Paloma Díaz, Ignacio Aedo, Combining Software Engineering and Design Thinking Practices in the Ideation Process of Augmented Digital Experiences, *Interacting with Computers*, Volume 32, Issue 3, May 2020, Pages 279–295.

DOI: 10.1093/iwc/iwaa020

Combining Software Engineering and Design Thinking Practices in the Ideation Process of Augmented Digital Experiences

PALOMA DÍAZ* AND IGNACIO AEDO

Computer Science Department, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911, Leganés, Spain

**Corresponding author: pdp@inf.uc3m.es*

Designing augmented and tangible experiences that intertwine human practices and expectations, interaction spaces and complex digital artifacts is a complex and multifaceted task that relies upon iterative and multidisciplinary ideation processes. Design thinking techniques have been traditionally used in ideation of such digital artifacts. In this paper, we posit that integrating some software engineering practices can improve ideation by providing a structure to the process and helping to build a shared and permanently documented design rationale. It is not a matter of software engineering *versus* design thinking but a question of developing a holistic understanding of technological development where discipline and creativity, rationality and emotions and quality centered and people centered coexist. Based on this assumption, we conceived a software tool called CoDICE that offers a virtual co-design space where augmented digital experiences are documented and analyzed in a shared and distributed way. The paper discusses how CoDICE contributes to alleviate some problems of co-design events including the need to support multiple co-design spaces, make explicit the co-design process and its goals, support documentation, justify design decisions, explore multiple ideas and generate a shared representation of the outcomes. Two scenarios are used to illustrate the tool utility: short-term co-design workshops in which the tool enabled multidisciplinary teams of novice designers to explore and structure their ideas and a long-term co-design project where the tool facilitated traceability, documentation, the reuse of design components and the shared elaboration of the design rationale and evolution of the deployed technologies.

HIGHLIGHTS

- A distributed virtual space to document and elaborate further design constituents
- An approach to integrate software engineering practices without compromising creativity
- Some challenges in co-design events with multidisciplinary and novice participants are addressed.
- A number of design boundary objects push reflection upon technological interventions from different perspectives including the scenarios of use, goals or potential users.
- Utility in long-term project and short-term events has been analyzed.

Keywords: systems and tools for design; software engineering; co-design

1. INTRODUCTION

Ubiquitous, augmented and social computing are creating a rich technological substratum to improve human experiences in all walks of life, from education, to work, leisure, healthcare and any other kind of mundane activity. Designing such experiences implies intertwining human practices and expectations, spaces and complex digital artifacts into cohesive interaction ecosystems in which three agents (namely people, technologies and spaces) and their interrelationships should be taken into account (Terrenghi *et al.*, 2009). The design focus is moved from technology deployment to devise how the artificial can augment our capacities as individuals or members of a group or community (Buchanan, 1992; Simon, 1996; Kaptelinin and Nardi, 2006; Zimmerman *et al.*, 2007; Binder *et al.*, 2011). This kind of creative design often implies facing a wicked problem that does not have a unique solution nor a linear way to solve it, whether because requirements are unclear or evolve over time or because stakeholders have competing values and priorities (Rittel and Weber, 1973; Nelson and Stolterman 2003; Binder *et al.*, 2011). The first step of this process is usually the early ideation stage, where designers need to frame the problem, for which they have to apply a reflective practice (Schön, 1983) and work closely with different types of professionals, including designers, experts in the domain of application and stakeholders (Sanders and Stappers, 2008) to integrate different matrixes of thought in a social creativity process (Fischer *et al.*, 2005; War and O'Neill, 2005). In this context, a design turns into co-design, that is, a participatory, long-term, open, exploratory and iterative process (Sanders and Westerlund, 2011; Binder *et al.*, 2011; Chandrasegaran *et al.*, 2013) where creativity, deliberation and argumentation are essential to provide a rationale on the design decisions (Buchanan, 1992). Co-design teams are made up of different kinds of participants who usually apply a design thinking approach and apply generative techniques like drawing, crafting, sketching or performing to freely shape the problem and ideate solutions. From the various interpretations on what design thinking is, in this paper, we conceive it as a *designerly way of thinking and doing* aimed at generating a value but without having clearly determined what is going to be created and how (Dorst, 2011) and that relies upon generative techniques to ideate and design innovative products (Sanders and Stappers, 2008).

The heterogeneity of participants in iterative co-design processes brings richness of perspectives, experiences and knowledge to the table as well as some challenges, including the need to make evident the purpose of each design task to all participants, to engage all of them in broad exploration processes that go beyond the first idea, to generate a shared understanding of problems and solutions and to facilitate the convergence to specific ways of action (Adams *et al.*, 2011; Sanders and Westerlund, 2011; Garde and van der Voort, 2012). We advocate that interspersing some software engineering practices, like documentation based on conceptual models of the domain, validation or traceability, can contribute to face some of such challenges as far the flexibility and richness of expression are not compromised (Forlizzi and Battarbee, 2004; Robertson and Radcliffe, 2009; Chandrasegaran *et al.*, 2013; Bernal *et al.*, 2015). What is required is a holistic process that takes profit from the flexibility and expressivity of generative methods and, at the same time, provides means to *softly structure and document* the process to facilitate communication about why and how artifacts were created and how they evolved.

In this paper, we discuss how we addressed such challenge using a software tool called CoDICE (CoDesigning digital Cultural Encounters) that supports the documentation of co-design projects related with developing digitally augmented experiences with physical objects. In this co-design activities, the voices of domain experts need to be integrated, but sometimes they do not participate since the techniques used are not so evident, they do not have time enough to explore ideas and they do not understand why techniques are applied or the goal and utility of the design outcomes (such as journeys, personas, scenarios . . .). Our goal is to provide multidisciplinary co-design teams with a number of design constituents that act as *meaningful holders* to collect all the design outcomes into relevant entities in augmented computing (in our case, the physical space, the users, the scenarios and the technology). These *holders*, usually called design boundary objects (DBOs henceforth) in the literature (Binder *et al.*, 2011), have been designed iteratively to gather different perspectives, from technical issues, to non-rational criteria, emotions, behaviors, interpretations or known uses. DBOs are interrelated in an explicit and visual way according to a conceptual model of the domain. Moreover, the tool provides a soft structure that can help to clarify the design process to novice designers and push more experienced designers to revisit their ideas from different perspectives. CoDICE has been used in several co-design workshops to structure ideas and reuse components, revisit ideas and document the design rationale behind their development and evolution. In particular, two kinds of scenarios are described in this paper: short-term co-design workshops and a long-term co-design project. In the first scenario, participants appreciated being guided by the tool, having the possibility of connecting different components in a visual and direct way and being able to produce a unique document with all their ideas. In the second scenario, the tool facilitated traceability of a long-term process, the reuse of design outcomes and the shared elaboration of the design rationale and the evolution of the ideas.

The remaining of the paper starts by framing the context of this research. CoDICE tool is described in Section 3 and its evaluation is reported in Section 4. The tool contributions and its utility are analyzed in Section 5 by focusing on the way it deals with some of the limitations of co-design workshops and describing the two scenarios of use. Lessons learnt and future research are drawn in the conclusions.

2. CONTEXT AND RELATED WORKS

This research is rooted on the limitations of running effective co-design sessions that capture feedback from all types of participants and on the benefits of integrating some software engineering practices to alleviate some of that limitations.

2.1. Challenges to support effective co-design in multidisciplinary and long-term projects

Co-design, understood as a social creativity process among designers and other professional and non-professional participants, spans over the whole ideation, design and building of design artifacts (Sanders and Stappers, 2008). The richness of perspectives, experiences and knowledge is a key feature of co-design, but it also poses some challenges, particularly if co-design involves novice participants or it is embedded into long-term projects. In the remaining of this section, we review some of these challenges focusing on those that can be solved by integrating software tools as *thin infrastructures* (Binder *et al.*, 2011). Such *thin infrastructures* provide a technical support to partly deal with some of the challenges but do not guarantee per se the success of co-design activities, for which *thick infrastructures* made up of skilled co-design leaders and appropriate design games and techniques are required (Binder *et al.*, 2011). A comprehensive analysis of co-design challenges can be found at Warr and O'Neill (2005), Bergman *et al.* (2007), Sanders and Stappers (2008), Kleinsmann and Valkenburg (2008), Sanders and Westerlund (2011) and Garde and van der Voort (2012).

First at all, co-design happens in different spaces including the experienced physical space, the workspace and the future situation of use (Sanders and Westerlund, 2011). These spaces are not only physically different but also conceptual. The possibility of doing part of the co-design process in the actual physical space where the product is supposed to be deployed or in the context recreated by the product might help to better understand the scenarios of use. Other activities, like specifying or exploring ideas further, can be moved to design rooms. Moreover, participants might have different rhythms of working. Some might require more time to understand the problem or to further elaborate their initial thoughts in private spaces escaping from the *evaluation apprehension syndrome* that might deter them from participating actively in co-located meetings (Warr and O'Neill, 2005). From a technological perspective, synchronous and asynchronous virtual design spaces can support such long-term and reflective practice in a more effective way tearing down time and space barriers (Díaz *et al.*, 2015).

