Revisiting and Broadening the Meta-Design Framework
for End-User Development

Gerhard Fischer

Department of Computer Science, University of Colorado, Boulder, USA

gerhard@colorado.edu

Daniela Fogli

Department of Information Engineering, University of Brescia, Brescia, Italy

daniela.fogli@unibs.it

Antonio Piccinno

Department of Computer Science, University of Bari "Aldo Moro", Bari, Italy

antonio.piccinno@uniba.it

Abstract

Our contribution will review, analyze, discuss, and synthesize the research work done over the last 10 years exploring *meta-design* as a major framework for *end-user development* (EUD). The overriding perspective of our approach is grounded in the basic assumptions that (1) designers can prompt and support change in a community of practice, but they cannot predetermine it and (2) design and use mutually shape one another in iterative, social processes. The chapter argues and provides evidence that EUD should not be restricted to create new technologies but its most important and far-reaching impact will be to *transform cultures* by empowering all people to become active contributors in personally meaningful activities. The individual sections discuss and describe our basic framework, EUD applications in different domains, new conceptual developments that broadened the concept of meta-design, the identification of design trade-offs and drawbacks, and design guidelines. All of these activities have contributed to revisiting and broadening the meta-design framework for end-user development.

Keywords:

design, meta-design, participatory design, transformative cultures, cultures of participation, co-evolution, socio-technical systems, design methodologies, design guidelines, design drawbacks and trade-offs.

Table of Contents

List of Figures

List of Tables

1 Introduction

In earlier developments, *End-User Development (EUD)* was conceived as "a 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" [Lieberman et al., 2006a]. However, if one analyses the variety of proposals in the EUD field in international journals or in the proceedings of the five editions of the International Symposium on EUD held up so far, such a definition is too restrictive. Due to the many possibilities provided by technology (e.g., Web 2.0 and 3.0, Internet of Things, smart appliances and devices), the term EUD today should be conceived as a broader umbrella, including methods, situations, and socio-technical environments allowing and empowering end users "to express themselves and being independent of high-tech scribes".

One influential framework for supporting EUD is *meta-design* [Fischer & Giaccardi, 2006] empowering all stakeholders (including end users) to be actively engaged in the *continuous development* of personally meaningful socio-technical systems [Fischer & Herrmann, 2011].

Historically, software design was initially dominated by professionals. *Professional-dominated design* is a methodology founded on the belief that professional experts understand the users' needs [Rittel, 1984]. At design time, they create artifacts which users "have to live with" at use time. Whitle professional-dominated design has its place, it often creates systems that are at odds with users' interests, needs, and background knowledge. Successively, *user-centered design* [Norman & Draper, 1986] has been a major step forward to transcend the limitations of professional-dominated design by analyzing the interests, needs, and background knowledge of users and envisioning how users are likely to use an artifact. Then, to better cope with the users' needs and include them into the design, *participatory design* (PD) [Schuler & Namioka, 1993] focused on system development at design time by involving end users more deeply in the design process as co-designers by empowering them to propose and generate design alternatives themselves. It requires the social inclusion and active participation of the users at design time by bringing developers and users together to envision the contexts of use. But, despite the efforts at design time, systems need to evolve at use time to fit new needs, account for changing tasks, deal with a great variety of subjects, contexts and evolving needs, and incorporate new technologies, making meta-design a necessity.

This chapter explores how this conceptualization of EUD supported by meta-design was advanced over the last decade. As indicated in Figure 1, the different sections describe

- the impact of EUD on transforming cultures and some specific developments exploring and supporting this transformation process;
- a description of applications in different domains that were influenced by meta-design;
- new conceptual developments that broadened the concept of meta-design;
- the identification of design trade-offs and drawbacks that need to be carefully considered.

