Fig. 7 shows the distribution of user classes in the selected papers. As in the other cases, this classification makes the total number of papers to be more than 165, since some proposals address more than one class of users.

The results reveal that there are 100 papers that aims at supporting *generic users* in digital artifact creation or tailoring. Among them, there are 6 papers that describe systems providing different interaction levels, by allowing also *domain experts* to perform advanced development activities. Papers that discuss solutions for generic users usually are more focused on demonstrating the feasibility of a EUD/EUP/EUSE technique, rather than on its immediate adoption in a specific domain.

As a whole, the needs of *domain experts* (office workers, managers, civil servants, architects, therapists, physicians, tourism operators, teachers, interaction designers, researchers, and so on) are addressed in 64 papers. As an example, [a38] proposes an architecture for EUP of robots to be used for therapies involving autistic children; in this case, end users are therapists with general computer skills, who could create simple training or general behavioral scenarios, by connecting existing behavioral blocks and by typing textual robot commands. Business analysts are instead the target users of the tools described in [a82] and [a97]; they extend spreadsheets' features by means of a natural language-based interface and data extraction for dataflow diagram visualization respectively. The paper [a115] presents a system for coaches and trainers of non-professional teams of athletes supporting them in monitoring and analyzing IoT data streams and detecting relevant events that require specific actions; EUD features are offered in this case to both IoT engineers and coaches and trainers. Interaction designers may be called on to contribute to the design of new digital artifacts (especially, video games [a47][a113]), without possessing competencies in programming; while the activity of researchers may be supported as well, for example to develop empirical studies on the use of sensor-equipped mobile devices that gather and process data concerning people's everyday lifestyles [a14] or to model scenarios on driving simulators [a105].

Finally, 7 papers consider *students* as end users: they mainly describe tools and techniques aimed at teaching programming and computational thinking (see for instance [a78][a81]).

4.7 Empirical validation

With the final research dimension (D7) we would like to investigate which kinds of empirical validation of the proposed methods or tools have been adopted in the selected papers. Please notice that one of the exclusion criteria was concerned with the absence of empirical validation. From the results of the analysis, we discovered that the majority of works (135, namely 82%) presents a formal experiment with real users. The average number of users that participated in formal experiments is equal to 30, even though a few studies involved a much higher number of participants, especially when students are the target users. For example, 300 students participated in the study of Collins et al. [a59], which investigated the suitability of using visual languages for teaching cognitive modelling; while, the paper [a124] presents a study on a set of success factors (usability of the development experience, user satisfaction and benefits for stakeholders) characterizing a hypermedia development tool, which involved 1000 students and 100 teachers through user testing, contextual inquiry and questionnaires. A high number of users is involved also in the recent papers of Lizcano et al. [a29][a138] (180 and 210 respectively), both addressing the need of generic users to easily create web-based composite applications (mashups). In the first study, 6 groups of 30 participants each were formed and a comparison between the author's visual mashup composition framework and other existing tools was performed, according to a between-subjects protocol. Similarly, in the second study, the sample of 210 participants was divided into 7 groups, and each group used a tool (including the author's one in its evolved version) for creating 7 composite web applications of increasing complexity; completion time and bugs were statistically compared.

Furthermore, 26 papers (16%) demonstrate the validity of their proposal through an informal experimentation, for instance by means of informal studies with few users [a68][a70][a95], workshops [a64][a67][a101], user simulation [a99], case studies [a47]. The remaining 4 papers [a8][a43][a69][a79] (3%) present a pilot study.

4.8 Mapping results

The results related to the seven dimensions of our research question have been combined in order to provide some additional insights about the EUD, EUP and EUSE fields. The objective was obtaining more information about how the results are related one another, and possibly identify research gaps. In the following only the most interesting mapping results are reported and discussed.

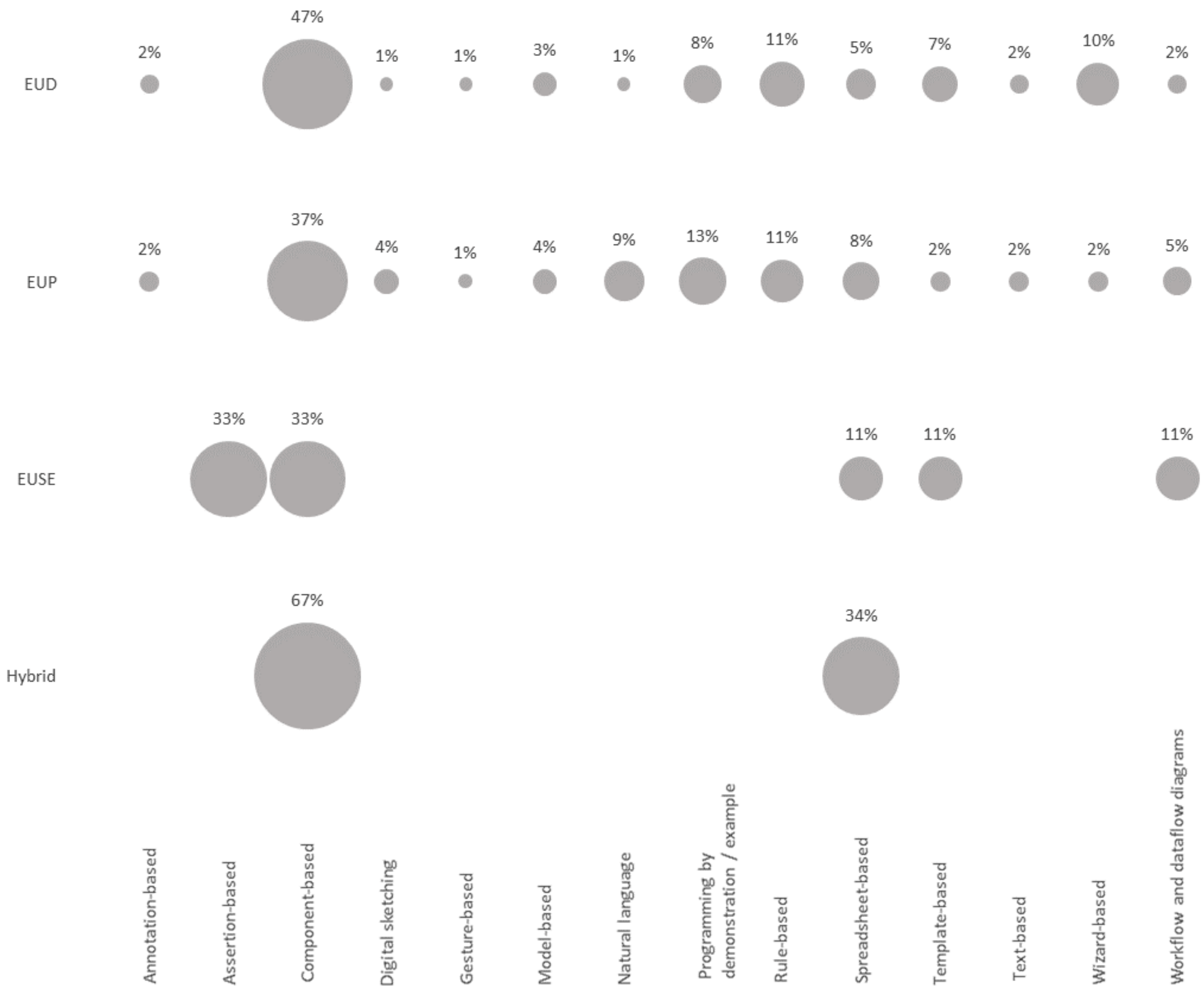

Fig. 8. Mapping D2 (techniques) to D1 (approaches).

Fig. 8 shows the result obtained by mapping the *type of technique* dimension (D2) to the *type of approach* dimension (D1) to investigate which are the most frequent types of techniques used in the three approaches (EUD, EUP, EUSE) or in a combination of them (that we called "hybrid"). The *component-based* technique appears to be the most adopted for both EUD (45 papers) and EUP (31 papers). The second most-used technique is *programming by demonstration/example* for EUP, and *rule-based* and *wizard-based* for EUD. Interestingly enough, *natural language* has been employed in eight EUP papers and only in one EUD paper; all these papers, except one, were published after 2010. This might be because the maturity of the technology has for a long time not been sufficient for practical use. Natural language could however represent a promising technique for the future, given its suitability to users' habits and skills. Further investigation is also necessary as far as novel techniques are concerned, such *as gesture-based* and *digital sketching*, whose frequency is pretty low in all approaches.

For EUSE, the type of approach with the smallest representation in this study, the most used techniques are *assertion-based* and *component-based* (3 papers each). The former is a technique aimed at helping end users avoid errors during programming; for this reason, it is a typical EUSE technique and has not been found in any EUD or EUP solution.

Going more in depth in the comparison between EUD and EUP, it is possible to observe that *template-based* and *wizard-based* are more used in EUD than in EUP. They include all those solutions where end users are supported in designing new artifacts by filling in a template with specific contents, or by a step-by-step interaction, respectively. On the other hand, *natural language* is more used in EUP than in EUD: this technique usually allows making programming easier by enabling end users to express their programs through simplified languages. From this analysis, it is possible to derive that techniques adopted in EUD are more oriented to empower end users to design or customize their artifacts, whilst in EUP the focus seems more on facilitating coding.

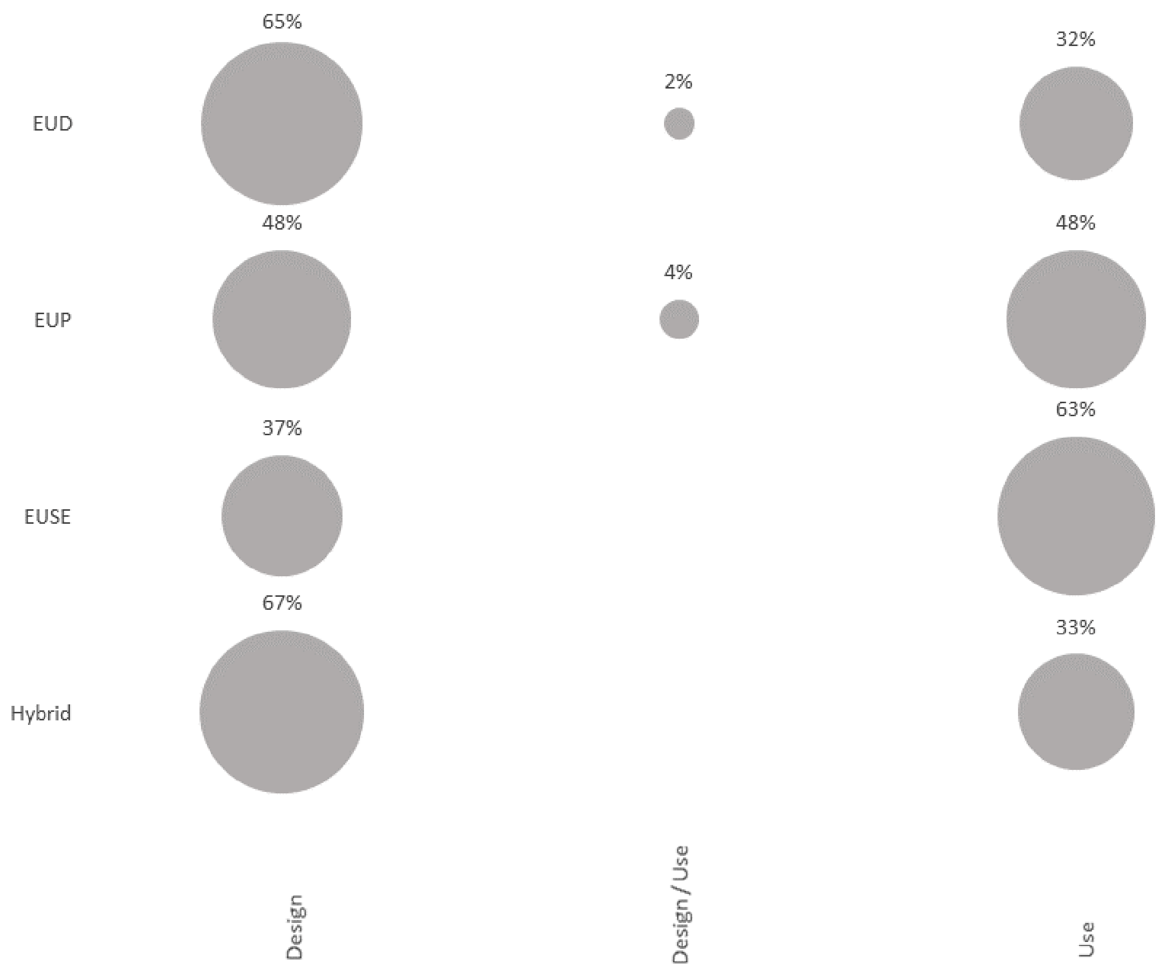

Fig. 9. Mapping D3 (phases) to D1 (approaches).

Fig. 9 shows the results obtained by mapping the *phase* dimension (D3) to the *type of approach* dimension (D1). As one may observe, the percentage of papers that discuss EUD solutions for the *design* phase is more than twice the percentage of papers that propose solutions for the *use* phase; papers adopting EUP approaches focus almost equally on the *design* and *use* phases; finally, papers adopting EUSE approaches propose more solutions concerning the *use* phase. Only three papers propose mixed solutions (called "Hybrid" in the figure), thus no significant observations could be inferred.

In the *design* phase, end users create some new artifact for themselves or for users belonging to a different community; whilst in the *use* phase, end users modify a digital artifact to cope with emerging needs. From this mapping, we may derive that researchers in EUD are more interested in studying solutions that enable end users to design new digital artifacts suitable to their community's work practice and objectives, rather than to modify systems created by others.

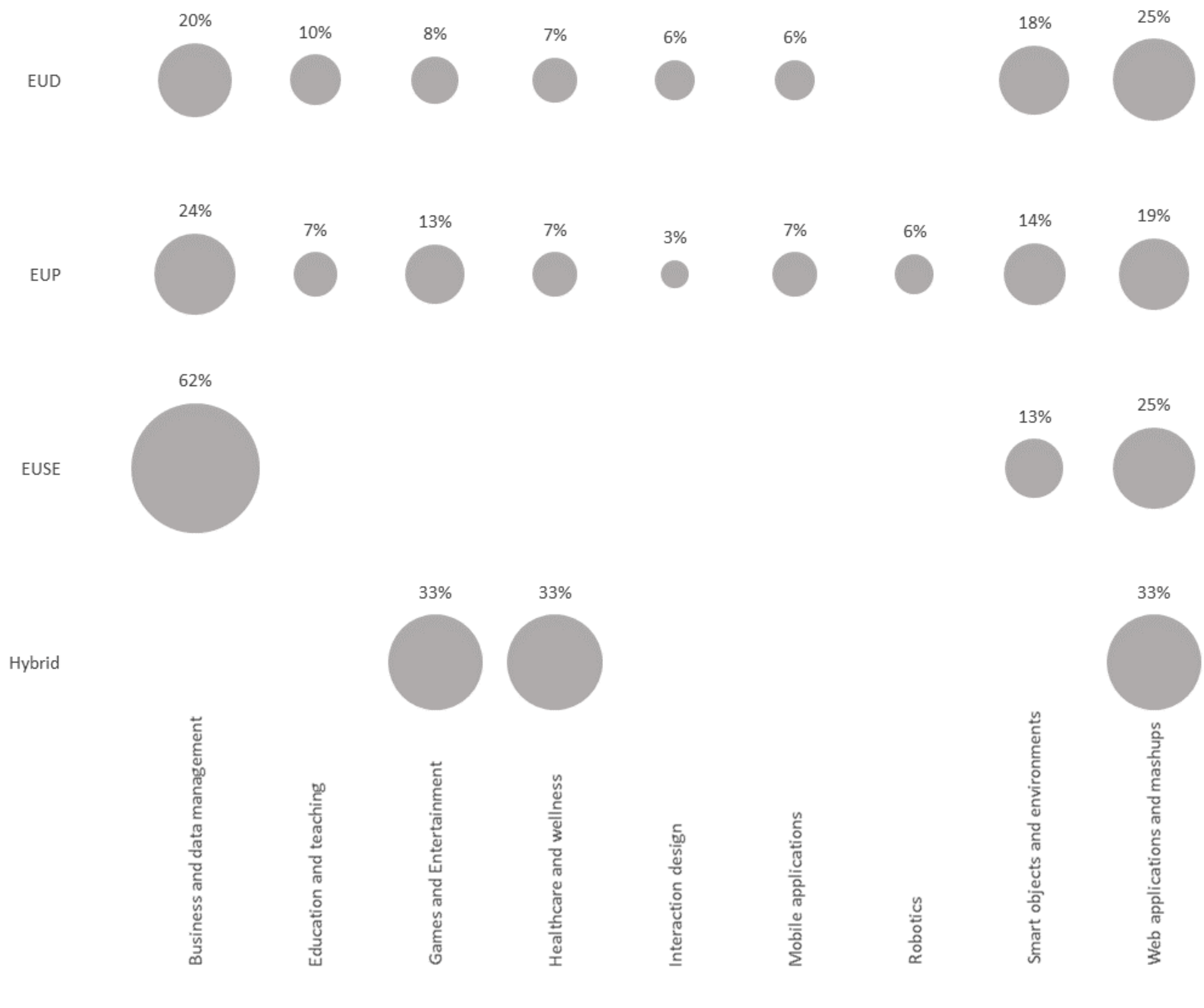

Fig. 10. Mapping D4 (application domains) to D1 (approaches).