Other typical problem consists of getting stuck into the first idea and limiting thus the exploration of other alternatives (Sanders and Westerlund, 2011). This can be due to the lack of motivation of novice participants who might feel their ideas are not creative enough or will be judged useless by more experienced participants (Warr and O'Neill, 2005; Sanders and Westerlund, 2011). Technological co-design spaces can be used to promote the exploration of multiple ideas in shared and private spaces.

Yet another relevant challenge is reconciling the variety of skills, expectations and attitudes that heterogeneous groups of co-designers have. In a phenomenographic study across disciplines (Adams *et al.*, 2011), six situated meanings of design were identified, including evidence-based decision making, organized translation, personal synthesis, intentional progression, directed creative exploration and freedom. Each of these definitions entails a different attitude toward design, a different set of skills and conceptual tools to face design tasks as well as different expectations about the participation. Co-design should integrate all of these mindsets and goals into a common arena (Buchanan, 1992; Bergman *et al.*, 2007). For this to be possible co-design spaces have to provide means to enable multidisciplinary communication in order to build a common understanding of both, the problem and the potential courses of action, to reflect in action and on action (Schön, 1983; O'Neill *et al.*, 1999; Binder *et al.*, 2011). To enable non-experts to express ideas in a natural and spontaneous way, co-design processes usually rely upon generative techniques that make it possible to freely define concepts using any kind of media, using tangibles, acting and playing (Sanders *et al.*, 2010; Martin and Haningtong, 2012; Giaccardi *et al.*, 2012). To generate a shared representation of the design outcomes and their design rationale that can be understood by all participants, DBOs such as personas, scenarios or protoarchitectures have been proposed (Cooper, 1999; Carroll and Rosson, 2003; Bergman *et al.*, 2007). DBOs encapsulate design knowledge in a flexible and expressive way to accomodate the needs of different disciplines and they evolve over time as more knowledge on the design constituents is gained through reflection in and on action (Binder *et al.*, 2011).

However, DBOs and generative techniques might not be enough to support a co-design process that involves a variety of design tasks. The unrestricted nature of divergent thinking techniques used to explore open-ended questions do not provide a clear process flow and, hence, co-design is often characterized as a fuzzy and non-linear iterative process (Rhea, 2003; Sanders and Stapers, 2008; Binder *et al.*, 2011). This fuzziness makes some participants, particularly novice designers and users, unaware of the goal of each design task and how they relate with the other tasks (Kleinsmann and Valkenburg, 2008; Sanders and Westerlund, 2011; Garde and van der Voort, 2012). In this sense, there is a need to make explicit the purpose and utility of each task performed as well as the interrelations among them.

Other challenges are related with the capacity to support reflection on the actions performed not only while they are performed but also after and, therefore, a design rationale and product genetic evolution should be properly documented (Bodker, 2000; Binder *et al.*, 2011). As discussed in Kleinsmann *et al.* (2008), other problems that affect co-design

TABLE 1. Challenges to carry out efficient co-design events that inspired CoDICE development.

Challenge	Rationale
Support multiple co-design spaces	Design happens in different places and at different moments (Sanders and Westerlund, 2011)
Support the exploration of multiple ideas	Design is a social process of creativity (Warr and O'Neill, 2005; Sanders and Westerlund, 2011)
Generate a shared representation of the outcomes	Multidisciplinary co-design teams need means to communicate effectively (O'Neill *et al.*, 1999)
Make explicit the process, the goals of design outcome and their relationships	All co-designers need to undertand the purpose and value of their participation (Kleinsmann and Valkenburg, 2008; Sanders and Westerlund, 2011; Garde and van der Voort, 2012)
Support documentation and information processing	Design knowledge needs to be transferred in easy way (Kleinsmann and Valkenburg, 2008)
Justify the design decisions	Design decisions and product evolution need to be communicated to learn from the experience (Bødker, 2000)

efficacy include factors that depend (i) on each participant, such us the interest about the design task or the knowledge about it, (ii) on the project, including the efficiency of information processing or the quality of the documentation, and (iii) on the organization, such as the allocation of tasks and responsibilities.

Table 1 summarizes some key challenges of co-design that inspired the development of the tool here presented. The tool, called CoDICE, alleviates some of these issues by using technology and assuming some basic practices of design thinking and software engineering. We will revisit this Table when discussing CoDICE contributions.

2.2. The role of software engineering in a holistic co-design process

Theories on information systems design (Klein and Hirschheim, 1991; Hevner *et al.*, 2004; Gregor and Jones, 2007) paved the way to enlarge the perspective of software engineering as a discipline aimed at designing socio-technical systems. In this work, we root on the conception of Klein and Hirscheim of software development as a *social action* focused on deploying technology that meets a set of objectively defined quality criteria as well as on envisioning ways to change social forms and behaviors (Klein and Hirschheim, 1991). This definition embraces the principles of design science (Simon, 1996; Cross, 2001; Krippendorf, 2005; Zimmerman *et al.*, 2007) and, therefore, it provides a conceptual fulcrum to elaborate a symbiotic relationship between two apparently opposite practices: creative design and software engineering. By symbiotic relationship, we mean a holistic process in which both disciplines contribute and depend on each other to support a better process; it is not a matter of software engineering *versus* design thinking but of software *and* design thinking, an effort to reconcile discipline and creativity, rationality and emotions and quality centered and people centered.

As described by Max Weber, social actions are driven by rational and non-rational forces (Weber, 1991). Rational forces are related with objective criteria and principles such as efficacy, cost, reliability (formal rationality) or with the need to apply a specific method to reach the goals (substantive rationality). Rational forces are the primary focus of software engineering and are usually gathered through well-documented product and process requirements (Pressman, 2010). On the other side, non-rational forces include subjective, personal and sociocultural criteria like emotions, aesthetics or resonance. This kind of forces are the focus of design thinking practices who seek at exploring the '*quality without a name*' (Alexander, 1979) that makes technologies useful, valuable and enjoyable. A holistic process can take benefit from both approaches. Software engineering fails to provide on its own a solution using analytical processes and metrics that do not take into account the non-rational forces that make a product successful and enjoyable. Mathematical models and visual notations cannot be used to explore ideas and promote discussion of options with end users and co-designers (Luebbe *et al.*, 2010). Agile and lean software engineering methods help to speed up the development process but might not be enough to deal with innovative and open-ended design, since they fail to support the conceptualization level required to make explicit the design rationale behind products development (Luebbe *et al.*, 2010) and the reflective design required to generate innovative products. These gaps are filled by generative tools that enable expressing ideas using richer and less structured communication options like a drawing, collaging (Sanders *et al.*, 2010) or even performing scenarios of use (Giaccardi *et al.*, 2012). However, some classical software engineering practices might provide advantages in long-term and distributed projects as well as for novice designers. For example, they encourage co-designers to apply a more disciplined, repeatable and measurable process where all the design tasks and their interrelations are clearly defined. Other additional advantages that can be borrowed from software engineering include the generation of a complete documentation of the co-design events, and the traceability of the outcomes that contributes to facilitate the artifacts evolution and maintenance (Pohl, 2010; Pressman, 2010; vom Brocke *et al.*, 2017).

2.3. Related works

There are many software tools that support specific design thinking activities such as collaborative annotation tools like those listed in Juntoo (2019) that are often used by experienced designers who are quite focused and proficient in their work to share and comment their design outcomes. Compared to them, CoDICE provides a shared design space that is structured according to the semantics of the domain of application, augmented computing. As discussed above, we posit that adding a more systematic and structured approach typical from software engineering has benefits both for novice designers who are guided through a process that is fuzzy and unstructured as well as to establish some discipline in multidisciplinary and long-term projects. Next paragraphs review tools that combine software engineering and design thinking practices.

In Luebbe *et al.* (2010), the authors transform the constructs of a software engineering method into tangible pieces that can be used along with physical elements such as post-its. In this way, they provide a flexible working space where mixed groups of software engineers, designers and users can discuss ideas in an expressive way but also including structured concepts. Similar works are focused on integrating software tools to produce typical outomes like affinity diagrams, including Klemmer *et al.* (2001), Harboe *et al.* (2013), Widjaja and Takahashi (2016) and Subramonyam *et al.* (2019). However, the use of digital tools to support creative design has been subject to criticism mainly due to the constraints they might impose to free ideation processes, including problems such as circumscribed thinking, premature fixation and bounded ideation (Robertson and Radcliffe, 2009; Chandrasegaran *et al.*, 2013). We posit that these issues appear when software tools are used as facilitators to express ideas, since the functionalities offered by the tool can limit the type of ideas generated (Robertson and Radcliffe, 2009) and do not support the unstructured thinking of designers and users (Chandrasegaran *et al.*, 2013). For that reason, our proposal is not using a software tool to generate ideas but to document and follow up the outcomes of co-design. We consider software tools as an additional virtual co-design space that coexists with other spaces, including the physical design space along with the activities performed to ideate and design solutions (Sanders and Westerlund, 2011). Another interesting approach to apply a software engineering perspective to design research projects is presented in vom Brocke *et al.* (2017). In this case, the tool is focused on documenting the process flow, that is, the tasks performed and their evolution in terms of iterations. Our approach is more focused on the design outcomes than on a design process that by nature is quite flexible and fuzzy.