Figure 1 synthesizes the structure of our contribution; while design guidelines proposed by various scholars to realize the conceptual developments above and cope with trade-offs and drawbacks are finally presented.
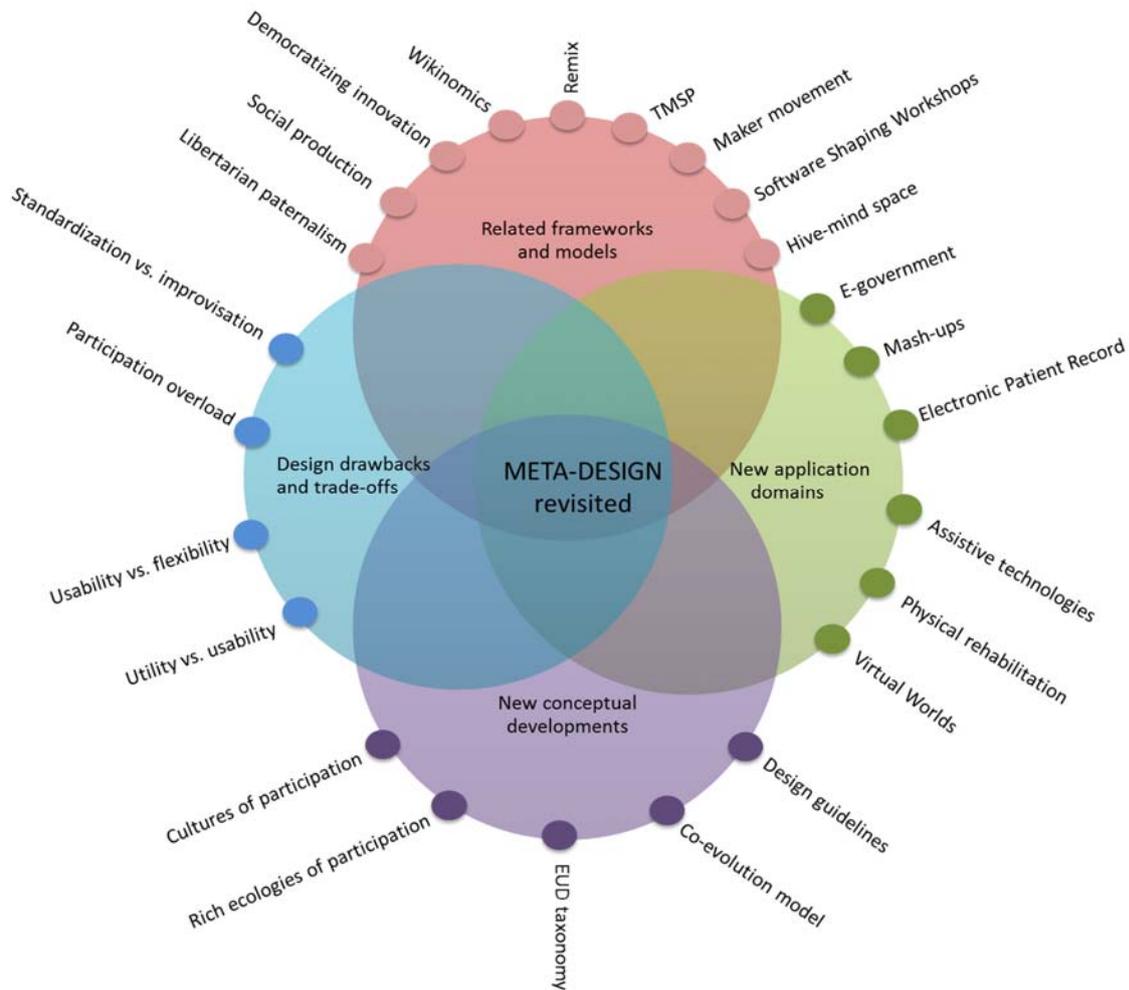
Figure 1: An Overview of the structure of our contribution

2   End-User Development: From Creating Technologies to Transforming Cultures

EUD is instrumental for "*the ability to reformulate knowledge, to express oneself creatively and appropriately, and to produce and generate information rather than simply to comprehend it*" [National Research Council, 1999]. It appeals to diverse audiences by supporting them in designing and building their own artifacts by situating computation in new contexts, by generating content, and by developing tools that democratize design, innovation, and knowledge creation [von Hippel, 2005]. This broad vision of EUD complements and transcends a *technological perspective* of EUD [Burnett & Scaffidi, 2013] that is closely related to:

- *End-User Programming (EUP)* that empowers and supports end users to program (with techniques such as: programming by demonstration, visual programming, scripting languages, and domain-specific languages) [Lieberman et al., 2006b]; and

- *End-User Software Engineering (EUSE)* that adds to EUP support for systematic and disciplined activities for the whole software lifecycle (including: reliability, efficiency, debugging support, and version control) [Burnett, 2009].

In other words, EUD provides the enabling conditions for putting owners of problems in charge by defining the technical and social conditions for broad participation in design activities [Fischer, 2013]. In this view, in a broader and updated definition, EUD is not only important in the context of software systems (the primary focus of EUP and EUSE), but it "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". Examples can be found in software systems, as well as in many other technological fields. In 3D modeling, for example, non-professional designers can today easily create their models and 3D print them to obtain the desired physical artifacts; or family members can easily create and control their smart home by using cheap off-the-shelf devices, smart objects and smartphones. Such a new perspective aims at seeing EUD more than a mere technique or tool, but includes conditions that allows end users to actually do EUD, easily and continuously, by taking advantage with respect to the traditional counterparts and finding this convenient and engaging during the time. This slightly differs, but includes, the definition given in the chapter by Ludwig et al. based on the concepts of the "gentle slope of complexity" [Ludwig et al., in print].

An early inspiration for conceptualizing EUD as a transformational culture was articulated by Ivan Illich with *convivial systems* envisioned to "give each person who uses them the greatest opportunity to enrich the environment with the fruits of his or her vision" [Illich, 1973].