Fig. 10 shows the results obtained by mapping the *application domain* dimension (D4) to the *type of approach* dimension (D1). As one may observe, papers proposing the EUD approach mostly address the domain of *web application and mashups* (21 papers), followed by the *business and data management* (17 papers), and *smart objects and environments* (15 papers) domains. The EUP and EUSE approaches mostly address the *business and data management* domain (17 and 5 papers respectively) followed by the *web applications and mashups* and *smart objects and environments* domains. These results suggest that in all the three approaches, the application domains where end users are mostly called on or would like to shape digital artifacts are the three mentioned above. This is an interesting similarity found among EUD, EUP and EUSE that could suggest the need for a joint research effort on these themes.

Furthermore, the majority of the few papers dealing with EUSE focus on the *business and data management* domain, because in this domain the quality of end-user products must be guaranteed; however, this also suggests that there is room for carrying out research on EUSE by applying its methods to other domains where system quality is fundamental, such as *healthcare and wellness* or *robotics*.

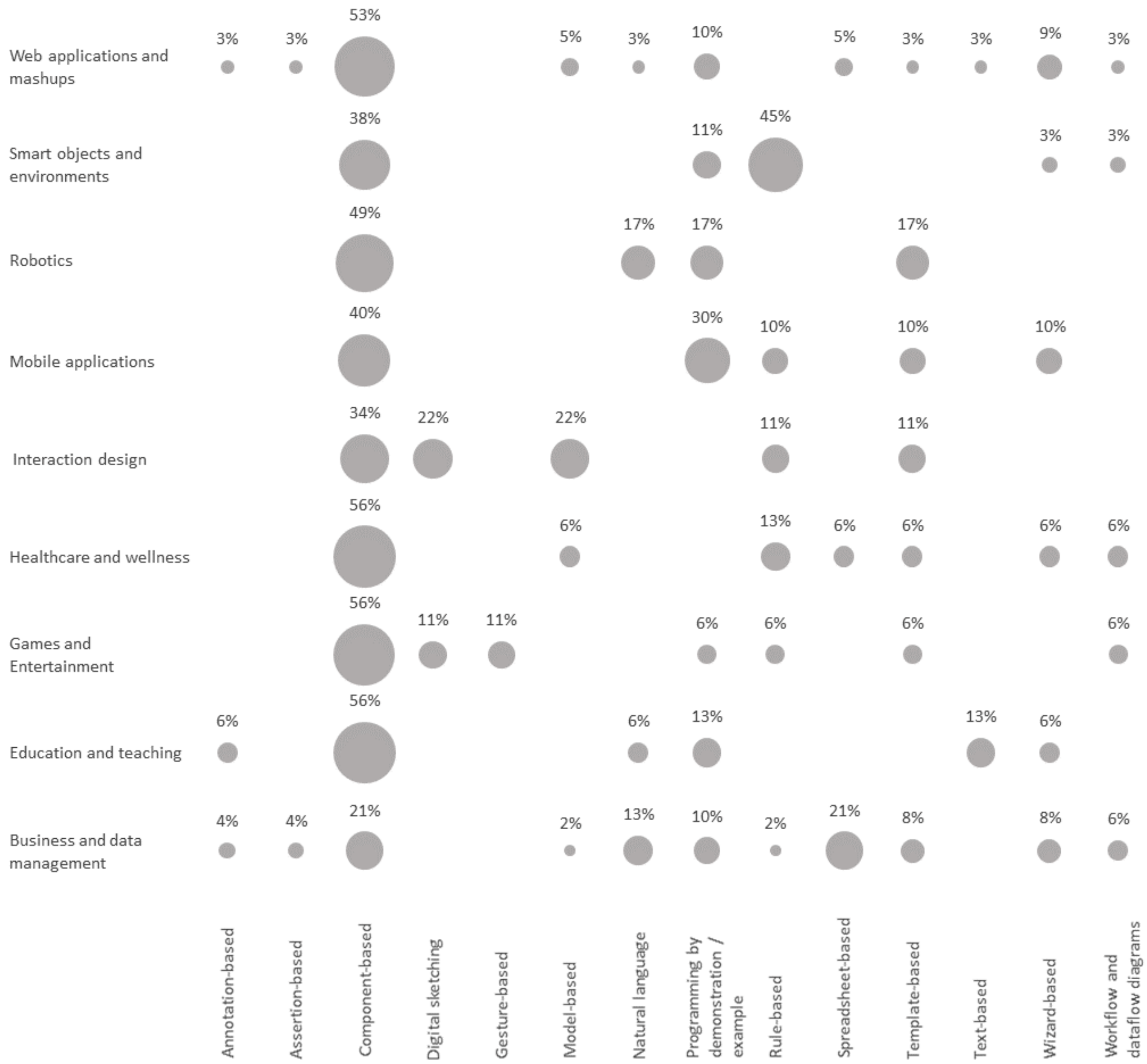

Fig. 11. Mapping D2 (techniques) to D4 (application domains).

In Fig. 11 the reader can find the results of mapping the *type of technique* dimension (D2) to the *application domain* dimension (D4). Here, it is interesting to observe that the *component-based* technique is the preferred technique in almost all domains, thus demonstrating its versatility. However, in the domain of *smart objects and environments*, the most applied technique is the *rule-based* one (13 papers out of 19): indeed, in this domain, the proposed solutions usually allow end users to perform trigger-action programming by creating rules. As expected, the *spreadsheet-based* technique has a high frequency in the *business and data management* domain (10 papers out of 13) where workers are used to deal with spreadsheets; whilst, somewhat surprisingly, in this domain, the *natural language* technique has been applied in 6 papers out of the 9 papers we found that propose this technique. In the *mobile applications* domain, *programming by demonstration/example* is probably more appropriate than other techniques, due to the peculiarities of the device (small screen, gesture interaction, etc.).

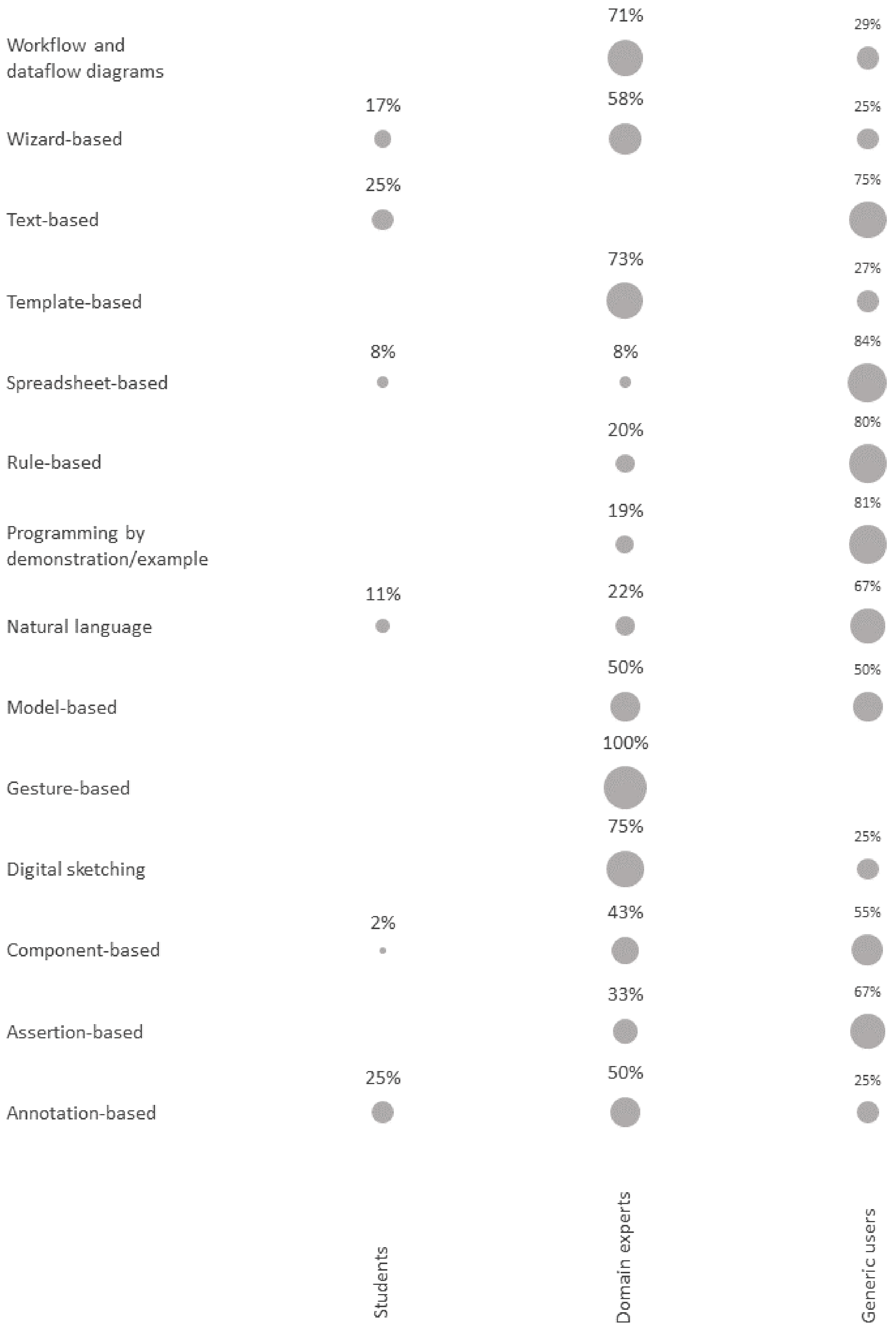

Fig. 12. Mapping D6 (users) to D2 (techniques).

Fig. 12 shows the results obtained by mapping the *class of users* dimension (D6) to the *type of technique* dimension (D2). Interestingly enough, *programming by demonstration/example, rule-based,* and *spreadsheet-based* seem considered more suitable to *generic users* than to *domain experts*. Instead, *gesture-based, digital sketching, template-based,* and *workflow/dataflow diagrams* have been mainly proposed for *domain experts*. This seems reasonable because the former techniques are more general and known for a long time, whist the latter usually require some specific training and might be based on knowledge about specific notations or habits of the considered domain.

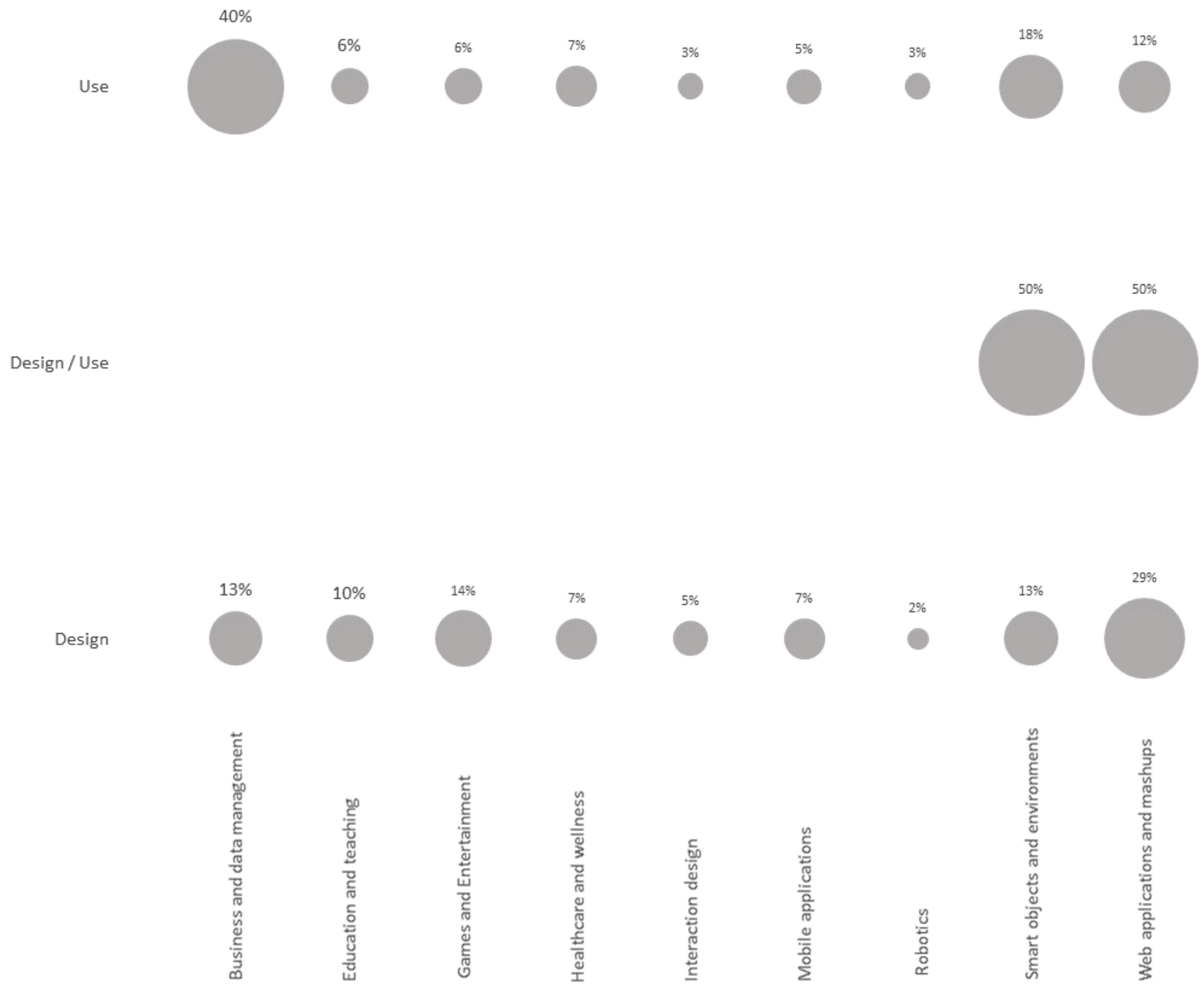

Fig. 13. Mapping D4 (application domains) to D3 (phases).

Results obtained from mapping the *application domain* dimension (D4) to the *phase* dimension (D3) (see Fig. 13) reveals that the most considered application domain in the use phase is *business and data management* (27 papers). Indeed, in this domain the need is mainly to allow end users to customize business and data processing tools, rather than create new tools from scratch. Instead, two application domains, namely *web applications and* *mashups*, and *games and entertainment*, are more addressed in the *design phase* (27 and 13 papers respectively) than in the *use phase*, thus highlighting that several tools have been proposed to support end users to create new artifacts in these domains.

Both phases are addressed in only 4 papers equally split in two domains (*smart objects and environments* and *web applications and mashups*). This suggests that, in these domains, end users have both the need to customize available digital artifacts and to create new ones. For example, in the *smart objects and environments* domain, an end user may need not only to create trigger-action rules for smart home control, but also to develop the behavior of an Arduino-based smart device. However, this limited number of papers (only 4) and domains (only 2) highlights an important gap in the research, which mainly addresses only one phase, instead of considering users' needs of participation in system shaping at both design and use times.

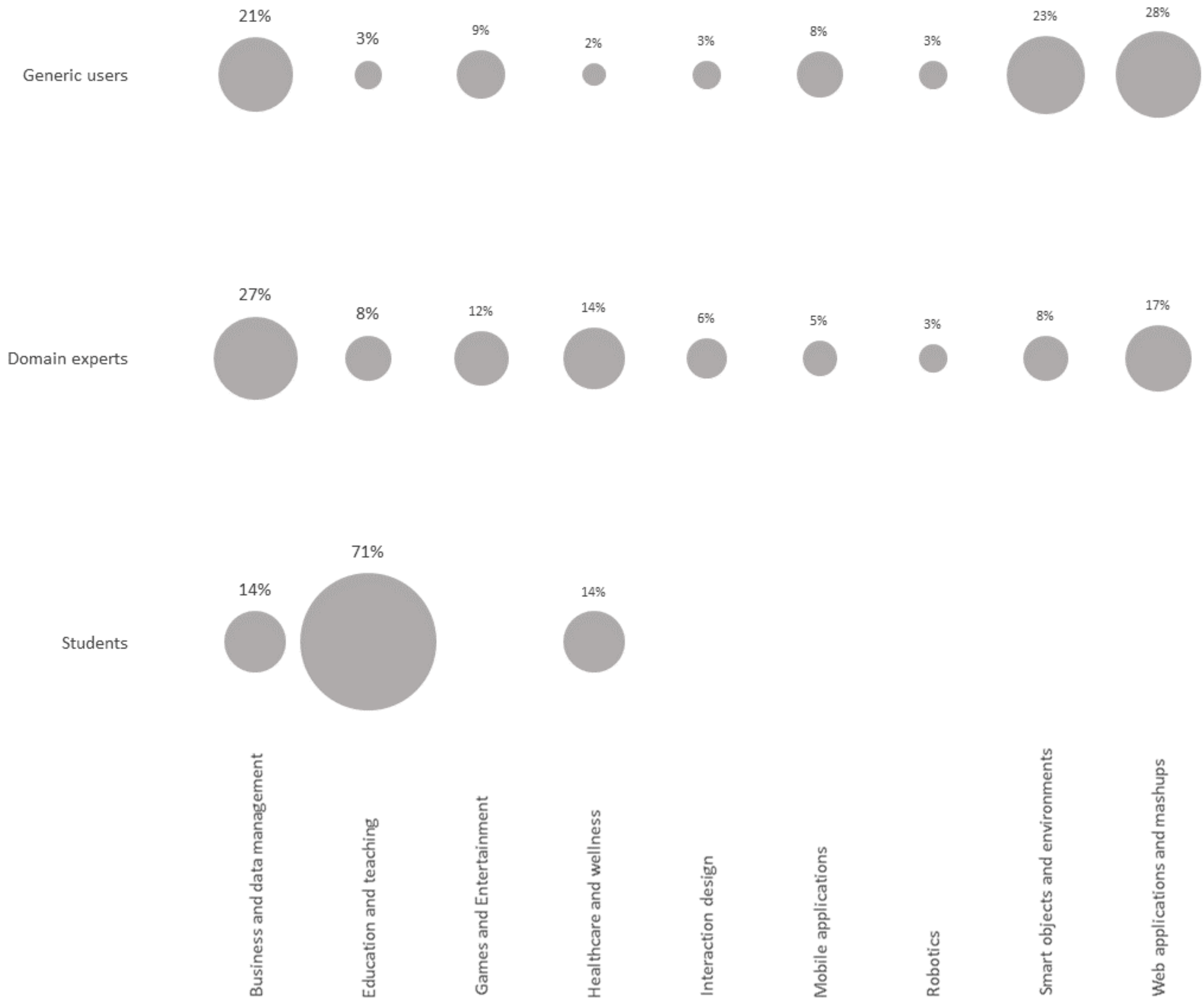

Fig. 14. Mapping D4 (application domains) to D6 (classes of users).