3. CODICE: A VIRTUAL CO-DESIGN SPACE TO DOCUMENT THE IDEATION AND DESIGN OF AUGMENTED EXPERIENCES WITH PHYSICAL OBJECTS

We faced the challenge of balancing design thinking and software engineering practices within the context of co-designing digital enhanced experiences with cultural heritage in the meSch project. The project involved 12 partners from different European universities, companies and museums who held regular co-design workshops around Europe. Several techniques were used and lots of outcomes were generated to identify potential artifacts, scenarios of use and users. Most of them were made up of physical materials that deteriorate over time, including paper based storyboards, features lists or personas (see left picture in Fig. 1) or they were video-recorded descriptions of scenarios and other ideas (see right picture in Fig. 1) that do not have any explicit semantic associated. Since more than one session was held and not all participants were always present, these outcomes needed to be shared and revisited. In order be able to collect such outcomes in a persistent and distributed way, the CoDICE tool was implemented. A design constituent includes the models, conceptualizations and artifacts related with an object of design (Binder *et al.*, 2011). Development was guided by requirements identified in the literature and by the experience gained in the co-design workshops as described in Díaz *et al.* (2015). During these workhops, the funcionalitites and the conceptual model of agumented computing was refined to allow experts and novice designers articulate better their ideas. The goal pursued was to provide software support to document all the outcomes in a meaningful, persistent and interrelated way and to implement a traceable and repeatable process, where ideas, decisions and evolutions can be revisited. Next paragraphs describe the co-design spaces supported, the DBOs offered and the functionalities of the tool.

3.1. CoDICE virtual co-design spaces

CoDICE is conceived as an additional virtual co-design space that supports documentation and further elaboration of ideas and early designs in co-design events. The tool assumes that design is both divergent and convergent (Rhea, 2003). During divergent design, co-design teams engage in framing the problem properly and stimulating creativity to ideate innovative solutions. Ideas are not evaluated or judged yet, the goal is to end up with as many ideas as possible. During convergent design, the branch of potential designs is narrowed down and the co-design team has to make choices about the best ideas

FIGURE 1. Examples of ephemeral and unstructured outcomes from co-design workshops.

using some (rational or non-rational) criteria. CoDICE assists heterogeneous teams through three virtual co-design spaces:

(1) *CoDICE-Mobile* makes it possible to collect videos and pictures during situated design and ideation tasks. Co-designers can take pictures of affinity diagrams or record use scenarios and other inspirational material that they will use to document and justify their ideas. In this way, they can apply any generative technique (see code-sign.website for information on techniques used during the project) as far as they collect the outcomes in any kind of electronic format, so that they can be kept it a permanent way. These elements are called *resources*, and they can be linked to any *idea* or *design* in CoDICE-Desktop. From a functional point of view, CoDICE-mobile is quite simple and its use can also be skipped since co-designers can use their own mobile phones cameras and directly upload the *resources* through the desktop interface, so the remaining of the paper only focuses on CoDICE-desktop and its two co-design spaces.
(2) *CoDICE-Desktop Ideas* makes it possible to keep track of all the ideas that emerged during divergent design. Ideas are documented in terms of *objects*, *encounters*, *personas* and *augmented scenarios*, four DBOs (see next subsection) that promote reflection upon the current features of the physical objects, the goals pursued in the interaction scenarios, the features of the targeted users and the potential augmented technologies that will be developed.
(3) *CoDICE-desktop Designs* collects the design constituents from convergent design as well as their evolution. *Ideas* that are mature enough are moved to this space where more information concerning *personas*, *scenarios*, *prototypes* and *requirements* is provided (see the full description in the next subsection). *Designs* here are early designs that will need to be prototyped and tested in the wild before being completely implemented. Information concerning this evolution can be added to the DBOs as it becomes available.

3.2. CoDICE DBOs

CoDICE is a virtual co-design space to store all kind of ideas and design outcomes, even those that did not evolve into final products. With that purpose, it provides co-designers with a number of DBOs that conceptualize the design space and that could be identified with design contituents. Figure 2 depicts the conceptual data model made up of such DBOs and their relationships. For the ideas co-design space the data model will read as 'Physical *objects* can be enhanced through *encounters* that make use one or more *augmented concepts* to improve the experience of one or more *personas*'. Similarly, the design space reads as '*Personas* experience *scenarios* in which *prototypes* are used. *Prototypes* might be evolutions of other *prototypes* and they satisfy a number of *requirements*'. The separation among *scenarios* and *prototypes* (called *encounters* and *augmented concepts* if they are still in the ideas space) makes it possible to envision scenarios that make use of several prototypes or prototypes that can be used in different scenarios, providing more opportunities for ideation by combining existing resources. Table 2 includes a description of the DBOs included in the tool.

3.3. CoDICE functionalities

The platform is implemented as a client-server architecture with two clients supporting two co-design spaces: CoDICE-mobile is an Android application that allows one to upload to a common server tagged pictures and videos, so they are

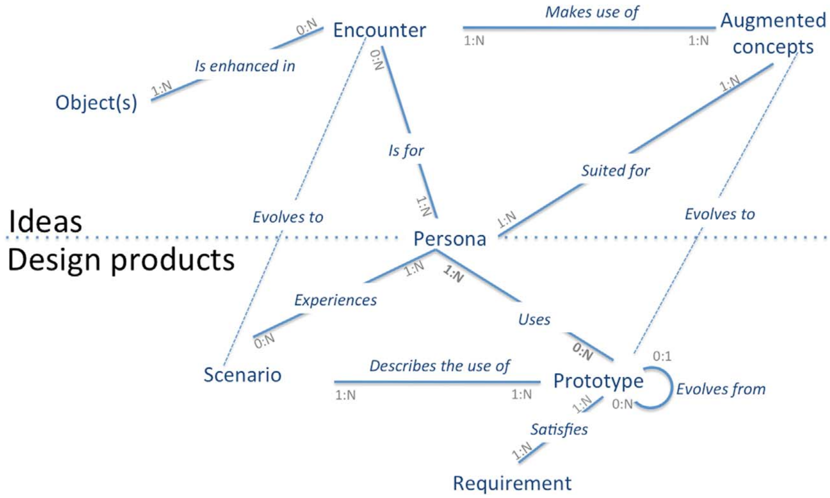

FIGURE 2. Conceptual model of the BDOs used in CoDICE-desktop to document ideas and early designs.

TABLE 2. DBOs in CoDICE Ideas and Design spaces.

Space	DBO name	DBO goal
Ideas	*Object*	It is used to reflect upon the physical objects to be augmented. Based on the artifact analysis method (Díaz, 2012), it pushes co-designers to analyze in-depth the objects features, interaction affordances and constraints and the emotions and feelings they inspire.
	Encounter	It collects information on potential augmented experiences. It could be considered a proto-scenario very informally defined to describe how the interaction might be at a glance.
	Augmented concept	This DBO encapsulates an informal description of the digital artifacts or smart objects that could be implemented to support the augmented *encounters* with the physical *objects*.
Ideas design	*Persona*	It is based on the personas concept as interpreted by Krippendorf (2005), that is, as archetypical users of an *augmented concept* and the ones for which *encounters* are envisioned.
Design	*Scenarios*	It is the evolution of an *encounter*. Scenarios provide more detailed information including the kinds of experiences covered.
	Prototypes	It is the evolution of an *augmented concept* and encapsulates the specification of the smart object that will be implemented. Prototypes can evolve into other prototypes whether to be used in a different scenario or due to limitations observed during the use of the prototype. Prototypes include not only technical information but also a design rationale and information about real uses.
	Requirements	It collects the software requirements that have to be fulfilled in the implementation. Requirements can be related to other requirements and can be subject to dependencies with other requirements.

Ideas can potentially evolve into design DBOs.

available in the desktop tool as resources to inspire or illustrate ideas and designs; CoDICE-desktop is a rich Internet application implemented with MS Silverlight that supports the ideas and designs spaces. The system has been implemented as a multi-device environment and can be used in cell phones, tablets, tabletops and desktop computers to support different technological configurations.

CoDICE-desktop provides functionalities to create and update reports on ideas and designs expressed using the DBOs described in Table 2. It also includes three additional elements that help to organize better the virtual co-design space: *resources, projects* and *workshops*. *Resources* are digital contents (such as images, videos, documents, files, etc.) that can be reused throughout the co-design process and that include all the materials captured using CoDICE-mobile or any other file uploaded to the tool. For example, a recording of a scenario performance, a formal document about standards to follow or a sketch done with a specialized sofware tool such as those reviewed in Chandrasegaran *et al.* (2013) can be uploaded as a *resource* and then they can be

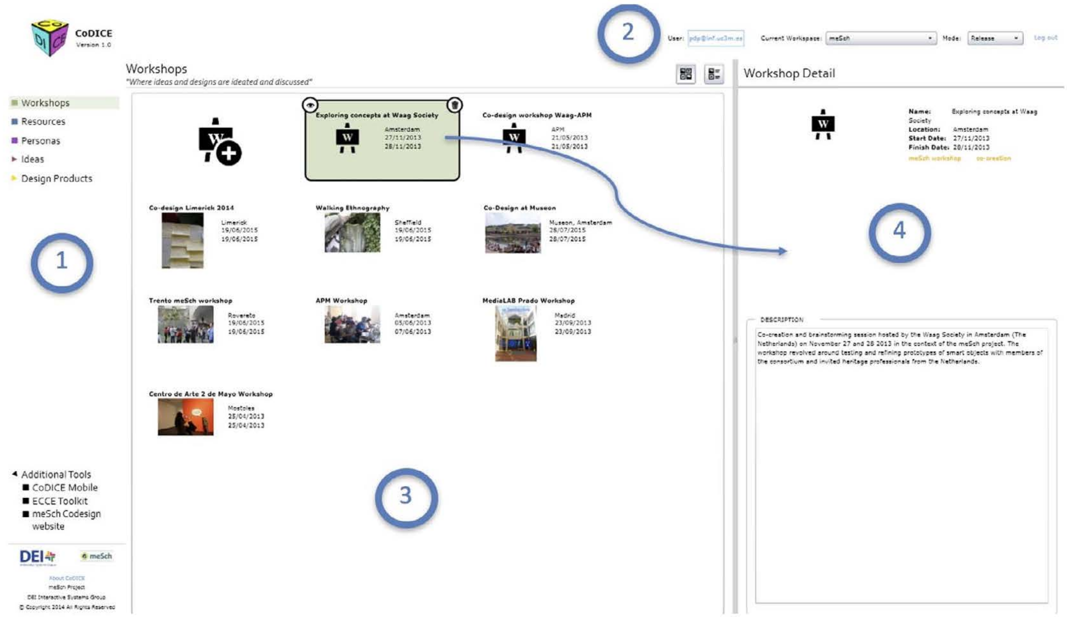

FIGURE 3. Interface of the CoDICE-desktop Gallery of components.

used to document any *idea* or *design*. *Projects* and *workshops* make it possible to group ideas and designs concerning specific co-design projects and co-design events within a project respectively. This a soft way to structure the process not as detailed as the one proposed in vom Brocke *et al.* (2017).

Each DBOs has a gallery and a component interface (see Figs 3 and 4). The gallery of components is the entry point to any component of CoDICE and it is shown whenever one of the categories in the left navigation tool is chosen (see area 1). For each type of component (*workshops* in Fig. 3), area 3 shows all the instances created. When one of these visual representations is chosen (see 'Exploring concepts at Waag society'), some information is shown in area 4 and two icons make it possible to edit the component (eye icon) or delete it (trash icon). Area 2 of the interface provides general information of the current session, such as the user name or the current project.

Figure 4 shows the interface for a DBO, that is, the interface when the 'eye' icon is selected in the gallery, in the figure an *augmented concept* called The Transparente Window. Note that the left navigation tool and the session information are removed from the image for readability purposes. Co-designers can name, tag and vote the idea (see area 1); provide details including any kind of multimedia material or *resource* (see the icons of different types of files in area 2); comment on it or link it to related components (see area 3); and click on the history button (see area 4) to see who did what on the object. The fields included in each DBO where identified through an iterative process based on formative evaluations (see Section 4). To support an open, evolutionary and flexible process, there are no mandatory fields. Ideas and designs might only have a few details that will be increased over time as co-designers gain knowledge on them.

4. CODICE EVALUATION

CoDICE was developed following an iterative process informed by formative evaluations that provided empirical information on perceived usability and utility from different audiences as reported in (Diaz et al., 2015; Díaz *et al.*, 2017*)*. In particular, seven formative evaluations were run: two exploratory focus groups to discuss early prototypes and ideas and five empirical evaluations with stakeholders and end users. In total, 97 different partipants were exposed to the tool, including designers, hci experts, cultural heritage professionals, students and programmers. This process contributed to adapt the tool to the necessities of multidisciplinary teams, respecting the requirements of different disciplines: from the freedom of expression valued by end users and creative designers, to the need to impose a certain level of rationality required by technical participants. For example, the first prototype included a functionality to create affinity diagrams that was not valued by the evaluators since they preferred to work with physical materials like post-its (Diaz *et al.*, 2015). This finding led to decide not to implement components to support to specific design techniques but instead to offer an additional virtual space to reflect on and elaborate further the digitized outcomes of such techniques. Next formative evaluations were aimed at validating the DBOs and the information they contained. Similarly, the tool functionality evolved as we observed users interacting with

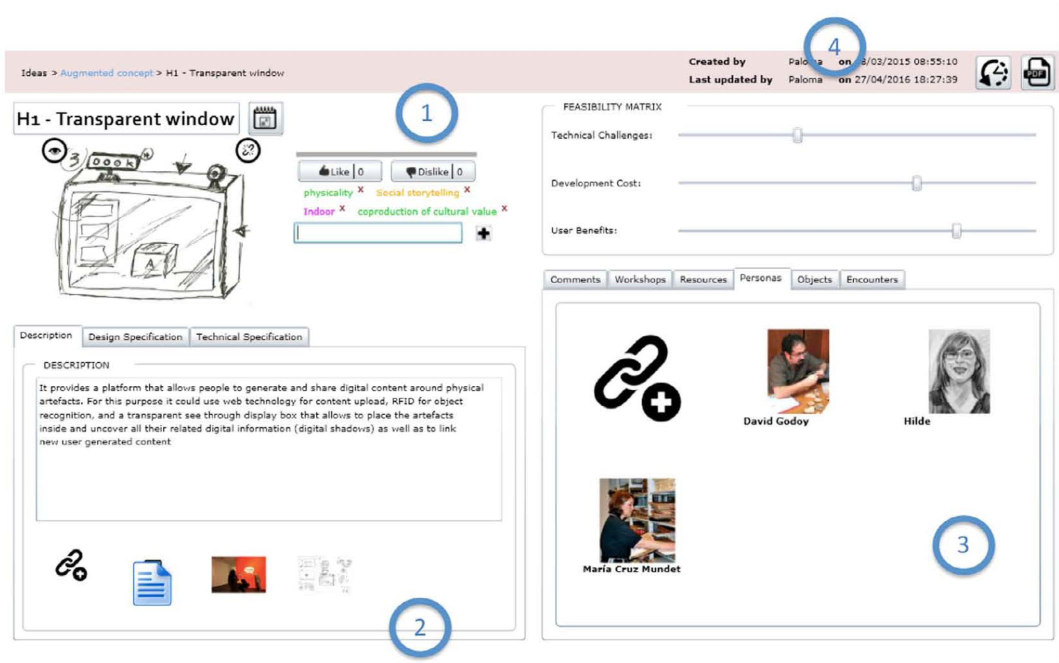

FIGURE 4. Interface of an augmented concept, that is, a potential prototype.

the first prototypes. For instance, validation and verification rules about the conceptual model relationships were lessened to achieve a compromise between not burdening designers with complex checking processes and yet guaranteeing a minimum level of elaboration before an idea is transformed into a prototype. For the shake of clarity and space, in this section, we only describe the summative evaluation performed once the tool was completed. Such summative evaluation consisted on two experiments.

4.1. First summative evaluation experiment

A first, summative evaluation was performed with 20 students of a Master in Cultural Heritage Management. These students had a varied profile, from humanities and social sciences to cultural heritage professionals working in private or public entities. None of them has a technical background though throughout the Master they are exposed to innovative uses of technology in different cultural settings. Before the co-design session started, they visited Parque Carranque, an archeological site where they were asked to take pictures that would serve as inspiration to ideate augmented experiences. In the next class, participants organized in groups used CoDICE during a 3 hour co-design session to propose one augmented experience. During the first 80 minutes, they were encouraged to brainstorm and propose multiple ideas that they documented using the ideation space. The next 80 minute slot was devoted to chosing one of the ideas and elaborate it further using the design space until they produced a whole description of the prototype, the scenarios of use and the potential personas who will benefit from it. At the end of the session, they presented their designs to the classroom and they filled a questionnaire on the utility of the tool. The questionnaire had two parts: the first part explored the utility of CoDICE DBOs and functionalities using a 5-value Likert scale ranging from 'very low' to 'very high'. In the second part, two open-ended questions collected participant's perceptions on the main benefits and drawbacks of the tool and they were also asked to state whether they will use it again and for which purposes.

As shown in in Table 3, all the DBOs and functionalities were unanimoulsy considered useful, all of them with a mean value higher than 3,5 except for the gallery of objects that was considered below this threshold. This might be due to the fact that the gallery of objects, where you get access to all the items within a category is more useful when multiple items are generated and need to be explored throught a longer period of time (see the scenario described in Section 5.2.1). Concerning the open questions, 14 out of 20 participants stated they would use it again and 2 of them thought it could be used also for ideating other types of technological interventions not necessarily related with augmented spaces. When asked about the tool utility, the 13 participants who answered highlighted the idea of being able to organize all the information around a potential augmented idea in an easy way and providing different perspectives of a technological product (potential users, potential uses, technical features and challenges . . .). From their answers, four main uses emerge: to identify needs and create designs, to capture all the ideas, to perform a deeper study of objects and

TABLE 3. Results of closed questions on utility of each BDO in the first summative evaluation ranked from 1 (very low) to 5 (very high).