If we analyze results obtained from mapping the *application domain* dimension (D4) to the *class of users* dimension (D6) (see Fig. 14), we can observe that *generic users* are mostly involved in the development and adaptation of *web applications and mashups* (28 papers), *smart objects and environments* (23 papers), and *business and data management* (21 papers). Instead, the needs of *domain experts* are considered also in other domains, beyond *business and data management* and *web applications and mashups:* they are *healthcare and wellness* and *games and entertainment*; this denotes the need of empowering different kinds of domain experts with methods and tools for system shaping, possibly capitalizing on the notations that characterize such specific domains.

As to the *students*, the most considered application domain is obviously *education and teaching*, even though some papers address the need of training students in specific domains, such as *business and data management* and *healthcare and wellness*.

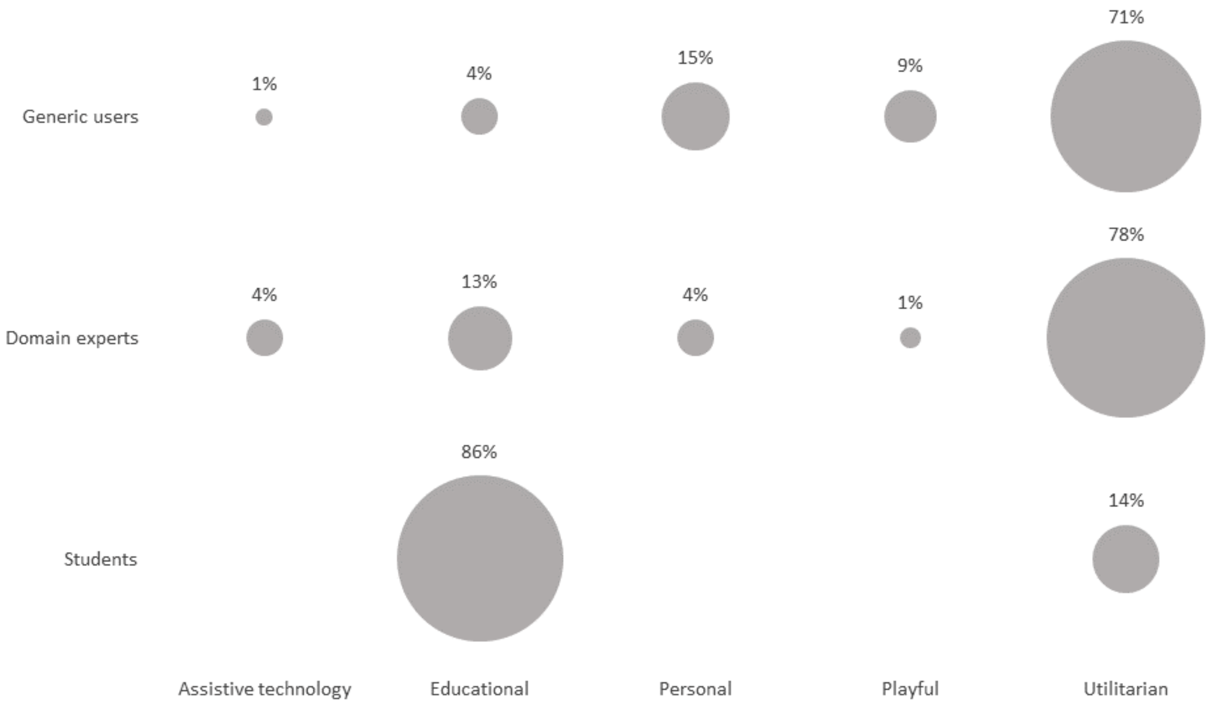

Fig. 15. Mapping D5 (target uses) to D6 (classes of users).

Fig. 15 shows the results obtained mapping the *target use* dimension (D5) to the *class of users* dimension (D6). Both *generic users* and *domain experts* are mostly involved in solutions dealing with the *utilitarian* target use (84 papers for *generic users* and 55 papers for *domain experts*), whilst students' needs are obviously concerned with educational purposes in most cases.

The high percentage of the *utilitarian* target use related to *generic users* suggests that the proposed solutions for enabling end users to craft digital artifacts are often designed independently of the specific domain, work practice or users' capabilities. On one hand, this might be due to the generality of the artifacts considered, like web applications (including mashups) or mobile applications; on the other hand, it might be that researchers are more interested in demonstrating the technical feasibility of a EUD/EUP/EUSE solution rather than in designing something useful for a specific user community. This highlights an interesting issue to be further investigated.

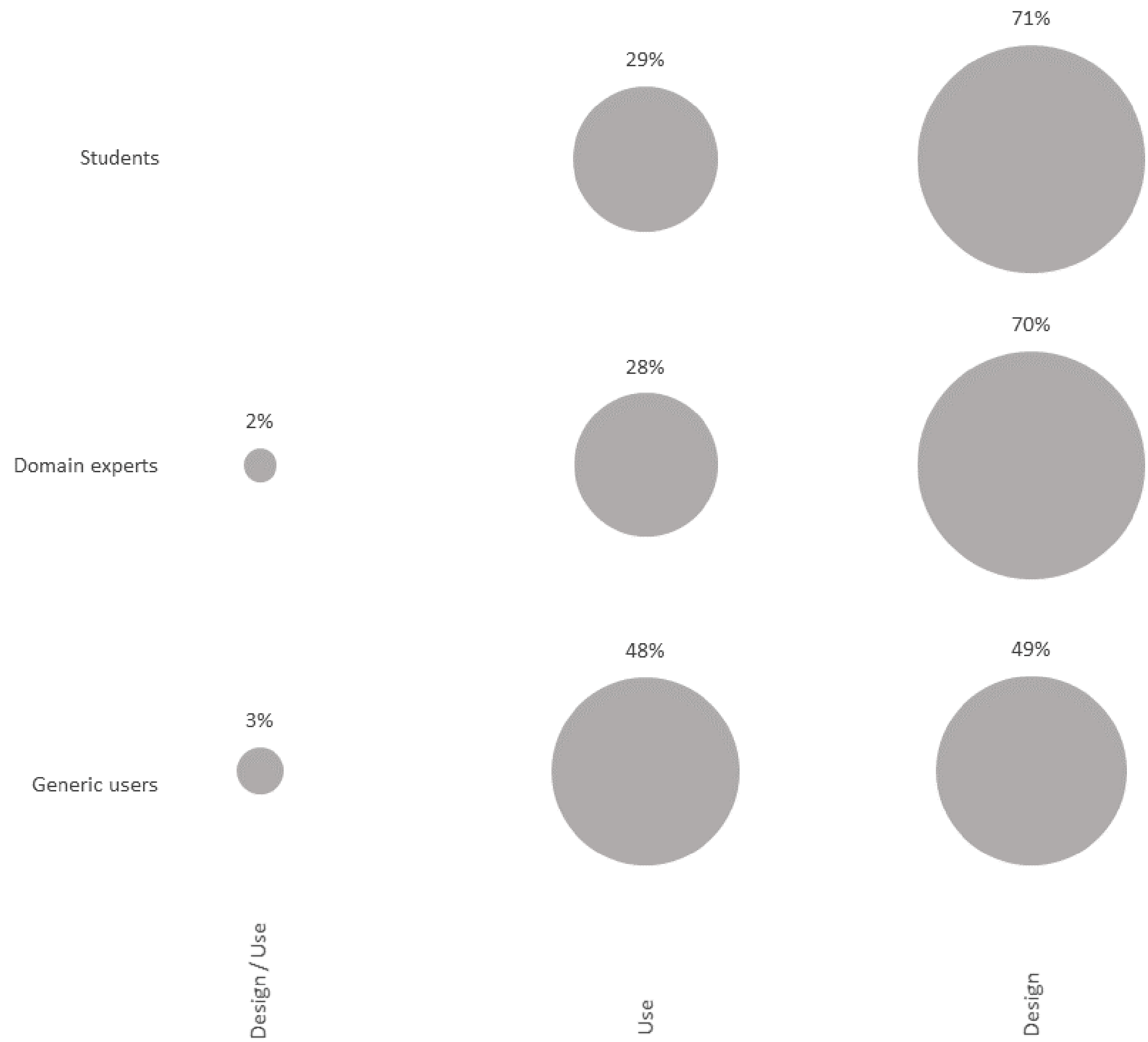

Fig. 16. Mapping D3 (phases) to D6 (classes of users).

Fig. 16. shows the results obtained from mapping the *phase of activity* dimension (D3) to the *class of users* dimension (D6). Here, one may observe that while *generic users* are almost equally involved in the *design* and *use* phases; the other target users (and especially *domain experts*) play a major role in the *design* phase. This indi-

cates that proposals often focus on supporting end users with specific professional (or educational) objectives to create their own digital artifacts, rather than to modify something built by an IT expert. However, we hope that future solutions consider both needs of personalized design and evolution at use time, since, even when the end user is supported to create a new artifact, he/she should be able to adapt it to new emerging requirements, according to the well-known co-evolution of systems and users phenomenon (Costabile et al., 2006); this twofold objective is pursued only in 4 papers of the 165 selected ones.

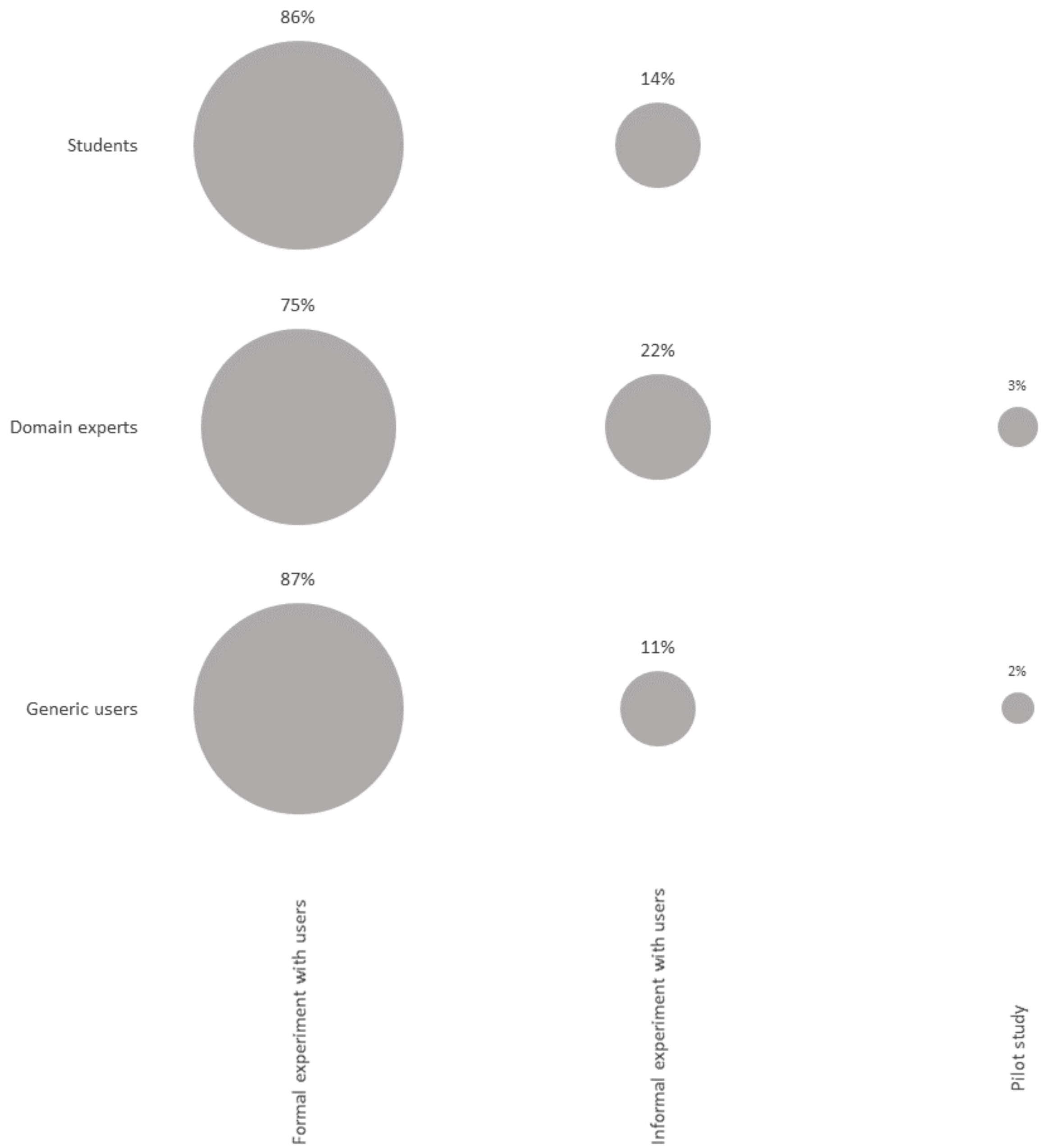

Fig. 17. Mapping D7 (empirical validation) to D6 (classes of users).

Finally, Fig. 17. shows the results obtained from mapping the *empirical validation* dimension (D7) to the *class of users* dimension (D6). As already observed, the most used approach for empirical validation was performing *formal experiments with users*. Specifically, when *generic users* were the target users, only few papers (13 papers out of 100) did not include a formal experiment; a similar result can be observed for papers that addressed *students* as class of users. However, higher rates of papers lack such type of evaluation when target users were *domain experts* (16 papers out of 64 papers). This result underlines how, in some projects, executing a rigorous experiment could be difficult, due to lack of time and availability of target users. However, since *formal experiment with users* represents the most accepted way in the HCI community to demonstrate the validity of an idea, researchers should strive to involve end users also in this phase, by convincing them of the importance of this kind of empirical validation.

5. Discussion

In this section, we will first summarize the main findings of our systematic mapping study; on the basis of these findings, we will then delineate some implications for future research in EUD, EUP and EUSE. Limitations of the study are discussed at the end of this section.

5.1 Findings

EUD, EUP and EUSE were born in the HCI field, and gave rise to specific conferences (e.g., the International Symposium on End-User Development), books (see (Lieberman et al., 2006b) and its sequel recently published by Springer (Paterno & Wulf, 2017)), and special issues in international journals (e.g., ACM TOCHI).
As a first general finding, the analysis carried out in this paper reveals that EUD, EUP and EUSE-related topics have attracted the attention of several research scholars also outside their own specific communities and the HCI field. Among the selected papers, there are papers published also on journals that do not have HCI in their main scope, such as Robotics and Autonomous Systems, Computer Standards & Interfaces, Computers & Education, Science of Computer Programming, Transportation Research Part C: Emerging Technologies, Future Generation Computing Systems, IEEE Transactions on Software Engineering, Information Systems, Journal of Medical Systems, Knowledge-Based Systems. Similarly, papers on EUD, EUP and EUSE have been published in the proceedings of several conferences, including International Conference on Games and Virtual Worlds for Serious Applications, IEEE International Symposium on Service Oriented System Engineering, Annual Frontiers in

Education Conference, International Conference on Interactive Mobile Communication Technologies and Learning, IEEE International Conference on Web Services, International Conference on Web Engineering, IEEE International Symposium on Software Reliability Engineering Workshops, International Conference on Software Engineering, and ACM SIGMOD International Conference on Management of Data. This finding demonstrates that the need for empowering end users to shape their digital artifacts is present in several domains and studied under different perspectives.

Second, as far as the definitions are concerned, we used the authors' classification of their proposed approach to classify papers as EUD, EUP or EUSE. However, selected papers sometimes do not adhere to known definitions available in literature, but just assume that their approach could be regarded as EUD or EUP or EUSE, thus generating misunderstandings in the reader and in the novice in the field. This might be related to the previous finding: indeed, the theme is currently spreading in different research areas, but scholars in these areas might not be aware of the existing literature and definitions.

Third, even though EUSE appears as fundamental for the quality of software created and modified by end users, the number of papers that propose a EUSE approach is very limited (only 9 papers were selected out of 165). This seems mainly due to the fact that only a small community (located in US) has been working on EUSE for a long time. In addition, most of these papers consider spreadsheet programming as a reference application and only very few works have tried to apply EUSE to other domains. Finally, only laboratory conditions are adopted for system testing in the papers proposing an EUSE approach.