	Av.	Mode	Var.
Separation between ideas and designs	3, 9	4	0,6
Adding resources	3, 7	4	0, 8
Gallery of components	3, 4	4	0,8
Objects	3, 8	4	0,5
Personas	4, 5	5	0,4
Encounters	3, 8	4	0,4
Augmented concepts	4, 6	5	0,2
Transition from ideas to designs	3, 7	4	0,4
Prototypes	4, 4	4	0,3
Scenarios	4, 2	4	0,5
Relation among entities	4, 0	4	0,4
PDFs generation	4, 0	4	0,4

experiences and to share all the information about the design. These uses, which are consistent with the evolving goals of the tool that were established during the formative evaluatios, were studied further in the next evaluation.

4.2. Second summative evaluation

The findings of this summative experiment might be somehow biased due to the homogeneous profile of the participants, none of them with a technical background. For this reason, a second experiment involving heterogeneous groups was carried out. To have varied profiles and to ensure that participants were personally interested in taking part in the session, a call for participation was issued through social networks. A total of 13 people took part in the workshop, none of whom was a member of the meSch project nor have had any previous experience with CoDICE. According to how they described their expertise and profile, it was a heterogeneous mix of people related with cultural heritage (students, academics and professionals) and people with a technical background (one civil engineer and two computer engineers). Participants were organized in two groups of 4 and one group of 5, each of which included one technical participant. The session was scheduled to last 4 hours. However, participants were so immersed in the task that it took over more than 30 minutes. After a short introduction of the workshop goals and the CoDICE interface for the ideas space, participants engaged in a 2 hour session of ideation. They were asked to be creative and do not think yet about the cost or the technical complexity of their ideas. They were proposed a method that started thinking on the physical objects, the encounter, the personas and, finally, the augmented concept, though this was just a recommendation. Most groups started to think about encounters and augmented concepts and then decided the kind of physical cultural heritage item or collection that will be involved and the type of visitors that could benefit from the experience. Though some physical objects were available in the room as design probes, participants with a cultural heritage background dominated discussions on the type of object to be augmented and they chose topics they were personally interested at. All of them documented at least one object, one persona, one encounter and one augmented concept. When the ideation process was completed, participants were introduced to the DBOs of the design space and they worked for 1 hour and a half on the convergent design process. They were asked to select one of the ideas they had during the ideation process and make it evolve into a design concept. All groups picked up an idea for which they provided a quite elaborated SWOT matrixes to justify their decisions. In particular, they focused on multidisciplinar issues such as the user experience, the features of the artifact, the perception of the cultural institutions and the context where the artifact was going tobe deployed.

Once participants finished, they were asked to fill a questionnaire about the experience. The questionnaire gathered information on perceived utility and on the potential uses of the tool. Concerning utility, a 5-value Likert scale was used to assess in an improved version of the questionnaire used in the first experiment that was too long since it explored each functionality in a separate way. In this case, only generic questions about the tool utility and ease of use were asked. Table 4 summarizes the results about utility that were all over 4, being the possibility of generating a file with all the information the one that was best ranked. Indeed, one of the participants said '*I'd like to share this file in social media to see if I find somebody interested in funding me!*'

To explore potential scenarios of use where CoDICE might provide a significant value, four options extracted from previous co-design events were assessed: identify needs and create designs, be able to capture all the ideas, carry out a deeper study of objects and experiences and share information with other groups. Again a 5-value Likert scale ranging from 1 (completely disagree) to 5 (fully agree) was used and, as shown in Fig. 5, all participants agreed with the four options being the

TABLE 4. Results of closed questions on utility in the second summative evaluation ranked from 1 (very low) to 5 (very high).

	Av.	Mode	Var.
Utility of the tool	4, 4	4	0, 24
Utility of the separation between ideas and designs	4, 2	4	0, 28
Utility of relating concepts	4, 2	4	0, 64
Utility of the transition among ideas and designs	4, 1	4	0, 69
Utility of the generation of pdf files with all the information	4, 6	5	0, 61

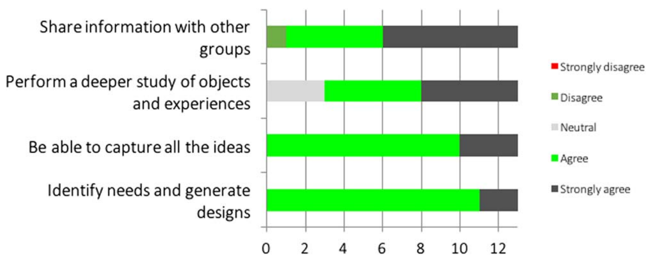

FIGURE 5. Summative evaluation of utility and potential uses of CoDICE.

ability to share information with other people the most valued scenario. Only one person disagreed with one option (share information) because the participant did not like the idea of sharing concepts among participants, but this is the main goal of co-design. The last question was whether they will use the tool again and why, and 11 out of 13 answered positively. Among the reasons to use CoDICE, participants indicated '*to explore ideas and develop them in a collaborative way*', '*to organize better my ideas about augmented experiences*' or '*to facilitate the organization of ideas through a guided process*'.

Finally, they were asked if they will use the tool again and 11 out of 13 participants said they would, primarily to explore ideas and develop them in a collaborative way. More evaluations involving more designers with varied backgrounds and agendas might be necessary to obtain sound conclusions on the tool utility tool. However, considering the results and that all groups were able to create and defend a design, we can conclude that CoDICE might be a useful tool to explore ideas, keep track of them and share the rationale supporting the decisions taken.

5. DISCUSSION ON CODICE UTILITY

In this section, we discuss the tool utility by describing how it combines software engineering and design thinking practices and introducing two different scenarios of use that focus on how it helps expert and novice designers. The section ends summarizing some of CoDICE contributions and limitations.

5.1. Balancing design thinking and software engineering practices in CoDICE

CoDICE enforces some basic software engineering practices while keeping the freedom and expressivity of design thinking methods to contribute to improve the outcomes of co-design events. Table 5 summarizes how our proposal contributes to deal with the challenges identified in Table 1 that are discussed in next paragraphs.

Support multiple co-design spaces. Having in mind the limitations that the use of computer based tools can impose in free thinking and exploration activities (Robertson and Radcliffe, 2009), CoDICE is not aimed at replacing or making digital the co-design activities envisioned by the group leaders to promote ideation and discussion. Instead, it is expected that some members of the co-design team will use it to document the outcomes once co-design activities have been completed, so the outcomes can be shared and revisited by all the participants on the assumption that sharing designs can contribute to improve design exploration and results as discussed in (Dow *et al.*, 2011). Moreover, DBOs can be linked with the workshop they were created to facilitate its identification in long-term processes involving more than one onsite co-design event. In this way, CoDICE extends the design sessions in a distributed virtual space were the design outcomes can be visualized, documented and reflected upon as suggested in (Sanders and Westerlund, 2011).

Support the exploration of multiple ideas. CoDICE-desktop Ideas is the virtual co-design space aimed at allowing co-designers to revisit all the ideas of the project, even those that are discarded. Figure 6 shows a screenshot of the collection of potential augmented concepts that emerged in the meSch project. In this case, the co-design team assigned alphanumerical codes to those ideas that were implemented (see 'H1 – Transparent window' or 'C11- The Loupe'), but those like 'Memento Mori' that were not elaborated are still kept in the database for future use. Since the tool expands the ideation process over time, additional material and thoughts can be added at any moment, and thus ideas that were initially rejected or not considered could at some point become worth to be moved to the design space.

Generate a shared representation of the outcomes. As described above, a number of DBOs are used to document the co-design outcomes. To implement a not too strict process

TABLE 5. CoDICE technological approach to contribute to alleviate co-design challenges.

Challenge	CoDICE approach
Support multiple co-design spaces	Situated design and free ideation are supported whether using CoDICE-mobile or directly uploading files to CoDICE-desktop. CoDICE-desktop supports synchronous and asyncrhonous documentation of and reflection upon the design outcomes, going beyond the limits of colocated meetings.
Support the exploration of multiple ideas	All ideas can be kept in the tool as well as the reasons why they did or did not evolve into design concepts.
Generate a shared representation of the outcomes	DBOS are organized as a conceptual model of the domain of application that has been tested through formative evaluations. Co-designers can document their outcomes using any kind of information resource (from text to pictures, videos, files..). DBOs can be revisited and commented by all participants at any time.
Make explicit the process, the goals of design outcome and their relationships	DBOs are related in order to make explicit the relationship among design tasks. Validation rules force to follow a certain process.
Support documentation and information processing	BDOs document the design outcomes in a flexible, meaninful and evolutionary way. Links in the tool interface facilite also traceability of the relations among outcomes as well as of their evolution.
Justify the design decisions	Ideas cannot be moved to the design space until they are not explored enough and have at least a SWOT matrix DBOs also include comments and votes from the co-designers. Prototypes include a section on known uses to provide evidence on their empirical evaluations.