A fourth finding is concerned with the phases of activities in which EUD and EUP approaches are employed. From the analysis, it emerges that several papers discussing EUD solutions propose domain-oriented design environments (Arias et al., 2015) to empower end users in the design of digital artifacts targeted not only to themselves, but more often to other people, who could be colleagues and co-workers, or users belonging to a different community, as in the case of multi-tiered proxy design problems (Fogli & Piccinno, 2013). EUP, on the other hand, is proposed indifferently for the design and use phases, and several papers discuss how to make programming easier for end users, rather than enabling them to create or adapt a digital artifact in an unwitting manner (Costabile et al., 2008). Actually, EUP dates back to the works by Bonnie Nardi (1993) and Alan Cypher (1993), which originally focused on code creation; whilst, the term EUD was coined later to denote different activities around the shaping of digital artifacts by end users.

As a fifth finding, we may observe that the domain that is most considered in recent publications is that of *smart objects and environments* (including IoT and AmI related aspects) (see Section 4.4), since users are more and more interested in defining the behavior of their devices and environments, such as a smart home. In addition, the analysis reveals that 24% (39 papers) of all selected papers propose solutions for empowering end users to create or adapt the behavior of hardware artifacts, such as smart objects, robots, healthcare devices or physical games.

Sixth, we may confirm the finding by Maceli that there is a lack of works that address the problem of end-user modifiable digital artifacts in voice-based computing research, social media, and crowdsourcing technologies (Maceli, 2017). On the other hand, we found a few recent papers that propose tangible and gesture-based interaction to support EUD or EUP, contrarily to Maceli's results.

As a seventh finding, we identified 14 types of techniques to support end users in shaping activities; among them, the most frequent technique that research scholars choose to deploy as a EUP or EUD solution is the *component-based* one. The habit of software developers to address the design of complex systems by applying the modularity principle of software engineering could be one of the main reasons for this polarization toward a component-based approach, as far as the intuitiveness of the interaction paradigm for most of the users.

Related to the previous finding, the analysis also reveals that the proposed techniques, different from the *component-based* one, are strictly related to the application domains, where they are designed by capitalizing on the habits and skills of the users (see the mapping between D2 and D4).

5.2 Implications

The findings discussed above have implications on the design of future tools empowering end users to modify or create digital artifacts and on the research directions that deserve scholars' attention.

First of all, the research community should agree on the different definitions of EUD, EUP and EUSE. As far as EUD is concerned, the definition coined in the frame of the EUD-Net and reported in (Lieberman et al., 2006a) is considered the reference for traditional software applications: "the set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact". However, given the recent technology evolutions and the importance of EUD in contexts different from pure software systems, Fischer et al. (2017) suggested to extend that definition as follows: EUD "encompasses methods, techniques, methodologies, situations, and socio-technical environments that allow end users to act as professionals in those domains in which they are not professionals". Indeed, end users can nowadays become end-user developers of systems that encompass a variety of software and hardware components, such as smart objects, interactive displays, robots, as well as any other interactive device available in an Internet of Things setting. As a consequence, new methods and tools are needed that span from software to hardware components of complex systems.

The most authoritative definition of EUP is probably the one provided in Ko et al. (2011): "programming to achieve the result of a program primarily for personal, rather public use"; while the definition of EUSE can be derived from one of the most influential papers in the field: "a holistic approach to the facets of software development in which end users engage. Its goal is to bring some of the gains from the software engineering community to end-user programming environments, without requiring training or even interest in traditional software engineering techniques" (Wilson et al., 2003)[a125]. In the analyzed papers, we observed that the solutions proposed in the frame of EUP and EUSE are mostly oriented to support coding activities. In particular, EUP focuses on making programming an easy activity also for users not expert in programming; while EUSE adds to EUP the possibility of debugging and testing the code created by the user, who will then use it for his/her personal or professional objectives. On the other hand, EUD provides a broader view of the problem, encompassing situations where end users participate in designing (not necessarily programming) their own tools or in developing systems for other people. In this broader view, EUD could be considered a general approach that encompasses both EUP and EUSE. To this end, more emphasis to the quality of the EUD process and of the resulting products is needed, in accordance with the research in EUSE. EUSE techniques, in fact, appear as fundamental also for the quality of digital artifacts designed, developed and modified by end users; in order words, end users should be provided with error-prevention and debugging facilities, in order to avoid that their systems lead to undesired behaviors or unpredictable situations (Burnett & Scaffidi, 2013). Since EUD should span throughout the development life cycle, the adoption of EUSE techniques may help dealing with documentation, reliability, maintainability and reusability of EUD solutions.

On the basis of the above considerations, we suggest here to refine the definition of EUD by taking into account that of EUSE and that provided in (Fischer et al., 2017) as follows: "EUD is the set of methods, techniques, tools, and socio-technical environments that allow end users to act as professionals in those ICT-related domains in which they are not professionals, by creating, modifying, extending and testing digital artifacts without requiring knowledge in traditional software engineering techniques".

Beyond definitions, another important implication for future research regards promoting a cross-fertilization of the three fields by considering the relationships existing among the techniques adopted in EUD, EUP and EUSE. As it emerges from our systematic mapping, EUD, EUP and EUSE are three overlapping solutions to the same problem, and a unifying framework could help understanding and dealing with their different facets. In particular, taking a meta-design perspective (Fischer & Giaccardi, 2006; Fischer et al., 2004) when proposing a new tool or technique might be helpful for communication among designers, domain experts and end users. Semiotic engineering is another general framework that could be adopted to better understand common and unique characteristics of EUD, EUP and EUSE (de Souza, 2005). De Souza has proposed to frame EUD in semiotic engineering (De Souza & Barbosa, 2006), and recent works have deepened that approach by means of specific case studies (Barricelli & Valtolina, 2017; de Souza, 2017; Dittrich & Piccinno, 2017; Fogli, 2017; Monteiro et al., 2017).

Related to the above issue and still considering the phases in which EUD/EUP/EUSE approaches have been adopted in the selected papers, it emerges that two needs should be satisfied: in several application domains, users require both to design new digital artifacts, and to evolve them at use time, through tailoring or extension activities. These two different needs should be considered and analyzed since the beginning of a project, in order to adopt proper frameworks and apply suitable design methodologies.

More in-depth investigation should then be carried out about the techniques adopted in the different application domains. In particular, it should be investigated whether the high frequency of component-based tools is related to an objective advantage of this technique or depends on the availability of development environments that support its implementation or, worst, is due to a bias of developers of EUD and EUP techniques, who are used to reason by decomposition (of problems) and composition (of solutions). Indeed, there are application domains where end users have their own visual languages and work practices; therefore, capitalizing on these ones for the sake of defining novel solutions might result a better way than composing virtual blocks or puzzles.

The above observation suggests at least three research directions:

1) Comparative studies of different techniques applied in the same domains with the same classes of users would be necessary; in the selection of papers for this mapping study we intentionally excluded the few papers that presented comparative studies, since they would not have answered our research question; however, a survey dedicated to this specific type of papers would be useful to investigate this issue.

2) In case there are regularities in the adoption of some techniques in specific domains or some techniques reveals successful for a particular application, it might be interesting to define design patterns for EUD, EUP and EUSE solutions.
3) Proposals in the EUD, EUP, and EUSE fields should be better related to the work practices and specific notations of end users in given domains, in order to design novel techniques that foster end users' participation in system modification and shaping over time without requiring a high cognitive effort. To this end, the Human-Work Interaction Design (HWID) (Barricelli et al., 2015; Valtolina et al., 2017) field may provide hints and suggestions for conducting this type of research; furthermore, it would be important to validate solutions through formal experimentations with domain experts, often missing in the selected papers.

5.3 Limitations of the study

Some limitations affect our systematic mapping study. First of all, we limited our research to the period January 2000 - May 2017. The year 2000 was chosen since the first special issue on the theme was published on the Computer-Supported Cooperative Work journal in 2000; anyway, the oldest papers we selected for the study were actually published in 2002. Then, we have had to limit our paper search to May 2017, in order to proceed with the analysis of papers, which required several months. At the date of submission, there are probably other published papers that could be of interest for this study.
Moreover, since our search strategy led to a total of 2717 papers, we decided to adopt several exclusion criteria to identify a set of significant papers that was feasible to analyze; this choice might have however biased our findings: for instance, we could have excluded original ideas appearing in PhD dissertations or preliminary papers not presenting an empirical validation.
Furthermore, we decided to only keep the most recent publications in all those cases where multiple papers reported on the same solution by the same authors; therefore, as to the seventh dimension (evaluation), it is likely that this caused a bias towards selecting papers with formal evaluations.
Inaccuracy in data extraction and misclassification of papers may occur in our results, since information has been extracted from papers by different reviewers, who have different experience in the research areas dealt with in the study. To alleviate this threat, during the application of exclusion criteria both in the first and second stage of paper selection (see Section 3.3), the work done by each reviewer was crosschecked (namely, each reviewer revised the work carried out by another reviewer), in order to avoid that some significant paper was wrongly

excluded; in addition, after the analysis of each selected paper, all doubtful situations were discussed at least by two reviewers.

Furthermore, there could be papers that describe methods and tools for EUD, EUP or EUSE that do not cite explicitly these concepts because the authors were not aware of these research fields; therefore, our study could have not reached this literature work neither through manual nor through automatic search.

Finally, our main purpose was just literature categorization, according to the usual goal of systematic mapping studies; therefore, we classified selected papers according to seven dimensions and identified some representative studies by means of a qualitative assessment based on publication venue and citations. For this reason, very few aggregations of empirical results have been performed: the identification of the 14 types of techniques represents one of the most interesting aggregations that we were able to propose.

6. Conclusion

In recent years, several scientific works have proposed methods and tools for empowering end users to carry out End-User Programming, End-User Development or End-User Software Engineering activities. However, no mapping studies exist that consider all these three overlapping approaches; only surveys and reviews of the work performed in one of them have been published, and often driven by researcher's expectations. This paper has presented a systematic mapping study in the research about EUD, EUP and EUSE with the aim of identifying the nature and extent of the available research. The selected 165 papers out of 2717 retrieved through a manual and an automatic search have been read and analyzed to answer our research question, by analyzing seven dimensions, from the type of the technique to the targeted class of users, from the application domain to the type of evaluation performed. The state-of-the-art in the three fields has thus been extracted, as well as trends and gaps have been identified. Implications for future research and directions of investigation have been delineated, by suggesting the development of aggregations and syntheses (see for instance the 14 classes of techniques) that could help designers and practitioners of future tools and systems. New research methods, such as a semiotic-based characterization of proposals, and the definition of design patterns of EUD, EUP and EUSE, have been also suggested.

Finally, the results obtained show the need of widespread and consolidated definitions, cross-fertilization among the three approaches (especially between EUSE and the other two fields), and more domain-dependent and user-oriented methods.

Future research is planned to study more general models and frameworks that could encompass the features of all the three approaches, without losing their specificity.

Acknowledgements

The authors would like to thank the anonymous reviewers for their fruitful comments and suggestions.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Akiki, P. A., Bandara, A. K., & Yu, Y. (2017). Visual simple transformations: Empowering end-users to wire internet of things objects. *ACM Transactions on Computer-Human Interaction, 24*(2). doi: 10.1145/3057857

Andersen, R., & Mørch, A. (2009). Mutual Development: A Case Study in Customer-Initiated Software Product Development. In V. Pipek, M. B. Rossen, B. deRuyter & V. Wulf (Eds.), *End-User Development* (pp. 31-49). Heidelberg: Springer.

Anderson, C. (2012). *Makers - The New Industrial Revolution*. New York: Crown Business.

Arias, E. G., Eden, H., & Fischer, G. (2015). *The Envisionment and Discovery Collaboratory (EDC): Explorations in Human-Centered Informatics* (Vol. 8): Morgan & Claypool Publishers LLC. doi: 10.2200/s00670ed1v01y201509hci032.

Barricelli, B. R., & Valtolina, S. (2015). Designing for End-User Development in the Internet of Things. In P. Díaz, V. Pipek, C. Ardito, C. Jensen, I. Aedo & A. Boden (Eds.), *End-User Development* (Vol. 9083, pp. 9-24): Springer International Publishing. doi: 10.1007/978-3-319-18425-8_2.

Barricelli, B. R., & Valtolina, S. (2017). A visual language and interactive system for end-user development of internet of things ecosystems. *Journal of Visual Languages & Computing, 40*(Supplement C), 1-19. doi: https://doi.org/10.1016/j.jvlc.2017.01.004

Barricelli, B. R., Valtolina, S., Gadia, D., Marzullo, M., Piazzi, C., & Garzulino, A. (2015). Participatory Action Design Research in Archaeological Context. In J. Abdelnour Nocera, B. R. Barricelli, A. Lopes, P. Campos & T. Clemmensen (Eds.), *Human Work Interaction Design. Work Analysis and Interaction Design Methods for Pervasive and Smart Workplaces: 4th IFIP 13.6 Working Conference, HWID 2015, London, UK, June 25-26, 2015, Revised Selected Papers* (pp. 192-211). Cham: Springer International Publishing. doi: 10.1007/978-3-319-27048-7_14.

Bellucci, A., Zarraonandia, T., Díaz, P., & Aedo, I. (2017). *End-user prototyping of cross-reality environments.* Proc. of Conference on Tangible, Embedded, and Embodied Interaction, 173-182. doi: 10.1145/3024969.3024975.

Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. (2008). *Using Mapping Studies in Software Engineering.* Proc. of PPIG 2008, 195-204.

Burnett, M. (2009). What Is End-User Software Engineering and Why Does It Matter? In V. Pipek, M. B. Rosson, B. Ruyter & V. Wulf (Eds.), *End-User Development* (Vol. 5435, pp. 15-28): Springer Berlin Heidelberg. doi: 10.1007/978-3-642-00427-8_2.

Burnett, M., Cook, C., & Rothermel, G. (2004). End-user software engineering. *Communications of the ACM, 47*(9), 53-58. doi: http://doi.acm.org/10.1145/1015864.1015889

Burnett, M., & Scaffidi, C. (2013). End-User Development. In M. Soegaard & R. F. Dam (Eds.), *The Encyclopedia of Human-Computer Interaction, 2nd*. Aarhus, Denmark: The Interaction Design Foundation.

Cabitza, F., Fogli, D., Lanzilotti, R., & Piccinno, A. (2017). Rule-based tools for the configuration of ambient intelligence systems: a comparative user study. *Multimedia Tools and Applications, 76*(4), 5221-5241. doi: 10.1007/s11042-016-3511-2

Costabile, M. F., Fogli, D., Letondal, C., Mussio, P., & Piccinno, A. (2003). *Domain-Expert Users and their Needs of Software Development.* Proc. of 2nd International Conference on Universal Access in Human-Computer Interaction, Crete, Greece, June 22-27, 532-536.

Costabile, M. F., Fogli, D., Marcante, A., & Piccinno, A. (2006). *Supporting Interaction and Co-evolution of Users and Systems.* Proc. of International Conference on Advanced Visual Interface, Venice, Italy, May, 23-26, 143-150. doi: 10.1145/1133265.1133294.

Costabile, M. F., Mussio, P., Parasiliti Provenza, L., & Piccinno, A. (2008). *Advanced visual systems supporting unwitting EUD.* Proc. of International Conference on Advanced Visual Interfaces (AVI), Naple, Italy, May 28-30, 313-316. doi: 10.1145/1385569.1385621.

Costabile, M. F., Mussio, P., Parasiliti Provenza, L., & Piccinno, A. (2009). Supporting End Users to Be Co-designers of Their Tools. In V. Pipek, M. B. Rosson, B. de Ruyter & V. Wulf (Eds.), *End-User Development* (Vol. 5435, pp. 70-85). Berlin / Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-00427-8.

Cypher, A. (Ed.). (1993). *Watch what I do: programming by demonstration*: MIT Press.

de Souza, C. S. (2005). *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*: The MIT Press.

de Souza, C. S. (2017). Semiotic Engineering: A cohering theory to connect EUD with HCI, CMC and more. In F. Paterno & V. Wulf (Eds.), *New Perspectives in End User Development*: Springer International Publishing. doi: 10.1007/978-3-319-60291-2.

De Souza, C. S., & Barbosa, S. D. J. (2006). A Semiotic Framing for End-User Development. In H. Lieberman, F. Paternò & V. Wulf (Eds.), *End User Development* (Vol. 9, pp. 401-426). Dordrecht, The Netherlands: Springer. doi: 10.1007/1-4020-5386-X_18.

Desolda, G., Ardito, C., & Matera, M. (2017). Empowering End Users to Customize their Smart Environments: Model, Composition Paradigms, and Domain-Specific Tools. *ACM Transactions on Computer-Human Interaction, 24*(2), 1-52. doi: 10.1145/3057859

Dittrich, Y., & Piccinno, A. (2017). Special Issue on Semiotics, Human-Computer Interaction and End-User Development. *Journal of Visual Languages & Computing, 40*, iv-vi. doi: https://doi.org/10.1016/S1045-926X(17)30107-6

Eisenberg, M., & Fischer, G. (1994). Programmable Design Environments: Integrating End-User Programming with Domain-Oriented Assistance *Human Factors in Computing Systems, CHI'94 (Boston, MA)* (pp. 431-437). New York: ACM.

Fernandez, A., Insfran, E., & Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology, 53*(8), 789-817. doi: http://dx.doi.org/10.1016/j.infsof.2011.02.007

Fischer, G. (2013). End-User Development: From Creating Technologies to Transforming Cultures. In Y. Dittrich, M. Burnett, A. Mørch & D. Redmiles (Eds.), *End-User Development* (Vol. 7897, pp. 217-222). Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-38706-7_16.