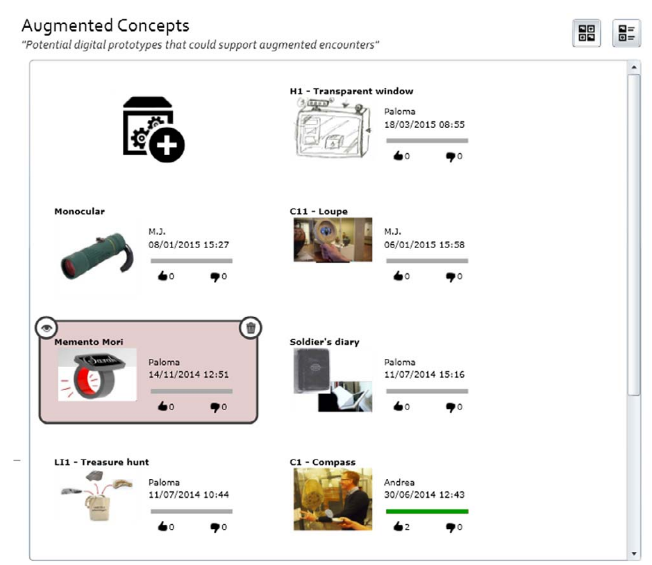

FIGURE 6. Ideas for augmented concepts generated throughout a long-term project.

flow and assuming that design is an open proces (Binder *et al.*, 2011), there is no obligation to use all the DBOs in a co-design event or project. Moreover, to provide more semantics to these outcomes, DBOs make up a conceptual model of the domain of application (augmented computing) that captures relevant relations and constraints (see Fig. 2). In particular, relations guided the inclusion of elements in the CoDICE interface to browse and add connections among DBOs, so that co-designers can have a broader view of each concept that includes related design constituents. As shown in Fig. 4, the tabs in area 3 make it possible to see which *personas*, *physical objects* and *encounters* are related with an *augmented concept*. On the other side, cardinalities in the model reflect possibilities of reusing a DBO in multiple relations (when they are n-ary) as well as constraints to at least make the DBO participate in one instance of the relationship (when they are set to 1) that turn into validation rules as discussed below.

Make explicit the process, the goals of each design outcome and its relationships with other design outcomes. CoDICE is a thin infrastructure (Binders *et al.*, 2011) that softly structures the co-design process while letting co-designers free to use any generative technique or tool to avoid any kind of circumscribed thinking. As a thin infrastructure, it does not make you necessarily more creative or successful per se, but it offers a distributed co-design space to share and revisit ideas. The process is structured both implicitly (guided somehow by the way DBOs are ordered in the tool interface and, therefore, not necessarily used) and explicitly (forced by a validation rule that make it mandatory exploring ideas before proposing designs). The process can be organized according to the goals and needs of the team since no flow is imposed, except that all the DBOs in the design space cannot be created from scratch, they have to be evolutions of DBOs in the ideas space. This practice might not match the practice of some professional designers, but co-design involves heterogeneous teams and for all those not used to generative techniques, having a framework to structure ideas before turning them into designs might be helpful as shown in Section 5.2. In any case, sharing ideas among team members eases individual exploration, conceptual blending and collaboration (Dow *et al.*, 2011).

Support documentation and information processing. Documentation is a way to communicate design knowledge among different participants and to support traceability (vom Brocke *et al.*, 2017). Traceability is a key feature of any software specification that consists of providing links among related entities or artifacts, to be able to look at the specification from different perspectives and to understand decisions (Pressman, 2010). It helps to support the generation of design rationale and has been used in software engineering for validation purposes (Pohl, 2010). As discussed before, CoDICE provides links in the interface among DBOs in the same or different space. Relationships are also used to validate the transformation of ideas into design concepts. To guarantee that, as posited by Krippendorf, design is a way to '*give significance by assigning it to a use, a user, or an owner*' (2005), two validation rules are enforced: an *augmented concept* cannot be moved to a *prototype* if (i) there is not any *encounter* that makes use of such concept and (ii) there is not at least one *persona* that would benefit from it. Similarly, an *encounter* has to be linked to at least one *persona* and one *augmented concept* to become a *scenario*. These validation rules are automatically fired when co-designers try to transform ideas into design concepts and they need to be satisfied before proceeding with the next step, filling the contents of a SWOT (strengths, weaknesses, opportunities and threats) matrix associated to the resulting DBO in the design space.

Justify the design decisions. Designers need to provide additional information about their products to understand how they can be used and their limitations as well as their evolution (Carroll and Rosson, 2003; Krippendorff, 2005; Binder *et al.*, 2011). To cope with all this kind of information in a manageable way, CoDICE provides opportunities to gradually document the features of the outcomes through a number of DBOs, including personas, scenarios and prototypes. Each of them provides a perspective of the technology that is focused on user goals and expectations, user experiences and contexts of use and technical capabilities, respectively. The tool also provides some elements to justify design decisions like the votes of the participants about each DBO and their comments (see arrows tagged with 1 in Fig. 7), its relations with other DBOs (arrows 2) or the SWOT matrix attached to each design DBO (arrow 3). Prototypes, that is, the final product to be generated, also include the validation and known uses fields (arrows 4) where co-designers can add information about how it was evaluated and how it worked in real experiences repectively.

The tool has components to trace back or forward the genetic evolution of artifacts (Krippendorff, 2005), from ideas, to design concepts to alternative design concepts. All this information is tied to the *prototype* concept. First at all, *prototypes* are linked with the idea (*augmented concept* in CoDICE terms) they evolved from (see arrow 5 in Figure 7). *Prototypes* can inspire other prototypes and, therefore, they can evolve over time. Arrow 6 show how each prototype provides information of the previous versions and it is also linked to evolutions if any. In the example, the cases were a different version of a previous prototype called the Plinth and they did evolve into Cases 2. To keep trace of all the changes done using the tool, users can click on the button in the top of Fig. 6 (see arrow 7) that expands a list of the updates including the name of the participant who did the modification. Finally, to have a complete view of all of this information, CoDICE users can generate pdf files of the DBOs (see arrows 8), where they can apply a cascade-printing option (multiple pdf file) including the information of all the related DBOs.

5.2. Scenarios of use

5.2.1. *Using CODICE to document a long-term and multidisciplinary project*

CoDICE was used to document the outcomes of the co-design events of the meSch project that were held during 1 year and a half among partners distributed accross Europe. Participants in the events had different backgrounds, including professional designers, cultural heritage professionals, software and hardware developers, and were not always present in all the co-design workshops. Many different kinds of techniques were used to engage participants in ideation of augmented experiences with cultural heritage. For a comprehensive description see *www.mesch-project.eu*. After the first year, a fully operational prototype of CoDICE was used to document a specific case: the conception and evolution of the Loupe (Diaz *et al.*, 2015), a physical loupe equipped with a phone to provide an augmented reality experience. A cultural heritage academic collaborated with a software engineer to collect and integrate all the available information about how the design concept was ideated and evolved over time. In particular, CoDICE made it possible to document and share how and when this concept was conceived and why other similar concepts were discarded. For example, the separation between scenarios and prototypes was exploited to document how the Loupe worked quite well in the '*layered narratives*' scenarios where it was used to provide several augmentations of the same physical object whilts it did not work when used for '*wayfinding*'. In a similar way, when creating the *personas* for a specific museum one elder woman was identified and that led to discard one of the artifacts envisioned, the monocular, for accesibility and usability reasons even though it seemed to be culturally resonant for the scenario. All this information was attached to the design concept, so all the co-designers had a broader view of it, even if they did not attend all the co-design workshops or did not take part in the evaluation experiences.

From this scenario of use, we learnt that in long-term, multidisciplinary and distributed projects, forcing creative teams to

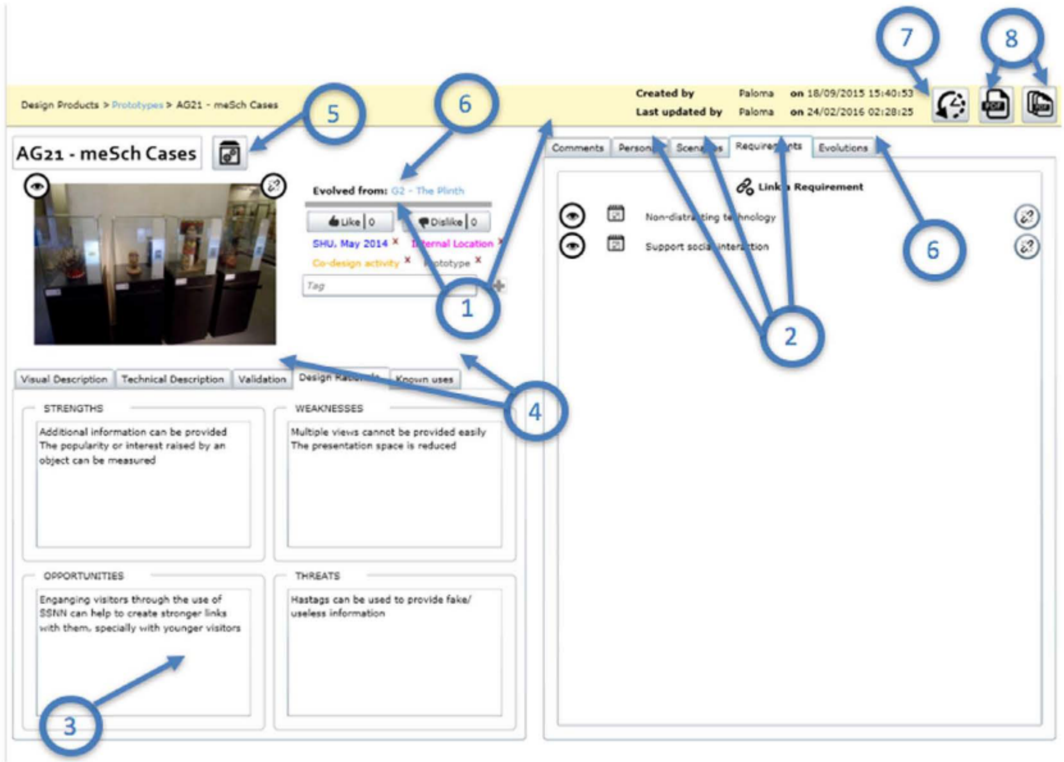

FIGURE 7. Interface for prototype entity in CoDICE. The different possibilities used to provide justification on the validity of the concept and to document its evolution are marked.