Fischer, G., Fogli, D., & Piccinno, A. (2017). Revisiting and Broadening the Meta-Design Framework for End-User Development. In F. Paternò & V. Wulf (Eds.), *New Perspectives in End-User Development* (pp. 61-97). Cham: Springer International Publishing. doi: 10.1007/978-3-319-60291-2_4.

Fischer, G., & Giaccardi, E. (2006). Meta-Design: A Framework for the Future of End User Development. In H. Lieberman, F. Paternò & V. Wulf (Eds.), *End User Development* (Vol. 9, pp. 427-457). Dordrecht, The Netherlands: Springer. doi: 10.1007/1-4020-5386-X_9.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A., & Mehandjiev, N. (2004). Meta-design: a manifesto for end-user development. *Communications of the ACM, 47*(9), 33-37. doi: http://doi.acm.org/10.1145/1015864.1015884

Fischer, G., Nakakoji, K., & Ye, Y. (2009). Metadesign: Guidelines for Supporting Domain Experts in Software Development. *IEEE Software, 26*(5), 37-44. doi: 10.1109/MS.2009.134

Fischer, G., & Ostwald, J. (2002). *Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation.* Proc. of Proceedings of the Participatory Design Conference (PDC'02), Malmö University, Sweden, 135-143.

Fogli, D. (2017). Weaving semiotic engineering in meta-design: A case study analysis. *Journal of Visual Languages & Computing, 40*(Supplement C), 113-127. doi: https://doi.org/10.1016/j.jvlc.2017.04.002

Fogli, D., Lanzilotti, R., & Piccinno, A. (2016). End-User Development Tools for the Smart Home: A Systematic Literature Review. In N. Streitz & P. Markopoulos (Eds.), *Distributed, Ambient and Pervasive Interactions: 4th International Conference, DAPI 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings* (pp. 69-79). Cham: Springer International Publishing. doi: 10.1007/978-3-319-39862-4_7.

Fogli, D., & Piccinno, A. (2013). Co-evolution of End-User Developers and Systems in Multi-tiered Proxy Design Problems. In Y. Dittrich, M. Burnett, A. Mørch & D. Redmiles (Eds.), *End-User Development* (Vol. 7897, pp. 153-168). Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-38706-7_12.

Garzotto, F., & Gonella, R. (2011). *An open-ended tangible environment for disabled children's learning.* Proc. of International Conference on Interaction Design and Children, Ann Arbor, Michigan, 52-61. doi: 10.1145/1999030.1999037.

Gulley, N. (2006). *Improving the Quality of Contributed Software on the MATLAB File Exchange.* Proc. of 2nd international workshop on End-User Software Engineering (WEUSE '06), Montréal, Quebec, Canada, April 23.

Hafidh, B., Al Osman, H., Arteaga-Falconi, J. S., Dong, H., & El Saddik, A. (2017). SITE: The simple internet of things enabler for smart homes. *IEEE Access, 5*, 2034-2049. doi: 10.1109/ACCESS.2017.2653079

Harrison, W. (2004). From the Editor: The Dangers of End-User Programming. *IEEE Software, 21*(4), 5-7. doi: 10.1109/MS.2004.13

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys, 37*(2), 83-137. doi: 10.1145/1089733.1089734

Kim, J. E., Fan, X., & Mosse, D. (2017). *Empowering end users for social internet of things.* Proc. of International Conference on Internet-of-Things Design and Implementation, 71-82. doi: 10.1145/3054977.3054987.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology, 51*(1), 7-15. doi: http://dx.doi.org/10.1016/j.infsof.2008.09.009

Kjeldskov, J., & Graham, C. (2003). A Review of Mobile HCI Research Methods. In L. Chittaro (Ed.), *Human-Computer Interaction with Mobile Devices and Services: 5th International Symposium, Mobile HCI 2003, Udine, Italy, September 2003. Proceedings* (pp. 317-335). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-540-45233-1_23.

Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M. B., Rothermel, G., Shaw, M., & Wiedenbeck, S. (2011). The state of the art in end-user software engineering. *ACM Computing Surveys, 43*(3), 1-44. doi: 10.1145/1922649.1922658

Lee, J., Garduño, L., Walker, E., & Burleson, W. (2013). *A tangible programming tool for creation of context-aware applications.* Proc. of ACM international joint conference on Pervasive and ubiquitous computing (UbiComp), Zurich, Switzerland, September 08 - 12, 391-400. doi: 10.1145/2493432.2493483.

Lieberman, H., Paternò, F., Klann, M., & Wulf, V. (2006a). End-User Development: An Emerging Paradigm. In H. Lieberman, F. Paternò & V. Wulf (Eds.), *End User Development* (Vol. 9, pp. 1-8). Dordrecht, The Netherlands: Springer. doi: 10.1007/1-4020-5386-X_1.

Lieberman, H., Paternò, F., & Wulf, V. (Eds.). (2006b). *End User Development* (Vol. 9). Dordrecht, The Netherlands: Springer. doi: 10.1007/1-4020-5386-X_9.

Maceli, M. G. (2017). Tools of the Trade: A Survey of Technologies in End-User Development Literature. In S. Barbosa, P. Markopoulos, F. Paternò, S. Stumpf & S. Valtolina (Eds.), *End-User Development: 6th International Symposium, IS-EUD 2017, Eindhoven, The Netherlands, June 13-15, 2017, Proceedings* (pp. 49-65). Cham: Springer International Publishing. doi: 10.1007/978-3-319-58735-6_4.

Martín, D., Alcarria, R., Sánchez-Picot, Á., & Robles, T. (2015). An Ambient Intelligence Framework for End-User Service Provisioning in a Hospital Pharmacy: a Case Study. *Journal of Medical Systems, 39*(10), 116. doi: 10.1007/s10916-015-0298-z

Monteiro, I. T., Salgado, L. C. de C., Mota, M. P., Sampaio, A. L., & de Souza, C. S. (2017). Signifying software engineering to computational thinking learners with AgentSheets and PoliFacets. *Journal of Visual Languages & Computing, 40*(Supplement C), 91-112. doi: https://doi.org/10.1016/j.jvlc.2017.01.005

Muller, M. J., & Kuhn, S. (1993). Participatory design. *Communications of the ACM, 36*(6), 24-28. doi: 10.1145/153571.255960

Myers, B. A., Ko, A. J., & Burnett, M. M. (2006). *Invited Research Overview: End-User Programming.* Proc. of CHI '06 Extended Abstracts on Human Factors in Computing Systems, Montréal, Québec, Canada, 75-80. doi: 10.1145/1125451.1125472.

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

Paternò, F. (2013). End User Development: Survey of an Emerging Field for Empowering People. *ISRN Software Engineering, 2013*, 11. doi: 10.1155/2013/532659

Paterno, F., & Wulf, V. (Eds.). (2017). *New Perspectives in End-User Development*. Cham, Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-60291-2.

Sammet, J. E. (1966). The use of English as a programming language. *Communications of the ACM, 9*(3), 228-230. doi: 10.1145/365230.365274

Sas, C., & Neustaedter, C. (2017). Exploring DIY practices of complex home technologies. *ACM Transactions on Computer-Human Interaction, 24*(2). doi: 10.1145/3057863

Scaffidi, C., Shaw, M., & Myers, B. (2005). *Estimating the Numbers of End Users and End User Programmers*. Paper presented at the IEEE Symposium on Visual Languages and Human-Centric Computing, 10.1109/vlhcc.2005.34.

Simonsen, J., & Robertson, T. (Eds.). (2013). *Routledge International Handbook of Participatory Design*. London, UK: Routledge.

Tetteroo, D., & Markopoulos, P. (2015). A Review of Research Methods in End User Development. In P. Díaz, V. Pipek, C. Ardito, C. Jensen, I. Aedo & A. Boden (Eds.), *End-User Development* (Vol. 9083, pp. 58-75): Springer International Publishing. doi: 10.1007/978-3-319-18425-8_5.

Turchi, T., Malizia, A., & Dix, A. (2017). TAPAS: A tangible End-User Development tool supporting the repurposing of Pervasive Displays. *Journal of Visual Languages & Computing, 39*(Supplement C), 66-77. doi: https://doi.org/10.1016/j.jvlc.2016.11.002

Valtolina, S., Barricelli, B. R., Fogli, D., Colosio, S., & Testa, C. (2017). Public Staff Empowerment in e-Government: A Human Work Interaction Design Approach. In S. Barbosa, P. Markopoulos, F. Paternò, S. Stumpf & S. Valtolina (Eds.), *End-User Development: 6th International Symposium, IS-EUD 2017, Eindhoven, The Netherlands, June 13-15, 2017, Proceedings* (pp. 119-134). Cham: Springer International Publishing. doi: 10.1007/978-3-319-58735-6_9.

van Doorn, M., de Vries, A., & Aarts, E. (2008). End-User Software Engineering of Smart Retail Environments: The Intelligent Shop Window. In E. Aarts, J. L. Crowley, B. de Ruyter, H. Gerhäuser, A. Pflaum, J. Schmidt & R. Wichert (Eds.), *Ambient Intelligence: European Conference, AmI 2008, Nuremberg, Germany, November 19-22, 2008. Proceedings* (pp. 157-174). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-89617-3_11.

Wilson, A., Burnett, M., Beckwith, L., Granatir, O., Casburn, L., Cook, C., Durham, M., & Rothermel, G. (2003). *Harnessing curiosity to increase correctness in end-user programming.* Proc. of Conference on Human Factors in Computing Systems, Ft. Lauderdale, FL, April 5-10, 305-312.

Appendix A: Selected papers

a1. Pane, J.F., Myers, B.A.: More Natural Programming Languages and Environments. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development, pp. 31-50. Springer Netherlands, Dordrecht (2006)

a2. Won, M., Stiemerling, O., Wulf, V.: Component-Based Approaches to Tailorable Systems. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development, pp. 115-141. Springer Netherlands, Dordrecht (2006)

a3. Stevens, G., Quaisser, G., Klann, M.: Breaking It Up: An Industrial Case Study of Component-Based Tailorable Software Design. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development, pp. 269-294. Springer Netherlands, Dordrecht (2006)

a4. Ghiani, G., Paternò, F., Spano, L.D.: Cicero Designer: An Environment for End-User Development of Multi-Device Museum Guides. In: Pipek, V., Rosson, M.B., de Ruyter, B., Wulf, V. (eds.) End-User Development: 2nd International Symposium, IS-EUD 2009, Siegen, Germany, March 2-4, 2009. Proceedings, pp. 265-274. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

a5. Spahn, M., Wulf, V.: End-User Development of Enterprise Widgets. In: Pipek, V., Rosson, M.B., de Ruyter, B., Wulf, V. (eds.) End-User Development: 2nd International Symposium, IS-EUD 2009, Siegen, Germany, March 2-4, 2009. Proceedings, pp. 106-125. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

a6. Cappiello, C., Daniel, F., Matera, M., Picozzi, M., Weiss, M.: Enabling End User Development through Mashups: Requirements, Abstractions and Innovation Toolkits. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) End-User Development: Third International Symposium, IS-EUD 2011, Torre Canne (BR), Italy, June 7-10, 2011. Proceedings, pp. 9-24. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

a7. Deng, Y., Churcher, C., Abell, W., McCallum, J.: Designing a Framework for End User Applications. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) End-User Development: Third International Symposium, IS-EUD 2011, Torre Canne (BR), Italy, June 7-10, 2011. Proceedings, pp. 67-75. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

a8. Repenning, A., Ahmadi, N., Repenning, N., Ioannidou, A., Webb, D., Marshall, K.: Collective Programming: Making End-User Programming (More) Social. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) End-User Development: Third International Symposium, IS-EUD 2011, Torre Canne (BR), Italy, June 7-10, 2011. Proceedings, pp. 325-330. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

a9. Díaz, P., Malizia, A., Navarro, I., Aedo, I.: Using Recommendations to Help Novices to Reuse Design Knowledge. In: Costabile, M.F., Dittrich, Y., Fischer, G., Piccinno, A. (eds.) End-User Development: Third International Symposium, IS-EUD 2011, Torre Canne (BR), Italy, June 7-10, 2011. Proceedings, pp. 331-336. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

a10. Booth, T., Stumpf, S.: End-User Experiences of Visual and Textual Programming Environments for Arduino. In: Dittrich, Y., Burnett, M., Mørch, A., Redmiles, D. (eds.) End-User Development: 4th International Symposium, IS-EUD 2013, Copenhagen, Denmark, June 10-13, 2013. Proceedings, pp. 25-39. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

a11. Zhu, L., Herrmann, T.: Meta-design in Co-located Meetings. In: Dittrich, Y., Burnett, M., Mørch, A., Redmiles, D. (eds.) End-User Development: 4th International Symposium, IS-EUD 2013, Copenhagen, Denmark, June 10-13, 2013. Proceedings, pp. 169-184. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

a12. Fogli, D.: Cultures of Participation in Community Informatics: A Case Study. In: Dittrich, Y., Burnett, M., Mørch, A., Redmiles, D. (eds.) End-User Development: 4th International Symposium, IS-EUD 2013, Copenhagen, Denmark, June 10-13, 2013. Proceedings, pp. 201-216. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

a13. Fry, C., Lieberman, H.: Decision-Making Should Be More Like Programming. In: Dittrich, Y., Burnett, M., Mørch, A., Redmiles, D. (eds.) End-User Development: 4th International Symposium, IS-EUD 2013, Copenhagen, Denmark, June 10-13, 2013. Proceedings, pp. 247-253. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
a14. Dax, J., Ludwig, T., Meurer, J., Pipek, V., Stein, M., Stevens, G.: FRAMES – A Framework for Adaptable Mobile Event-Contingent Self-report Studies. In: Díaz, P., Pipek, V., Ardito, C., Jensen, C., Aedo, I., Boden, A. (eds.) End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings, pp. 141-155. Springer International Publishing, Cham (2015)
a15. Fogli, D., Giaccardi, E., Acerbis, A., Filisetti, F.: Physical Prototyping of Social Products Through End-User Development. In: Díaz, P., Pipek, V., Ardito, C., Jensen, C., Aedo, I., Boden, A. (eds.) End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings, pp. 217-222. Springer International Publishing, Cham (2015)
a16. Menestrina, Z., De Angeli, A., Busetta, P.: APE: End User Development for Emergency Management Training. In Proc. of: International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), pp. 1-4. IEEE (2014)
a17. Xinyi, L., Hailong, S., Hanxiong, W., Richong, Z., Xudong, L.: Using Sequential Pattern Mining and Interactive Recommendation to Assist Pipe-like Mashup Development. In Proc. of: International Symposium on Service Oriented System Engineering, pp. 173-180. IEEE (2014)
a18. Ahern, T.C.: The effectiveness of visual programming for model building in middle school. In Proc. of: Annual Frontiers in Education Conference, pp. S3D-8-S3D-13. IEEE (2008)
a19. Tzafilkou, K., Chouliara, A., Protogeros, N., Karagiannidis, C., Koumpis, A.: Engaging end-users in creating data-intensive mobile applications: A creative ???e-learning-by-doing??? approach. In Proc. of: International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), pp. 274-278. IEEE (2015)
a20. Akiki, P.A., Bandara, A.K., Yu, Y.: Visual simple transformations: Empowering end-users to wire internet of things objects. ACM Transactions on Computer-Human Interaction, 24(2)(2017). doi: 10.1145/3057857
a21. Desolda, G., Ardito, C., Matera, M.: Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools. ACM Transactions on Computer-Human Interaction, 24(2)(2017). doi: 10.1145/3057859
a22. Ghiani, G., Manca, M., Paterno, F., Santoro, C.: Personalization of context-dependent applications through trigger-action rules. ACM Transactions on Computer-Human Interaction, 24(2)(2017). doi: 10.1145/3057861
a23. Chen, X., Li, Y.: Improv: An input framework for improvising cross-device interaction by demonstration. ACM Transactions on Computer-Human Interaction, 24(2)(2017). doi: 10.1145/3057862
a24. Korpipää, P., Malm, E.J., Rantakokko, T., Kyllönen, V., Kela, J., Mäntyjärvi, J., Häkkilä, J., Känsälä, I.: Customizing user interaction in smart phones. IEEE Pervasive Computing, 5(3), 82-90 (2006). doi: 10.1109/MPRV.2006.49
a25. Tzeremes, V., Gomaa, H.: XANA: An end user software product line framework for smart spaces. In Proc., pp. 5831-5840. IEEE Computer Society (2016)
a26. Janner, T., Siebeck, R., Schroth, C., Hoyer, V.: Patterns for enterprise mashups in B2B collaborations to foster lightweight composition and end user development. In Proc., pp. 976-983. (2009)
a27. Danado, J., Paternò, F.: Puzzle: A mobile application development environment using a jigsaw metaphor. Journal of Visual Languages and Computing, 25(4), 297-315 (2014). doi: 10.1016/j.jvlc.2014.03.005
a28. Ioannidou, A., Repenning, A., Webb, D.C.: AgentCubes: Incremental 3D end-user development. Journal of Visual Languages and Computing, 20(4), 236-251 (2009). doi: 10.1016/j.jvlc.2009.04.001
a29. Lizcano, D., Alonso, F., Soriano, J., López, G.: A component- and connector-based approach for end-user composite web applications development. Journal of Systems and Software, 94, 108-128 (2014). doi: 10.1016/j.jss.2014.03.039
a30. Faravelon, A., Céret, E.: Privacy conscious web apps composition. In Proc. of: IEEE International Conference on Research Challenges in Information Science, pp. 1-12. IEEE Computer Society (2014)
a31. Dinmore, M.: Design and evaluation of a literate spreadsheet. In Proc. of: Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 15-18. IEEE (2012)
a32. Ferreira, J.J., De Souza, C.S., De Castro Salgado, L.C., Slaviero, C., Leitao, C.F., Moreira, F.D.F.: Combining cognitive, semiotic and discourse analysis to explore the power of notations in visual programming. In Proc. of: Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 101-108. IEEE (2012)
a33. Chin, J.S.Y., Callaghan, V., Clarke, G.: A programming-by-example approach to customising digital homes. In Proc. of: IET International Conference on Intelligent Environments, pp. 1-8. IEEE (2008)
a34. Han, H., Tokuda, T.: A Method for Integration of Web Applications Based on Information Extraction. In Proc. of: International Conference on Web Engineering, pp. 189-195. IEEE (2008)
a35. Buchina, N., Kamel, S., Barakova, E.: Design and evaluation of an end-user friendly tool for robot programming. In Proc. of: IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 185-191. IEEE (2016)
a36. Hofer, B., Wotawa, F.: Mutation-based spreadsheet debugging. In Proc. of: International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 132-137. IEEE (2013)
a37. Wiedenbeck, S., Engebretson, A.: Comprehension Strategies of End-User Programmers in an Event-Driven Application. In Proc. of: Symposium on Visual Languages - Human Centric Computing, pp. 207-214. IEEE (2004)
a38. Barakova, E.I., Gillesen, J.C.C., Huskens, B.E.B.M., Lourens, T.: End-user programming architecture facilitates the uptake of robots in social therapies. Robotics and Autonomous Systems, 61(7), 704-713 (2013). doi: 10.1016/j.robot.2012.08.001
a39. Fogli, D., Parasiliti Provenza, L.: A meta-design approach to the development of e-government services. Journal of Visual Languages and Computing, 23(2), 47-62 (2012). doi: 10.1016/j.jvlc.2011.11.003
a40. Sutcliffe, A., Papamargaritis, G.: End-user development by application-domain configuration. Journal of Systems and Software, 91(1), 85-99 (2014). doi: 10.1016/j.jss.2013.11.1121
a41. Schobel, J., Pryss, R., Schickler, M., Ruf-Leuschner, M., Elbert, T., Reichert, M.: End-User Programming of Mobile Services: Empowering Domain Experts to Implement Mobile Data Collection Applications. In Proc. of: International Conference on Mobile Services (MS), pp. 1-8. IEEE (2016)