be systematic pays off. CoDICE is not expected to make you more creative, but it can help teams to be more structured and to revisit ideas and outcomes once co-design workshops are over, so that eventually designs will be improved by sharing them (Dow *et al.*, 2011). The use of the *workhosps* concept to collect relevant issues raised in specific co-located activities (such as open questions or activities carried out) and the capability of navigating from ideas to prototypes and from prototypes to their evolution were useful ways to keep trace of the work done throughout the project. However, experts and novice designers have different ways to approach design (Kim *et al.*, 2011) and while the first ones rely upon their design abilities, novice designers usually require higher guidance and the use of more structured and formal methods (Seidel and Fixson, 2013) like the structuring proposed by CoDICE. Being able to revisit preliminary ideas and to visualize their evolution provided a better understanding on the final outcomes and on the reasons that motivated the team decisions. This was particularly relevant in this scenario since not all team members participated in all the activities. Concerning limitations, some designers found documenting ideas boring and unnecessary for them, so this task was assumed by more technical backgrounds. Moreover, some of them did not want to be forced to follow a specific flow (‘*Why can I not move an idea to the prototype level without defining a scenario or persona*?’). This might be considered restrictive by a very experienced designer, but it makes things easier for novice designers and stakeholders. Rules provided a guidance on the process to follow and helped to put ideas in context, associating them with a scenario of use and the personas who will use the prototype. Indeed, rules guaranteed a complete documentation of all the ideas. At the end of the project, the tool was used to document both success and failure cases so the CoDICE space became a permanent memory of the whole co-design process (see Fig. 6).

5.2.2. *Using CoDICE to support co-design events with novice designers*

In order to explore the use of the tool outside the boundaries the project, we also ran several co-design events with novice designers with no predefined agenda nor knowledge on design or software engineering. In particular, we run five events: three of them were short sessions of 4 hour run with students of a cultural heritage master (2 experiences involving 43 and 20 students, this latter reported in Section 4.1) or mixed background participants (one experience with 13 people reported in Section 4.2); and the two remaining workshops where 3 week projects that involved 1 weekly co-located session of 2 hours per week and that involved students from a course on interaction design (8 and 6 in the two sessions). All of these experiences tested the tool *in-the-wild*; it was used by completely novice users in an uncontrolled environment where participants were using their laptops and connecting remotely to a server.

This scenario made it possible to analyze how novice designers use and perceive the use of the tool to ideate an innovative artifact. All the teams developed several ideas and converged to a unique design, a process in which they appreciated being guided by the tool since they were not experts in design nor in software engineering. In the words of one of the participants, '*Starting from a lot of ideas, CoDICE helped to organize them in the best way, working step-by-step you could see how the project evolved*'. This is an important aid for novice designers that might be not so welcome by experts as it happened in the previous sceanrio. In all cases, one or two members of the group were in charge of populating CoDICE DBOs so they could be accessed by other members when needed. In the 3 week projects, this capability was particularly useful, since participants could work in a distributed and asynchronous way. As one of the participants explained '*The main benefit is that it allows you to make collaborative brainstorming, without the need to be all together in the same room. It allows to have something persistent at the end of a co-design workshop and allows ideas to be shared'*. Moreover, since all participants in one workshop shared the same CoDICE project, there was some level of crosspollination that was valued as a useful option to get inspiration from others' ideas.

All the participants valued the use of CoDICE DBOs to organize their ideas in a clear and communicable way and even some of them pinpointed this benefit as a way to enable non-technical students to ideate digital futures. They highlighted that structuring their ideas could contribute to make them better understood by other people, like their managers at cultural institutions ('*If you print this pdf, it is clear you have reflected upon the idea quite a lot, it is not like you just got it on your way to the meeting*'). Moreover, both technical and non-technical participants valued the aid provided by the tool in terms of making it explicit a design flow and a number of DBOs, which according to them helped to improve communication and '*put some order*' (in one of the participant's words), specially among heterogeneous profiles. The capability of having all the elements connected was valued by the participants in the workshops as a way to provide additional value to their proposals. As one of them said '*You can describe in an easy way your project, what do you want to do, your goals, how the experience will be and who will use it. And I like that!*'.

5.3. CoDICE contribution and trade-offs

To sum up, compared to general purpose collaborative tools or to design thinking and annotation tools, CoDICE is a domain specific tool that goes beyond sharing documents by integrating software engineering practices to explore further problems and ideas in a meaningful, structured and traceable way. It is meaningful and structured since all the design outcomes are organized into DBOs that gather key domain entities (physical objects and spaces, users, scenarios of use and technologies) and their relationships. It is traceable since all the relationships among DBOs are explicit and visible and the information on the design rationale and the evolution of the ideas is attached to them. In particular, three main contributions of CoDICE emerged when used in co-design events:

- *It softly structures the design process.* The separation between the ideation and the design space, the inclusion of DBOs to explore the problem and the solution and the constraints imposed by the conceptual model provide some structure to the design process. When entering for the first time a project, designers are guided through a number of design constituents that push them to analyze the problem from different perspectives. This feature has been found particularly useful in workshops like those reported in Section 5.2 where experts in a domain such as cultural heritage engaged in ideation tasks without having any prior knowledge on software engineering or design thinking techniques. Participants used the CoDICE DBOs to articulate their ideas in a structured way that facilitated discussion and the elaboration of a shared design rationale.
- *It encourages a multiperspective exploration of ideas.* As said before, CoDICE offers a number of DBOs that capture the semantics of the domain of augmented computing and whose elements include features that are relevant for different types of participants, from emotions, to cultural interpretations, user stereotypes, evaluation outcomes or technical characteristics. The ability to explore ideas from different perspectives was particularly valued by non expert designers since it forced them to explore further or even discard their initial thoughts. As one of the participants said '*We were convinced this was a good idea till we moved to the definition of the scenario and the personas. Then we realised it was just overcomplicated*'.
- *It provides a process to explicitly capture design rationale and trace the evolution of the design constituents.* The tool provides many mechanisms to trace designs, from elements in the interface that connect related design constituents, to the whole pdf file documenting a prototype where all the process is reported. In the scenario in Section 5.2.1, the capability of tracing long-terms processes made it possible to keep an structured view on how all the prototypes were ideated, designed and tested and even on how they evolved into other prototypes to deal with new requirements.

The tool does not impose a unique process since we consider there is no *one-size-fits-all* method that can suit all the casuistry around design projects. However it still forces designers to think about the ideas from different perspectives before deciding what to develop. This is a clear trade-off of CoDICE: very experienced designers find boring and limiting documenting ideas and usually other members of the team had to use the tool for documentation purposes; however at the end of the project you have a permanent, commented and shared memory of all the decisions that were taken. As a clear limitation, we also found that novice designers cannot use CoDICE without any guidance since the freedom to establish your own process can lead to confusion. For that reason, there was always an introductory session with examples to understand the goal of each DBO, the validation rules, and how they relate with the design tasks. Once this introduction was done, even non-technical users like cultural heritage students were able to create their own scenarios, personas and protoypes.

6. CONCLUSIONS

This paper describes an approach to use a software tool to integrate a certain systematic flow and the rationale forces of software engineering practices with the non-rationale forces and generative ideation methods of design thinking, in order to support a holistic process to design augmented computing ecosystems. The approach provides an alternative virtual co-design space made up of a software tool called CoDICE that participants can use to share, revisit and reflect upon ideas using a number of DBOs. Though some structure and flow is forced through the conceptualization of the domain of application, the tool allows for any generative technique to be applied since CoDICE is used as a complementary documentation tool not as a co-design aiding tool.

The two scenarios of use here presented describe different co-design events where the use of a software tool like CoDICE might be useful to encourage co-designers to explore more than one idea, interrelate their ideas and provide a justification about their decisions. Moreover, DBOs make it possible to reflect upon technological interventions from different perspectives: from the interaction affordances and limitations technologies have but also from the potential goals they could satisfy, the interaction scenarios where they make sense or the kind of users who will enjoy them. After having used the tool to document the outcomes of long-term projects as well as in short term co-design activities, some uses of the tool emerged as a way to improve the co-design process. Among them, we can identify the capability to document and revisit all the ideas, not only those that succeeded; to be able to reflect upon objects and experiences and not only technology; to share and reuse design outcomes among teams members and even among different teams; and to be able to identify different kinds of needs and perspectives.