a42. March, A.I.: End-user development and learning in second life: The box as multipurpose building block. In Proc. of: Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 208-212. IEEE (2016)

a43. Smiley, K., Patel, P., Harding, J.: From Ideas to Implementations: Closing the Gaps between Technical Experts and Software Solutions. In Proc. of: International Workshop on Continuous Software Evolution and Delivery (CSED), pp. 1-4. IEEE (2016)

a44. Nakazawa, R., Ohmura, R.: Cooperation definition method for smart appliances with single colloquial sentence. In Proc. of: International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), pp. 1-5. IEEE (2016)

a45. Burnett, M., Cook, C., Pendse, O., Rothermel, G., Summet, J., Wallace, C.: End-user software engineering with assertions in the spreadsheet paradigm. In Proc. of: International Conference on Software Engineering. Proceedings., pp. 93-103. IWWW (2003)

a46. Kuttal, S.K., Sarma, A., Rothermel, G.: On the benefits of providing versioning support for end users: An empirical study. ACM Transactions on Computer-Human Interaction, 21(2), 1-43 (2014). doi: 10.1145/2560016

a47. Obrenovic, Z., Martens, J.B.: Sketching interactive systems with Sketchify. ACM Transactions on Computer-Human Interaction, 18(1), 1-38 (2011). doi: 10.1145/1959022.1959026

a48. Kulesza, T., Stumpf, S., Wong, W.K., Burnett, M.M., Perona, S., Ko, A., Oberst, I.: Why-oriented end-user debugging of naive Bayes text classification. ACM Transactions on Interactive Intelligent Systems, 1(1), 1-31 (2011). doi: 10.1145/2030365.2030367

a49. Cappiello, C., Matera, M., Picozzi, M.: A UI-centric approach for the End-User Development of multidevice mashups. ACM Transactions on the Web, 9(3), 1-40 (2015). doi: 10.1145/2735632

a50. Díaz, O., Arellano, C.B., Azanza, M.: A language for end-user Web augmentation: Caring for producers and consumers alike. ACM Transactions on the Web, 7(2), 1-51 (2013). doi: 10.1145/2460383.2460388

a51. Fang, L., Hepting, D.H.: Assessing End-User Programming for a Graphics Development Environment. In: An, A., Lingras, P., Petty, S., Huang, R. (eds.) Active Media Technology: 6th International Conference, AMT 2010, Toronto, Canada, August 28-30, 2010. Proceedings, pp. 411-423. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

a52. van Doorn, M., de Vries, A., Aarts, E.: End-User Software Engineering of Smart Retail Environments: The Intelligent Shop Window. In: Aarts, E., Crowley, J.L., de Ruyter, B., Gerhäuser, H., Pflaum, A., Schmidt, J., Wichert, R. (eds.) Ambient Intelligence: European Conference, AmI 2008, Nuremberg, Germany, November 19-22, 2008. Proceedings, pp. 157-174. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)

a53. Davidyuk, O., Gilman, E., Milara, I.S., Mäkipelto, J., Pyykkönen, M., Riekki, J.: iCompose: context-aware physical user interface for application composition. Central European Journal of Computer Science, 1(4), 442-465 (2011). doi: 10.2478/s13537-011-0031-z

a54. Xue, S., Wu, B., Cheng, B., Qiao, X., Chen, J.: Mobile business process personalization for end users. China Communications, 12(12), 1-12 (2015). doi: 10.1109/CC.2015.7385516

a55. Lizcano, D., López, G., Soriano, J., Lloret, J.: Implementation of end-user development success factors in mashup development environments. Computer Standards and Interfaces, 47, 1-18 (2016). doi: 10.1016/j.csi.2016.02.006

a56. Martín, D., Lamsfus, C., Alzua-Sorzabal, A.: A cloud-based platform to develop context-aware mobile applications by domain experts. Computer Standards and Interfaces, 44, 177-184 (2016). doi: 10.1016/j.csi.2015.08.009

a57. Su, J.M., Huang, C.F.: An easy-to-use 3D visualization system for planning context-aware applications in smart buildings. Computer Standards and Interfaces, 36(2), 312-326 (2014). doi: 10.1016/j.csi.2012.07.004

a58. Marchiori, E.J., Torrente, J., Del Blanco, A., Moreno-Ger, P., Sancho, P., Fernández-Manjón, B.: A narrative metaphor to facilitate educational game authoring. Computers and Education, 58(1), 590-599 (2012). doi: 10.1016/j.compedu.2011.09.017

a59. Collins, T.D., Fung, P.: A visual programming approach for teaching cognitive modeling. Computers and Education, 39(1), 1-18 (2002). doi: 10.1016/S0360-1315(01)00087-2

a60. Nicolaos, P., Katerina, T.: Simple-talking database development: Let the end-user design a relational schema by using simple words. Computers in Human Behavior, 48, 273-289 (2015). doi: 10.1016/j.chb.2015.02.002

a61. Guo, B., Zhang, D., Imai, M.: Toward a cooperative programming framework for context-aware applications. Personal and Ubiquitous Computing, 15(3), 221-233 (2011). doi: 10.1007/s00779-010-0329-1

a62. Welbourne, E., Balazinska, M., Borriello, G., Fogarty, J.: Specification and Verification of Complex Location Events with Panoramic. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive Computing: 8th International Conference, Pervasive 2010, Helsinki, Finland, May 17-20, 2010. Proceedings, pp. 57-75. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

a63. Wachtel, A., Weigelt, S., Tichy, W.F.: Initial Implementation of Natural Language Turn-Based Dialog System. In Proc., pp. 49-56. Elsevier B.V. (2016)

a64. Garzotto, F., Gonella, R.: An open-ended tangible environment for disabled children's learning. In Proc. of: International Conference on Interaction Design and Children, pp. 52-61. ACM (2011)

a65. Gardiner, S., Tomasic, A., Zimmerman, J.: SmartWrap: Seeing datasets with the crowd's eyes. In Proc. of: Web for All Conference, pp. 1-10. Association for Computing Machinery, Inc (2015)

a66. Akama, H., Yamamuro, M., Menjo, T., Satoh, T.: Integrating multimedia data processing parts in cloud into folder programming environment. In Proc. of: International Conference on Information Integration and Web-based Applications and Services, pp. 130-137. ACM (2011)

a67. Bogdan, C., Kaindl, H., Falb, J., Popp, R.: Modeling of interaction design by end users through discourse modeling. In Proc. of: International Conference on Intelligent User Interfaces, pp. 305-308. ACM (2008)

a68. Nichols, J., Lau, T.: Mobilization by demonstration: Using traces to re-author existing web sites. In Proc. of: International Conference on Intelligent User Interfaces, pp. 149-158. ACM (2008)

a69. Zbick, J., Nake, I., Jansen, M., Milrad, M.: MLearn4web - A web-based framework to design and deploy cross-platform mobile applications. In Proc. of: International Conference on Mobile and Ubiquitous Multimedia, pp. 252-255. Association for Computing Machinery (2014)

a70. Lin, J., Wong, J., Nichols, J., Cypher, A., Lau, T.A.: End-user programming of mashups with Vegemite. In Proc. of: International Conference on Intelligent User Interfaces, pp. 97-106. ACM (2009)

a71. Gervasio, M.T., Murdock, J.L.: What were you thinking? Filling in missing dataflow through inference in learning from demonstration. In Proc. of: International Conference on Intelligent User Interfaces, pp. 157-166. ACM (2009)

a72. Maués, R.D.A., Barbosa, S.D.J.: Keep doing what I just did: Automating smartphones by demonstration. In Proc. of: International Conference on Human-computer Interaction with Mobile Devices and Services, pp. 295-303. ACM (2013)
a73. Koesnandar, A., Elbaum, S., Rothermel, G., Hochstein, L., Scaffidi, C., Stolee, K.T.: Using assertions to help end-user programmers create dependable web macros. In Proc. of: International Symposium on Foundations of Software Engineering, pp. 124-134. ACM (2008)
a74. Wang, G., Yang, S., Han, Y.: Mashroom: End-user mashup programming using nested tables. In Proc. of: International Conference on World Wide Web, pp. 861-870. ACM (2009)
a75. Little, G., Miller, R.C.: Translating keyword commands into executable code. In Proc. of: Annual ACM Symposium on User Interface Software and Technology, pp. 135-144. ACM (2008)
a76. Van Kleek, M., Moore, B., Karger, D.R., André, P., Schraefel, M.C.: Atomate it! End-user context-sensitive automation using heterogeneous information sources on the web. In Proc. of: International World Wide Web Conferenc, pp. 951-960. ACM (2010)
a77. Hartmann, M., Schreiber, D., Mühlhäuser, M.: AUGUR: Providing context-aware interaction support. In Proc. of: Symposium on Engineering Interactive Computing Systems, pp. 123-131. ACM (2009)
a78. Starbird, K., Palen, L.: More than the usual suspects: The physical self and other resources for learning to program using a 3D Avatar environment. In Proc. of: iConference: Inspiration, Integrity, and Intrepidity, pp. 614-621. ACM (2011)
a79. Criel, J., Vanderhulst, G., Kawsar, F., Trappeniers, L.: A co-creation platform for creative engagement of end-users in a connected object space. In Proc. of: International Workshop on Networking and Object Memories for the Internet of Things, pp. 9-13. ACM (2011)
a80. Lee, J., Garduño, L., Walker, E., Burleson, W.: A tangible programming tool for creation of context-aware applications. In Proc. of: International Joint Conference on Pervasive and Ubiquitous Computing, pp. 391-400. ACM (2013)
a81. Repenning, A.: Conversational programming exploring interactive program analysis. In Proc. of: International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, pp. 63-74. Association for Computing Machinery (2013)
a82. Gulwani, S., Marron, M.: NLyze: Interactive programming by natural language for spreadsheet data analysis and manipulation. In Proc. of: International Conference on Management of Data, pp. 803-814. Association for Computing Machinery (2014)
a83. Maleki, M.M., Woodbury, R.F., Neustaedter, C.: Liveness, localization and lookahead: Interaction elements for parametric design. In Proc. of: Conference on Designing Interactive Systems, pp. 805-814. Association for Computing Machinery (2014)
a84. Haraty, M., McGrenere, J.: Designing for advanced personalization in personal task management. In Proc. of: Conference on Designing Interactive Systems, pp. 239-250. Association for Computing Machinery, Inc (2016)
a85. Scaffidi, C., Dove, A., Nabi, T.: LondonTube: Overcoming hidden dependencies in cloud-mobile-web programming. In Proc. of: Conference on Human Factors in Computing Systems, pp. 3498-3508. Association for Computing Machinery (2016)
a86. Chang, K.S.P., Myers, B.A.: Using and exploring hierarchical data in spreadsheets. In Proc. of: Conference on Human Factors in Computing Systems, pp. 2497-2507. Association for Computing Machinery (2016)
a87. Houben, S., Golsteijn, C., Gallacher, S., Johnson, R., Bakker, S., Marquardt, N., Capra, L., Rogers, Y.: Physikit: Data engagement through physical ambient visualizations in the home. In Proc. of: Conference on Human Factors in Computing Systems, pp. 1608-1619. Association for Computing Machinery (2016)
a88. Huang, J., Cakmak, M.: Code3: A System for End-to-End Programming of Mobile Manipulator Robots for Novices and Experts. In Proc. of: International Conference on Human-Robot Interaction, pp. 453-462. IEEE Computer Society (2017)
a89. Jia-Jun Li, T., Azaria, A., A. Myers, B.: SUGILITE: Creating Multimodal Smartphone Automation by Demonstration. Conference on Human Factors in Computing Systems. ACM, Denver, Colorado, USA (2017) 6038-6049
a90. Castelli, N., Ogonowski, C., Jakobi, T., Stein, M., Stevens, G., Wulf, V.: What Happened in my Home?: An End-User Development Approach for Smart Home Data Visualization. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM, Denver, Colorado, USA (2017) 853-866
a91. Stratton, A., Bates, C., Dearden, A.: Quando: Enabling Museum and Art Gallery Practitioners to Develop Interactive Digital Exhibits. In: Barbosa, S., Markopoulos, P., Paternò, F., Stumpf, S., Valtolina, S. (eds.) End-User Development: 6th International Symposium, IS-EUD 2017, Eindhoven, The Netherlands, June 13-15, 2017, Proceedings, pp. 100-107. Springer International Publishing, Cham (2017)
a92. Corcella, L., Manca, M., Paternò, F.: Personalizing a Student Home Behaviour. In: Barbosa, S., Markopoulos, P., Paternò, F., Stumpf, S., Valtolina, S. (eds.) End-User Development: 6th International Symposium, IS-EUD 2017, Eindhoven, The Netherlands, June 13-15, 2017, Proceedings, pp. 18-33. Springer International Publishing, Cham (2017)
a93. Berenz, V., Suzuki, K.: Targets-Drives-Means: A declarative approach to dynamic behavior specification with higher usability. Robotics and Autonomous Systems, 62(4), 545-555 (2014). doi: 10.1016/j.robot.2013.12.010
a94. Riss, U.V.: TAPIR: Wiki-Based Task and Personal Information Management Supporting Subjective Process Management. In: Oppl, S., Fleischmann, A. (eds.) S-BPM ONE - Education and Industrial Developments: 4th International Conference, S-BPM ONE 2012, Vienna, Austria, April 4-5, 2012. Proceedings, pp. 220-235. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
a95. Eagan, J.R., Stasko, J.T.: The Buzz: Supporting user tailorability in awareness applications. In Proc. of: Conference on Human Factors in Computing Systems, pp. 1729-1738. ACM (2008)
a96. Rojas, L.A., Macías, J.A.: Bridging the gap between information architecture analysis and software engineering in interactive web application development. Science of Computer Programming, 78(11), 2282-2291 (2013). doi: 10.1016/j.scico.2012.07.020
a97. Hermans, F., Pinzger, M., Van Deursen, A.: Supporting professional spreadsheet users by generating leveled dataflow diagrams. In Proc. of: International Conference on Software Engineering, pp. 451-460. ACM (2011)
a98. Lagares Lemos, A., Chai Barukh, M., Benatallah, B.: DataSheets: A Spreadsheet-Based Data-Flow Language. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) Service-Oriented Computing: 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings, pp. 616-623. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
a99. Lawrance, J., Abraham, R., Burnett, M., Erwig, M.: Sharing reasoning about faults in spreadsheets: An empirical study. In Proc. of: 2006 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2006, Co-located with the 2006 Symposium on Software Visualisation, SoftVis 2006, and the 2006 Conference of the Psychology of Programming Interest Group, PPIG 2006, pp. 35-42. IEEE (2006)

a100. Saad-Sulonen, J., Cabrera, A.B.: Setting up a public participation project using the urban mediator tool: A case of collaboration between designers and city planners. In Proc. of: Nordic Conference on Human-Computer Interaction, pp. 539-542. ACM (2008)
a101. Rea, D.J., Igarashi, T., Young, J.E.: Paint board- Prototyping interactive character behaviors by digitally painting storyboards. In Proc. of: International Conference on Human-Agent Interaction, pp. 315-322. Association for Computing Machinery, Inc (2014)
a102. Grönvall, E., Kramp, G.: LinkLights: A modular, user adaptable system to support rehabilitation practices. In Proc. of: International Conference on PErvasive Technologies Related to Assistive Environments, pp. 1-8. ACM (2011)
a103. Tetteroo, D., Vreugdenhil, P., Grisel, I., Michielsen, M., Kuppens, E., Vanmulken, D., Markopoulos, P.: Lessons learnt from deploying an end-user development platform for physical rehabilitation. In Proc. of: Conference on Human Factors in Computing Systems, pp. 4133-4142. Association for Computing Machinery (2015)
a104. Abraham, R., Erwig, M.: Inferring templates from spreadsheets. In Proc. of: International Conference on Software Engineering, pp. 182-191. ACM (2006)
a105. Bhatti, G., Brémond, R., Jessel, J.P., Dang, N.T., Vienne, F., Millet, G.: Design and evaluation of a user-centered interface to model scenarios on driving simulators. Transportation Research Part C: Emerging Technologies, 50, 3-12 (2015). doi: 10.1016/j.trc.2014.09.011
a106. Romano, M., Aedo, I., Díaz, P.: Designing an End-User Augmented Reality Editor for Cultural Practitioners. In: García, C.R., Caballero-Gil, P., Burmester, M., Quesada-Arencibia, A. (eds.) Ubiquitous Computing and Ambient Intelligence: 10th International Conference, UCAmI 2016, San Bartolomé de Tirajana, Gran Canaria, Spain, November 29 – December 2, 2016, Proceedings, Part I, pp. 365-371. Springer International Publishing, Cham (2016)
a107. Carmien, S.P., Fischer, G.: Design, adoption, and assessment of a socio-technical environment supporting independence for persons with cognitive disabilities. In Proc. of: Conference on Human Factors in Computing Systems, pp. 597-606. ACM (2008)
a108. Repenning, A., Ioannidou, A.: Agent Warp Engine: Formula based shape warping for networked applications. In Proc. of: Working Conference on Advanced Visual Interfaces, pp. 279-286. ACM (2008)
a109. Catala, A., Pons, P., Jaen, J., Mocholi, J.A., Navarro, E.: A meta-model for dataflow-based rules in smart environments: Evaluating user comprehension and performance. Science of Computer Programming, 78(10), 1930-1950 (2013). doi: 10.1016/j.scico.2012.06.010
a110. Nebeling, M., Leone, S., Norrie, M.C.: Crowdsourced Web Engineering and Design. In: Brambilla, M., Tokuda, T., Tolksdorf, R. (eds.) Web Engineering: 12th International Conference, ICWE 2012, Berlin, Germany, July 23-27, 2012. Proceedings, pp. 31-45. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
a111. Radeck, C., Blichmann, G., Meißner, K.: CapView – Functionality-Aware Visual Mashup Development for Non-programmers. In: Daniel, F., Dolog, P., Li, Q. (eds.) Web Engineering: 13th International Conference, ICWE 2013, Aalborg, Denmark, July 8-12, 2013. Proceedings, pp. 140-155. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
a112. Hess, J., Wan, L., Pipek, V., Kuestermann, G.: Using paper and pen to control home-IT: Lessons learned by hands-on experience. In Proc. of: European Interactive TV Conference, pp. 203-211. ACM (2011)
a113. Smith, J.D., Graham, T.C.N.: Raptor: Sketching games with a tabletop computer. In Proc. of: International Academic Conference on the Future of Game Design and Technology, pp. 191-198. ACM (2010)
a114. Fogli, D., Colosio, S., Sacco, M.: Managing accessibility in local e-government websites through end-user development: a case study. Universal Access in the Information Society, 9(1), 35-50 (2010). doi: 10.1007/s10209-009-0158-z
a115. Barricelli, B.R., Valtolina, S.: A visual language and interactive system for end-user development of internet of things ecosystems. Journal of Visual Languages and Computing, 40, 1-19 (2017). doi: 10.1016/j.jvlc.2017.01.004
a116. Namoun, A., Wajid, U., Mehandjiev, N., Owrak, A.: User-centered design of a visual data mapping tool. In Proc. of: International Working Conference on Advanced Visual Interfaces, pp. 473-480. ACM (2012)
a117. Scaffidi, C., Myers, B., Shaw, M.: Tool support for data validation by end-user programmers. In Proc. of: International Conference on Software Engineering, pp. 867-870. ACM (2008)
a118. Freeman, D., Balakrishnan, R.: Tangible actions. International Conference on Interactive Tabletops and Surfaces. ACM, Kobe, Japan (2011) 87-96
a119. Tschudnowsky, A., Pietschmann, S., Niederhausen, M., Hertel, M., Gaedke, M.: From Choreographed to Hybrid User Interface Mashups: A Generic Transformation Approach. In: Casteleyn, S., Rossi, G., Winckler, M. (eds.) Web Engineering: 14th International Conference, ICWE 2014, Toulouse, France, July 1-4, 2014. Proceedings, pp. 145-162. Springer International Publishing, Cham (2014)
a120. Tsandilas, T., Letondal, C., Mackay, W.E.: Musink: Composing music through augmented drawing. In Proc. of: International Conference Extended Abstracts on Human Factors in Computing Systems, pp. 819-828. ACM (2009)
a121. Edgcomb, A., Vahid, F.: MNFL: The monitoring and notification flow language for assistive monitoring. In Proc. of: International Health Informatics Symposium, pp. 191-200. ACM (2012)
a122. Xie, B., Shabir, I., Abelson, H.: Measuring the usability and capability of App Inventor to create mobile applications. In Proc. of: International Workshop on Programming for Mobile and Touch, pp. 1-8. Association for Computing Machinery, Inc (2015)
a123. Verou, L., Zhang, A.X., Karger, D.R.: Mavo: Creating interactive data-driven web applications by authoring HTML. In Proc. of: Symposium on User Interface Software and Technology, pp. 483-496. Association for Computing Machinery, Inc (2016)
a124. Bolchini, D., Garzotto, F., Paolini, P.: Investigating success factors for hypermedia develoopment tools. In Proc. of: Conference on Hypertext and Hypermedia, pp. 187-191. ACM (2008)
a125. Wilson, A., Burnett, M., Beckwith, L., Granatir, O., Casburn, L., Cook, C., Durham, M., Rothermel, G.: Harnessing curiosity to increase correctness in end-user programming. In Proc. of: Conference on Human Factors in Computing Systems, pp. 305-312. ACM (2003)
a126. Borges, C.R., Macías, J.A.: Feasible database querying using a visual end-user approach. In Proc. of: Symposium on Engineering Interactive Computing Systems, pp. 187-192. ACM (2010)
a127. Bellucci, A., Zarraonandia, T., Díaz, P., Aedo, I.: End-user prototyping of cross-reality environments. In Proc. of: Conference on Tangible, Embedded, and Embodied Interaction, pp. 173-182. Association for Computing Machinery, Inc (2017)
a128. Nestler, T., Namoun, A., Schill, A.: End-user development of service-based interactive web applications at the presentation layer. In Proc. of: Symposium on Engineering Interactive Computing Systems, pp. 197-206. ACM (2011)

a129. Kim, J.E., Fan, X., Mosse, D.: Empowering end users for social internet of things. In Proc. of: International Conference on Internet-of-Things Design and Implementation, pp. 71-82. Association for Computing Machinery, Inc (2017)
a130. Koitz, R., Slany, W.: Empirical comparison of visual to hybrid formula manipulation in educational programming languages for teenagers. In Proc. of: Workshop on Evaluation and Usability of Programming Languages and Tools, pp. 21-30. Association for Computing Machinery, Inc (2014)
a131. Cockburn, A., Bryant, A.: Do it this way: equal opportunity programming for kids. In Proc. of: Australian Conference on Computer-Human Interaction, pp. 246-251. IEEE (1996)
a132. Mehandjiev, N., Namoun, A., Lécué, F., Wajid, U., Kleanthous, G.: End Users Developing Mashups. In: Bouguettaya, A., Sheng, Q.Z., Daniel, F. (eds.) Web Services Foundations, pp. 709-736. Springer New York, New York, NY (2014)
a133. Vuorimaa, P., Laine, M., Litvinova, E., Shestakov, D.: Leveraging declarative languages in web application development. World Wide Web, 19(4), 519-543 (2016). doi: 10.1007/s11280-015-0339-z
a134. Laga, N., Bertin, E., Crespi, N., Bedini, I., Molina, B., Zhao, Z.: A flexible service selection for executing virtual services. World Wide Web, 16(3), 219-245 (2013). doi: 10.1007/s11280-012-0184-2
a135. Toomim, M., Drucker, S.M., Dontcheva, M., Rahimi, A., Thomson, B., Landay, J.A.: Attaching UI enhancements to websites with end users. In Proc. of: International Conference Extended Abstracts on Human Factors in Computing Systems, pp. 1859-1868. ACM (2009)
a136. Dey, A.K., Hamid, R., Beckmann, C., Li, I., Hsu, D.: A CAPpella: Programming by demonstration of context-aware applications. In Proc. of: Conference on Human Factors in Computing Systems - Proceedings, pp. 33-40. ACM (2004)
a137. Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M.: Enabling End Users to Proactively Tailor Underspecified, Human-Centric Business Processes: "Programming by Example" of Weakly-Structured Process Models. In: Filipe, J., Cordeiro, J. (eds.) Enterprise Information Systems: 10th International Conference, ICEIS 2008, Barcelona, Spain, June 12-16, 2008, Revised Selected Papers, pp. 307-320. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
a138. Lizcano, D., Alonso, F., Soriano, J., López, G.: Web-centred end-user component modelling. Future Generation Computer Systems, 54, 16-40 (2016). doi: 10.1016/j.future.2015.07.002
a139. Danado, J., Davies, M., Ricca, P., Fensel, A.: An Authoring Tool for User Generated Mobile Services. In: Berre, A.J., Gómez-Pérez, A., Tutschku, K., Fensel, D. (eds.) Future Internet - FIS 2010: Third Future Internet Symposium, Berlin, Germany, September 20-22, 2010. Proceedings, pp. 118-127. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
a140. Gleichmann, M., Hasart, T., Bruder, I., Forbrig, P.: Supporting the Spreadsheet Idea for Interactive Database Applications. In: Forbrig, P., Paternó, F., Mark Pejtersen, A. (eds.) Human-Computer Interaction: Second IFIP TC 13 Symposium, HCIS 2010, Held as Part of WCC 2010, Brisbane, Australia, September 20-23, 2010. Proceedings, pp. 122-132. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
a141. Hafidh, B., Al Osman, H., Arteaga-Falconi, J.S., Dong, H., El Saddik, A.: SITE: The simple internet of things enabler for smart homes. IEEE Access, 5, 2034-2049 (2017). doi: 10.1109/ACCESS.2017.2653079
a142. Coutaz, J., Crowley, J.L.: A First-Person Experience with End-User Development for Smart Homes. IEEE Pervasive Computing, 15(2), 26-39 (2016). doi: 10.1109/MPRV.2016.24
a143. Bellucci, A., Vianello, A., Florack, Y., Jacucci, G.: Supporting the Serendipitous Use of Domestic Technologies. IEEE Pervasive Computing, 15(2), 16-25 (2016). doi: 10.1109/MPRV.2016.26
a144. Ruthruff, J.R., Burnett, M., Rothermel, G.: Interactive fault localization techniques in a spreadsheet environment. IEEE Transactions on Software Engineering, 32(4), 213-239 (2006). doi: 10.1109/TSE.2006.37
a145. de Lima, J.P.C., Carlos, L.M., Schardosim Simão, J.P., Pereira, J., Mafra, P.M., da Silva, J.B.: Design and implementation of a remote lab for teaching programming and robotics. IFAC-PapersOnLine, 49(30), 86-91 (2016). doi: 10.1016/j.ifacol.2016.11.133
a146. Saldivar, J., Vairetti, C., Rodríguez, C., Daniel, F., Casati, F., Alarcón, R.: Analysis and improvement of business process models using spreadsheets. Information Systems, 57, 1-19 (2016). doi: 10.1016/j.is.2015.10.012
a147. Dörner, C., Yetim, F., Pipek, V., Wulf, V.: Supporting business process experts in tailoring business processes. Interacting with Computers, 23(3), 226-238 (2011). doi: 10.1016/j.intcom.2011.03.001
a148. Macías, J.A., Castells, P.: Providing end-user facilities to simplify ontology-driven web application authoring. Interacting with Computers, 19(4), 563-585 (2007). doi: 10.1016/j.intcom.2007.01.006
a149. Macías, J.A., Paternò, F.: Customization of Web applications through an intelligent environment exploiting logical interface descriptions. Interacting with Computers, 20(1), 29-47 (2008). doi: 10.1016/j.intcom.2007.07.007
a150. Firmenich, S., Rossi, G., Winckler, M., Palanque, P.: An approach for supporting distributed user interface orchestration over the Web. International Journal of Human Computer Studies, 72(1), 53-76 (2014). doi: 10.1016/j.ijhcs.2013.08.014
a151. Ghiani, G., Paternò, F., Spano, L.D., Pintori, G.: An environment for End-User Development of Web mashups. International Journal of Human Computer Studies, 87, 38-64 (2016). doi: 10.1016/j.ijhcs.2015.10.008
a152. Martín, D., Alcarria, R., Sánchez-Picot, Á., Robles, T.: An Ambient Intelligence Framework for End-User Service Provisioning in a Hospital Pharmacy: a Case Study. Journal of Medical Systems, 39(10), 116 (2015). doi: 10.1007/s10916-015-0298-z
a153. Zhu, N., Grundy, J., Hosking, J., Liu, N., Cao, S., Mehra, A.: Pounamu: A meta-tool for exploratory domain-specific visual language tool development. Journal of Systems and Software, 80(8), 1390-1407 (2007). doi: 10.1016/j.jss.2006.10.028
a154. Lizcano, D., Soriano, J., López, G., Gutiérrez, J.J.: Automatic verification and validation wizard in web-centred end-user software engineering. Journal of Systems and Software, 125, 47-67 (2017). doi: 10.1016/j.jss.2016.11.025
a155. Neris, V.P.A., Baranauskas, M.C.C.: Designing tailorable software systems with the users' participation. Journal of the Brazilian Computer Society, 18(3), 213-227 (2012). doi: 10.1007/s13173-012-0070-x
a156. Ardito, C., Costabile, F.M., Desolda, G., Lanzilotti, R., Matera, M., Piccinno, A., Picozzi, M.: User-driven visual composition of service-based interactive spaces. Journal of Visual Languages and Computing, 25(4), 278-296 (2014). doi: 10.1016/j.jvlc.2014.01.003
a157. Drey, Z., Consel, C.: Taxonomy-driven prototyping of home automation applications: A novice-programmer visual language and its evaluation. Journal of Visual Languages and Computing, 23(6), 311-326 (2012). doi: 10.1016/j.jvlc.2012.07.002
a158. Francese, R., Risi, M., Tortora, G.: Iconic languages: Towards end-user programming of mobile applications. Journal of Visual Languages & Computing, 38(Supplement C), 1-8 (2017). doi: 10.1016/j.jvlc.2016.10.009

a159. Turchi, T., Malizia, A., Dix, A.: TAPAS: A tangible End-User Development tool supporting the repurposing of Pervasive Displays. Journal of Visual Languages and Computing, 39, 66-77 (2017). doi: 10.1016/j.jvlc.2016.11.002
a160. Blackwell, A.F.: Palimpsest: A layered language for exploratory image processing. Journal of Visual Languages and Computing, 25(5), 545-571 (2014). doi: 10.1016/j.jvlc.2014.07.001
a161. Chang, K.S.P., Myers, B.A.: Gneiss: spreadsheet programming using structured web service data. Journal of Visual Languages and Computing, 39, 41-50 (2017). doi: 10.1016/j.jvlc.2016.07.004
a162. Neumann, C., Metoyer, R.A., Burnett, M.: End-user strategy programming. Journal of Visual Languages & Computing, 20(1), 16-29 (2009). doi: 10.1016/j.jvlc.2008.04.005
a163. Castelli, V., Bergman, L., Lau, T., Oblinger, D.: Sheepdog, parallel collaborative programming-by-demonstration. Knowledge-Based Systems, 23(2), 94-109 (2010). doi: 10.1016/j.knosys.2009.06.008
a164. Valderas, P., Torres, V., Mansanet, I., Pelechano, V.: A mobile-based solution for supporting end-users in the composition of services. Multimedia Tools and Applications, 76(15), 16315-16345 (2017). doi: 10.1007/s11042-016-3910-4
a165. Zhong, Y., Liu, C.: A domain-oriented end-user design environment for generating interactive 3D virtual chemistry experiments. Multimedia Tools and Applications, 72(3), 2895-2924 (2014). doi: 10.1007/s11042-013-1554-1