CoDICE is just a design case about how to integrate some software engineering practices to support more efficient information sharing and design rationale generation, but more research into the use of computational aids for co-design without compromising creativity is still required. In this context, the use of augmented and virtual reality environments to support creativity tasks is a promising direction. The affordances of virtual spaces in terms of engagement, intrinsic motivation, spatial or collaborative learning can be exploited to generate richer computational co-design spaces, particularly now that the availability and prices of the devices are changing dramatically the spectrum of domains where these technologies can be applied.

Funding

meSch project European Commission FP7 (600851).

REFERENCES

Adams, R. S., Daly, S. R., Mann, L. M. and Dall'Alba, G. (2011) Being a professional: Three lenses into design thinking, acting, and being. *Des. Stud.*, 32, 588–607.

Alexander, C. (1979) *The Timeless Way of Building*. Oxford University Press.

Bergman, M., Lyytinen, K. and Mark, G. (2007) Boundary objects in design: an ecological view of design artifacts. *J. Assoc. Inf. Syst.*, 8, 546.

Bernal, M., Haymaker, J. R. and Eastman, C. (2015) On the role of computational support for designers in action. *Des. Stud.*, 41, 163–182.

Binder, T., De Michelis, G., Ehn, P., Jacucci, G., Linde, P. and Wagner, I. (2011) *Design Things*. MIT Press.

Bødker, S. (2000) Scenarios in user-centred design—setting the stage for reflection and action. *Interact. Comput.*, 13, 61–75.

Buchanan, R. (1992) Wicked problems in design thinking. *Des. Issues*, 8, 5–21.

Carroll, J. M. and Rosson, M. B. (2003) Design Rationale as Theory. In *HCI models, Theories and Frameworks: Toward a Multidisciplinary Science*, pp. 431–461.

Chandrasegaran, S. K., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F. and Gao, W. (2013) The evolution, challenges, and future of knowledge representation in product design systems. *Comput. Aided Des.*, 45, 204–228.

Cooper, A. (1999) *The Inmates Are Running the Asylum*. Macmillan.

Cross, N. (2001) Designerly ways of knowing: design discipline versus design science. *Des. Issues*, 17, 49–55.

Díaz, L. (2012) Developing design education and knowledge for heritage. In *Proceedings of the 7th conference of the Turkish Design History Society, 5T A New Affair: Design History and Digital Design Museums, Yazar University, Izmir, Turkey, 17–18 May*.

Díaz, P., Aedo, I. and van der Vaart, M. (2015) Engineering the Creative Co-design of Augmented Digital Experiences with Cultural Heritage. In *End-User Development*, pp. 42–57. Lecture Notes in Computer Science, vol 9083. Springer, Cham.

Díaz, P., Aedo, I. and Bellucci, A. (2017) Integrating End Users in Early Ideation and Prototyping: Lessons from an Experience in Augmenting Physical Objects. In *New Perspectives in End-User Development*, pp. 385–411. Springer, Cham.

Dorst, K. (2011) The core of 'design thinking' and its application. *Des. Stud.*, 32, 521–532.

Dow, S., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, D. and Klemmer, S. (2011) Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Pro-*

ceedings of the SIGCHI conference on human factors in computing systems, ACM, pp. 2807–2816.
Fischer, G., Giaccardi, E., Eden, H., Sugimoto, M. and Ye, Y. (2005) Beyond binary choices: integrating individual and social creativity. *Int. J. Hum. Comput. Stud.*, 63, 482–512.
Forlizzi, J. and Battarbee, K. (2004) Understanding experience in interactive systems. In *Proceedings of the 5th conference on designing interactive systems: processes, practices, methods, and techniques*, pp. 261–268. ACM.
Garde, J. A. and van der Voort, M. C. (2012) Participants' interpretations of PD workshop results. In *Proceedings of the 12th participatory design conference: exploratory papers, workshop descriptions, industry cases-volume 2*, ACM, pp. 5–8.
Giaccardi, E., Paredes, P., Díaz, P. and Alvarado, D. (2012) Embodied narratives: a performative co-design technique. In *Proceedings of the designing interactive systems conference*, pp. 1–10. ACM.
Gregor, S. and Jones, D. (2007) The anatomy of a design theory. *J. Assoc. Inf. Syst.*, 8, 312.
Harboe, G., Doksam, G., Keller, L. and Huang, E. M. (2013) Two Thousand Points of Interaction: Augmenting Paper Notes for a Distributed User Experience. In *Distributed User Interfaces: Usability and Collaboration*, pp. 141–149. Human–Computer Interaction Series. Springer, London.
Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) Design science in information systems research. *MIS Quart.*, 28, 75–105.
Juntoo (2019) http://blog.juntoo.co/design-collaboration-tools/ (accessed June 2019).
Kaptelinin, V. and Nardi, B. A. (2006) *Acting with Technology: Activity Theory and Interaction Design*. MIT Press.
Kim, Y. S., Jin, S. T. and Lee, S. W. (2011) Relations between design activities and personal creativity modes. *J. Eng. Des.*, 22, 235–257.
Klein, H. K. and Hirschheim, R. (1991) Rationality concepts in information system development methodologies. *Account. Manag. Inform. Tech.*, 1, 157–187.
Kleinsmann, M. and Valkenburg, R. (2008) Barriers and enablers for creating shared understanding in co-design projects. *Des. Stud.*, 29, 369–386.
Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M. and Landay, J. A. (2001) The designers' outpost: a tangible interface for collaborative web site. In *Proceedings of the 14th annual ACM symposium on user interface software and technology*, pp. 1–10. ACM, New York, NY, USA.
Krippendorff, K. (2005) *The Semantic Turn: A New Foundation for Design*. CRC Press.
Luebbe, A., Edelman, J., Steinert, M., Leifer, L. and Weske, M. (2010) Design thinking implemented in software engineering tools. In *8th design thinking research symposium (DTRS8)*. University of Technology, Sydney, Australia.
Martin, B. and Hanington, B. (2012) *Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions*. Rockport Publishers.
Nelson, H. G. and Stolterman, E. (2003) *The Design Way: Intentional Change in an Unpredictable World: Foundations and Fundamentals of Design Competence*. Educational Technology.
O'Neill, E., Johnson, P. and Johnson, H. (1999) Representations and user-developer interaction in cooperative analysis and design. *Hum. Comput. Interact.*, 14, 43–91.
Pohl, K. (2010) *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer Publishing Company, Incorporated.
Pressman, R. S. (2010) *Software Engineering a Practitioner's Approach* (7th). McGraw Hill.
Rhea, D. (2003) Bringing Clarity to the "Fuzzy Front End". In *Design Research: Methods and Perspectives*. The MIT Press, Cambridge.
Rittel, H. W. and Webber, M. M. (1973) Planning problems are wicked. *Polity*, 4, 155–169.
Robertson, B. F. and Radcliffe, D. F. (2009) Impact of CAD tools on creative problem solving in engineering design. *Comput. Aided Des.*, 41, 136–146.
Sanders, E. B. N. and Stappers, P. J. (2008) Co-creation and the new landscapes of design. *Codesign*, 4, 5–18.
Sanders, E. B.-N. and Westerlund, B. (2011) Experiencing, exploring and experimenting in and with co-design spaces. In *Proc. Nordic design research conference*. NORDES.
Sanders, E. B. N., Brandt, E. and Binder, T. (2010) A framework for organizing the tools and techniques of participatory design. In *ACM proceedings of the 11th biennial participatory design conference*, pp. 195–198, ACM.
Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action*, Vol. 5126. Basic Books.
Seidel, V. P. and Fixson, S. K. (2013) Adopting design thinking in novice multidisciplinary teams: the application and limits of design methods and reflexive practices. *J. Prod. Innov. Manag.*, 30, 19–33.
Simon, H. A. (1996) *The Sciences of the Artificial*. MIT Press.
Subramonyam, H., Drucker, S. M. and Adar, E. (2019) Affinity lens: data-assisted affinity diagramming with augmented reality. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, p. 398, ACM.
Terrenghi, L., Quigley, A. and Dix, A. (2009) A taxonomy for and analysis of multi-person-display ecosystems. *Pers. Ubiquitous Comput.*, 13, 583–598.
vom Brocke, J., Fettke, P., Gau, M., Houy, C., Maedche, A., Morana, S. and Seidel, S. (2017) Tool-support for design science research: design principles and instantiation (May 23, 2017). https://ssrn.com/abstract=2972803 or http://dx.doi.org/10.2139/ssrn.2972803. Last access: September 2020.
Warr, A. and O'Neill, E. (2005) Understanding design as a social creative practice. In *Proceedings of the 5th conference on creativity& cognition (C\&C '05)*, pp. 118–127. ACM, New York, USA.
Weber, M. (1991) The Nature of Social Action In Runciman, W. G. (ed.), *Weber: Selections in Translation*. Cambridge University Press.
Widjaja, W. and Takahashi, M. (2016) Distributed interface for group affinity-diagram brainstorming. *Concurr. Eng.*, 24, 344–358.
Zimmerman, J., Forlizzi, J. and Evenson, S. (2007) Research through design as a method for interaction design research in HCI. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 493–502, ACM.