Appendix B: Paper classification

Paper	D1 - Approach	D2 - Technique	D3 - Phase	D4 - Application domain	D5 - Target use	D6 - Target users	D7 - Empirical validation
a1	EUP	Component-based	Use	Games and Entertainment	Utilitarian	Generic users	Formal experiment with users
a2	EUD	Component-based	Use	Interaction design	Utilitarian	Generic users	Formal experiment with users
a3	EUD	Component-based	Design	Business and data management	Utilitarian	Domain experts	Informal experiment with users
a4	EUD	Component-based	Design	Games and Entertainment	Utilitarian	Domain experts	Formal experiment with users
a5	EUD	Component-based	Design	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a6	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a7	EUD	Spreadsheet-based	Design	Business and data management	Utilitarian	Domain experts	Informal experiment with users
a8	EUP	Component-based	Design	Games and Entertainment	Utilitarian	Generic users	Pilot study
a9	EUD	Model-based	Design	Interaction design	Utilitarian	Domain experts	Formal experiment with users
a10	EUP	Component-based	Design	Smart objects and environments	Utilitarian	Domain experts	Formal experiment with users
a11	EUD	Component-based / Template-based	Use	Interaction design	Utilitarian	Domain experts	Formal experiment with users
a12	EUD	Component-based / Wizard-based	Design	Healthcare and wellness	Utilitarian	Domain experts	Formal experiment with users
a13	EUD	Spreadsheet-based	Design	Business and data management	Utilitarian	Generic users	Formal experiment with users
a14	EUD	Component-based	Design	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a15	EUD	Component-based	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a16	EUD	Component-based	Use	Healthcare and wellness	Utilitarian	Domain experts	Formal experiment with users
a17	EUP	Workflow and dataflow diagrams	Use	Web applications and mashups	Utilitarian	Domain experts	Informal experiment with users
a18	EUD	Component-based	Use	Education and teaching	Educational	Domain experts	Formal experiment with users
a19	EUD	Wizard-based	Design	Education and teaching	Educational	Students	Formal experiment with users
a20	EUD	Component-based	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a21	EUD	Component-based / Rule-based	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a22	EUD	Rule-based	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a23	EUD	Programming by demonstration/example	Use	Education and teaching	Utilitarian	Generic users	Formal experiment with users
a24	EUD	Rule-based	Use	Mobile applications	Utilitarian	Generic users	Formal experiment with users
a25	EUD	Component-based	Design	Smart objects and environments	Utilitarian	Domain experts / Generic users	Formal experiment with users
a26	EUD	Component-based	Use	Business and data management	Utilitarian	Domain experts	Informal experiment with users
a27	EUD	Component-based	Design	Mobile applications	Utilitarian	Generic users	Formal experiment with users
a28	EUD	Component-based	Design	Games and Entertainment	Playful	Generic users	Formal experiment with users
a29	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a30	EUP	Model-based	Use	Web applications and mashups	Utilitarian	Generic users	Informal experiment with users
a31	EUP	Spreadsheet-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a32	EUP	Component-based	Design	Games and Entertainment	Playful	Generic users	Formal experiment with users
a33	EUP	Programming by demonstration/example	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a34	EUP	Component-based	Design	Web applications and mashups	Utilitarian	Domain experts	Informal experiment with users
a35	EUP	Natural language	Design	Robotics	Assistive technology	Generic users	Formal experiment with users
a36	EUP	Spreadsheet-based	Use	Business and data management	Utilitarian	Generic users	Informal experiment with users
a37	EUP	Component-based	Design	Education and teaching	Educational	Domain experts	Formal experiment with users

a38	EUP	Component-based	Design	Healthcare and wellness	Assistive technology	Domain experts	Formal experiment with users
a39	EUD	Wizard-based	Design	Web applications and mashups	Utilitarian	Domain experts	Formal experiment with users
a40	EUD	Wizard-based / Natural language	Design	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a41	EUP	Wizard-based	Design	Mobile applications	Utilitarian	Domain experts	Formal experiment with users
a42	EUD	Component-based	Design	Education and teaching	Educational	Domain experts	Formal experiment with users
a43	EUD	Workflow and dataflow diagrams	Use	Business and data management	Utilitarian	Domain experts	Pilot study
a44	EUP	Rule-based	Use	Smart objects and environments	Utilitarian / Personal	Generic users	Formal experiment with users
a45	EUSE	Assertion-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a46	EUSE	Component-based	Design	Web applications and mashups	Utilitarian	Domain experts	Formal experiment with users
a47	EUP	Digital sketching	Design	Interaction design	Educational	Domain experts	Informal experiment with users
a48	EUP	Component-based / Natural language	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a49	EUD	Component-based	Design	Web applications and mashups	Utilitarian / Personal	Generic users	Formal experiment with users
a50	EUP	Component-based	Use	Web applications and mashups	Utilitarian	Domain experts	Formal experiment with users
a51	EUP	Component-based	Design	Games and Entertainment	Utilitarian	Generic users	Formal experiment with users
a52	EUSE	Component-based	Design	Smart objects and environments	Utilitarian	Domain experts	Formal experiment with users
a53	EUD	Component-based	Use	Smart objects and environments	Utilitarian / Personal	Generic users	Formal experiment with users
a54	EUD	Wizard-based	Design	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a55	EUD	Component-based	Design	Web applications and mashups	Utilitarian / Personal	Generic users	Formal experiment with users
a56	EUD	Rule-based	Design	Games and Entertainment	Utilitarian	Domain experts	Formal experiment with users
a57	EUD	Rule-based	Design / Use	Smart objects and environments	Utilitarian	Domain experts	Formal experiment with users
a58	EUD	Component-based	Design	Education and teaching	Educational	Domain experts	Formal experiment with users
a59	EUP	Component-based	Design	Education and teaching	Educational	Students	Formal experiment with users
a60	EUD	Wizard-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a61	EUP	Component-based / Rule-based	Design / Use	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a62	EUSE	Component-based / Workflow and dataflow diagrams	Design	Business and data management	Utilitarian	Domain experts	Informal experiment with users
a63	EUP	Natural language	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a64	EUD	Model-based	Design	Healthcare and wellness	Educational / Assistive technology	Domain experts	Informal experiment with users
a65	EUP	Component-based / Programming by demonstration/example	Design	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a66	EUP	Component-based	Design / Use	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a67	EUD	Model-based	Design	Interaction design	Utilitarian	Generic users	Informal experiment with users
a68	EUP	Programming by demonstration/example	Design	Mobile applications	Utilitarian	Generic users	Informal experiment with users
a69	EUP	Component-based	Design	Mobile applications	Educational	Domain experts	Pilot study
a70	EUP	Programming by demonstration/example	Design	Web applications and mashups	Utilitarian / Personal	Generic users	Informal experiment with users
a71	EUP	Programming by demonstration/example	Use	Business and data management	Utilitarian	Domain experts	Informal experiment with users
a72	EUD	Programming by demonstration/example	Use	Mobile applications	Personal	Generic users	Formal experiment with users
a73	EUSE	Assertion-based	Use	Web applications and mashups	Utilitarian	Domain experts	Formal experiment with users
a74	EUP	Spreadsheet-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a75	EUP	Natural language	Use	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a76	EUP	Natural language / Rule-based	Use	Business and data management	Personal	Generic users	Formal experiment with users
a77	EUP	Model-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a78	EUP	Text-based	Design	Education and teaching	Educational	Students	Formal experiment with users

a79	EUD	Component-based	Design	Smart objects and environments	Playful	Generic users	Pilot study
a80	EUP	Rule-based / Programming by demonstration/example	Use	Smart objects and environments	Playful / Personal	Generic users	Formal experiment with users
a81	EUP	Natural language / Annotation-based	Use	Education and teaching	Educational	Students	Formal experiment with users
a82	EUP	Component-based / Natural language	Use	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a83	EUP	Model-based	Use	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a84	EUP	Component-based / Natural language	Use	Business and data management	Utilitarian / Personal	Generic users	Formal experiment with users
a85	EUP	Component-based / Workflow and dataflow diagrams	Design	Healthcare and wellness	Utilitarian / Personal	Domain experts	Formal experiment with users
a86	EUP	Spreadsheet-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a87	EUP	Rule-based	Design	Smart objects and environments	Playful / Educational	Generic users	Formal experiment with users
a88	EUP	Component-based / Programming by demonstration/example	Design	Robotics	Utilitarian	Domain experts / Generic users	Formal experiment with users
a89	EUD	Programming by demonstration/example	Use	Mobile applications	Utilitarian / Personal	Generic users	Formal experiment with users
a90	EUD	Wizard-based	Use	Smart objects and environments	Playful / Personal	Generic users	Formal experiment with users
a91	Hybrid	Component-based	Design	Games and Entertainment	Utilitarian	Domain experts	Informal experiment with users
a92	EUD	Rule-based	Use	Smart objects and environments	Playful / Personal	Generic users	Formal experiment with users
a93	EUP	Component-based / Template-based	Use	Robotics	Utilitarian	Domain experts	Formal experiment with users
a94	EUD	Annotation-based / Template-based	Use	Business and data management	Utilitarian / Personal	Domain experts	Informal experiment with users
a95	EUP	Template-based	Use	Business and data management	Personal	Generic users	Informal experiment with users
a96	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Domain experts	Formal experiment with users
a97	EUP	Workflow and dataflow diagrams	Use	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a98	EUP	Spreadsheet-based	Use	Business and data management	Utilitarian / Personal	Generic users	Formal experiment with users
a99	EUP	Spreadsheet-based	Use	Business and data management	Utilitarian / Personal	Generic users	Informal experiment with users
a100	EUD	Wizard-based	Use	Business and data management	Utilitarian	Domain experts	Informal experiment with users
a101	EUP	Digital sketching	Design	Games and Entertainment	Utilitarian	Domain experts	Informal experiment with users
a102	EUP	Component-based	Use	Healthcare and wellness	Utilitarian / Personal	Domain experts	Formal experiment with users
a103	EUD	Component-based	Design	Healthcare and wellness	Utilitarian	Domain experts	Formal experiment with users
a104	EUSE	Template-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a105	EUD	Template-based	Design	Games and Entertainment	Utilitarian	Domain experts	Formal experiment with users
a106	EUP	Component-based	Design	Games and Entertainment	Utilitarian	Domain experts	Formal experiment with users
a107	EUD	Component-based / Template-based	Design	Healthcare and wellness	Utilitarian	Domain experts	Formal experiment with users
a108	Hybrid	Spreadsheet-based	Use	Healthcare and wellness	Educational	Students	Informal experiment with users
a109	EUP	Workflow and dataflow diagrams	Use	Smart objects and environments	Utilitarian / Personal	Generic users	Formal experiment with users
a110	EUD	Component-based / Template-based	Design	Web applications and mashups	Utilitarian	Domain experts	Informal experiment with users
a111	EUD	Component-based	Design / Use	Web applications and mashups	Utilitarian / Personal	Generic users	Formal experiment with users
a112	EUD	Component-based	Design	Games and Entertainment	Playful	Generic users	Formal experiment with users
a113	EUD	Digital sketching / Gesture-based	Design	Games and Entertainment	Utilitarian / Playful	Domain experts	Formal experiment with users
a114	EUD	Wizard-based	Design	Web applications and mashups	Utilitarian	Domain experts	Formal experiment with users
a115	EUD	Rule-based	Design	Healthcare and wellness	Utilitarian	Domain experts	Formal experiment with users
a116	EUD	Annotation-based / Template-based	Use	Business and data management	Utilitarian	Domain experts	Formal experiment with users
a117	EUP	Wizard-based	Design	Business and data management	Utilitarian	Students	Formal experiment with users
a118	EUP	Programming by demonstration/example	Use	Games and Entertainment	Utilitarian / Playful	Generic users	Informal experiment with users
a119	EUD	Component-based	Design	Web applications and mashups	Utilitarian / Playful	Generic users	Formal experiment with users

a120	EUP	Gesture-based	Design	Games and Entertainment	Utilitarian	Domain experts	Formal experiment with users
a121	EUP	Component-based	Use	Healthcare and wellness	Utilitarian	Generic users	Formal experiment with users
a122	EUP	Component-based	Design	Mobile applications	Educational	Generic users	Informal experiment with users
a123	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a124	EUD	Component-based	Design	Education and teaching	Educational	Students	Formal experiment with users
a125	EUSE	Assertion-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a126	EUD	Programming by demonstration/example	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a127	EUD	Component-based	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a128	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a129	EUP	Rule-based	Use	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a130	EUP	Component-based / Text-based	Design	Education and teaching	Educational	Generic users	Formal experiment with users
a131	EUD	Component-based / Programming by demonstration/example / Text-based	Use	Education and teaching	Educational	Generic users	Formal experiment with users
a132	EUD	Component-based	Use	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a133	EUD	Component-based / Text-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a134	EUD	Wizard-based	Design	Web applications and mashups	Utilitarian	Generic users	Informal experiment with users
a135	EUP	Programming by demonstration/example	Design	Web applications and mashups	Utilitarian	Domain experts / Generic users	Formal experiment with users
a136	EUP	Programming by demonstration/example	Use	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a137	EUD	Programming by demonstration/example	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a138	EUP	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a139	EUD	Template-based	Design	Mobile applications	Utilitarian	Domain experts / Generic users	Formal experiment with users
a140	EUD	Spreadsheet-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a141	EUD	Rule-based	Use	Smart objects and environments	Utilitarian	Domain experts / Generic users	Formal experiment with users
a142	EUD	Rule-based	Use	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a143	EUP	Rule-based	Use	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a144	EUSE	Spreadsheet-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a145	EUP	Component-based	Use	Robotics	Educational	Generic users	Formal experiment with users
a146	EUD	Spreadsheet-based	Use	Business and data management	Utilitarian	Generic users	Formal experiment with users
a147	EUD	Component-based	Design	Business and data management	Utilitarian	Generic users	Formal experiment with users
a148	EUD	Programming by demonstration/example	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a149	EUD	Programming by demonstration/example	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a150	EUP	Annotation-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a151	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a152	EUD	Rule-based	Design	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a153	EUD	Component-based	Design	Interaction design	Utilitarian	Domain experts	Formal experiment with users
a154	Hybrid	Component-based	Design	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a155	EUP	Digital sketching / Rule-based	Design	Interaction design	Utilitarian	Generic users	Formal experiment with users
a156	EUD	Component-based	Design	Web applications and mashups	Utilitarian	Domain experts / Generic users	Formal experiment with users
a157	EUD	Component-based	Use	Smart objects and environments	Utilitarian	Generic users	Formal experiment with users
a158	EUP	Component-based	Design	Mobile applications	Utilitarian	Generic users	Formal experiment with users

a159	EUD	Workflow and dataflow diagrams	Use	Games and Entertainment	Utilitarian	Generic users	Formal experiment with users
a160	EUP	Component-based	Use	Games and Entertainment	Utilitarian	Generic users	Formal experiment with users
a161	EUP	Component-based / Spreadsheet-based	Use	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a162	EUP	Component-based / Rule-based	Use	Healthcare and wellness	Utilitarian	Generic users	Informal experiment with users
a163	EUP	Programming by demonstration/example	Design	Business and data management	Utilitarian	Generic users	Formal experiment with users
a164	EUD	Component-based	Use	Web applications and mashups	Utilitarian	Generic users	Formal experiment with users
a165	EUD	Component-based	Design	Education and teaching	Educational	Domain experts	Formal experiment with users

Authors' Biographies

Barbara Rita Barricelli

Barbara Rita Barricelli is currently Assistant Professor at the Department of Computer Science of the University of Milano (Italy). Her research is focused on addressing the communication problems between different communities of practice by means of interactive information systems built taking into account internationalization and localization needs of the end users. She applied computer semiotics theories and semiotic engineering methods for designing and developing internationalized interactive systems, localized to users' culture, gender, role played in specific contexts/domains and to the digital platform used to access the systems. She is Chair of IFIP TC13 Working Group 6 Human Work Interaction Design.

Fabio Cassano

Fabio Cassano is currently a PhD Student at the Computer Science Department of the University of Bari "Aldo Moro" (Italy) and member of the Interaction, Visualization, Usability & UX (IVU) Lab. He got at the Polytechnic University of Bari the Bachelor Degree in Computer Engineering in 2013 and the Master Science Degree in Computer Engineering with honors, in 2015, with a thesis in Human-Machine Interaction. His research interests focus on End-User Development, Ambient Intelligence, Internet of Things, Home Assistance, Active Aging. He is member of SIGCHI Italy (the Italian Chapter of ACM SIGCHI) and EUSSET (EUropean Society for Socially-Embedded Technologies).

Daniela Fogli

Daniela Fogli is currently Associate Professor at the Department of Information Engineering of the University of Brescia (Italy). She received the Laurea degree in Computer Science from the University of Bologna (Italy), in 1994 and the PhD degree in Information Engineering from the University of Brescia, in 1998. From 1998 to 2000, she worked at the Joint Research Centre of the European Commission. Her research interests include meta-design, end-user development, web usability and accessibility, decision support systems. She is member of ACM (Association of Computing Machinery) and SIGCHI Italy (the Italian Chapter of ACM's Special Interest Group on Computer-Human Interaction).

Antonio Piccinno

Antonio Piccinno is currently Assistant Professor at the Computer Science Department of the University of Bari "Aldo Moro" (Italy). He received a Laurea Degree with full marks and honors in Computer Science from the University of Bari. In 2005 he got the PhD in Computer Science from the University of Bari. His research interests focus on End-User Development, visual language theories, human-computer interaction, multimedia and multimodal interaction, web interfaces. He is in the Steering Committee of the International Symposium on End-User Development and served as Program Co-Chair in the third edition (IS-EUD 2011) held in Torre Canne (Brindisi), Italy. He is member of ACM (Association of Computing Machinery), ACM SIGCHI (ACM's Special Interest Group on Computer-Human Interaction) and SIGCHI Italy (the Italian Chapter of ACM SIGCHI).