2.1   Meta-Design: Framing and Supporting EUD as a Cultural Transformation

Meta-design derives from the observation that designing a system that can sufficiently anticipate all possible uses in advance (that is, when the system is created) is an impossible task. This idea led for example to the downfall of expert systems and of closed systems in general [Fischer & Scharff, 2000]. Closed systems typically create a sharp separation between design and use; however, providing functionality of interactive systems that is fixed when the system is created has important implications on how it will be used. As a consequence, it has been estimated that 40 to 60 percent of a system's cost over its lifetime is spent after the original system design is finished, not only to cope with the traditional need of "maintenance", but rather to carry out all those enhancement activities whose need is noticed by domain experts during the use of the system [Fischer & Scharff, 2000].

To this end, meta-design promotes the design of open systems that users can modify and evolve at use time [Henderson & Kyng, 1991]. As open systems are used, users will encounter mismatches and opportunities serving as potential sources for new insights and new understandings, and giving rise to the *co-evolution* between system and users [Costabile et al., 2007; Fogli & Piccinno, 2013a]. Therefore, meta-design as "design for design after design" is a fundamentally different design methodology compared for example to user-centered design and participatory design, which substantially promote "design for use before use" [Binder et al., 2011; Ye & Fischer, 2007]. Indeed, the latter approaches force all the design intelligence to the earliest part of the design process, when everyone knows the least about what is really needed. In a world that is not predictable, meta-design allows taking into account improvisation, evolution, and innovation by including the emergent and making it an opportunity for more creative and adequate solutions to problems [Fischer & Giaccardi, 2006]. For these reasons, meta-design is an interdisciplinary activity, bringing together multiple perspectives from different stakeholders and areas of expertise: from designers having specific knowledge in mathematics, computer science, and engineering, but who are ignorant of the

problem domain, to end users, who are experts in the problem domain, but ignorant of the domain of software solutions [Fischer, 2000]. Such a *"symmetry of ignorance"* [Rittel, 1984] (or "asymmetry of knowledge") can be an advantage for social creativity instead of an obstacle for design. This is particularly true for ill-defined problems, whose solution cannot be delegated to professional software developers, but requires that end users, as owners of problems, be put in charge. For example, in an *interview with a geoscientist* of the University of Colorado reported in [Fischer et al., 2009], it emerged that this end user, after a three months period in acquiring programming knowledge, spent an hour every day on average in the development of software for data analysis. This was necessary, since there was not any suitable software available and explaining the needs to a software developer was not possible due their variability as the research progressed. Therefore, the geoscientist, even though not considering himself a software developer, arrived at accepting software development as an essential task of his daily work. A meta-design approach would have probably been better suited to such a situation, by involving the geoscientist in the design of an open system to be shaped to his own needs at use time, without requiring him to spend three months learning a programming language. Such an approach would be even more useful in other domains (such as the medical one), in which domain experts are not interested and not motivated to invest time in learning technical skills that are not directly related to their work, rather they are willing to manipulate building blocks that make sense to their work practice [Cabitza & Simone, in print].

In summary, meta-design does not only encompass the study and development of enabling technologies for EUD, but also and above all sustaining a *cultural transformation* [Benkler, 2006; Fischer, 2013; von Hippel, 2005]. Therefore, the primary objective of meta-design is to allow and support end users to become end-user developers of their systems, where, nowadays, the term "system" denotes all the software and hardware components such as smartphones, smart watches, interactive displays, as well as the low cost devices that contribute to create the so-called Internet of Things [Barricelli & Valtolina, 2015; Cabitza et al., 2016].

2.2   Integrating and Relating Meta-Design with other Frameworks

Framing meta-design as a cultural transformation from *closed systems* (designed at design time and fixed at use time) to the design of *open systems* that users can modify and evolve at use time relates meta-design with a number of other frameworks summarized in Table 1 and briefly described below.

Table 1: Overview of Related Frameworks

Framework	Relationship to Meta-Design
Libertarian Paternalism	Providing evidence for the different ways how control can be divided between designers and end users
Social Production	Illustrating the possibilities and the power how individual autonomy can lead to interesting new artifacts
Democratizing Innovation	Allowing professional amateurs to do things because they want to do them
Wikinomics	Supporting mass collaboration
Remix	Indicating the intellectual property challenges with evolving artifacts
Technology Mediated Social Participation	Representing a model for new scientific communities
Maker Movement	Technology-based extension of a "Do-it-Yourself (DIY)" culture

Libertarian Paternalism. An interesting perspective and framework for EUD is provided by the book "Nudge: Improving Decisions about Health, Wealth, and Happiness" [Thaler & Sunstein, 2009]. The fundamental concept explored in the book is *libertarian paternalism*. The *libertarian* aspect of their approach *"lies in the straightforward insistence that, in general, people should be free to do what they like and to opt out of undesirable arrangements if they want to do so"* (p. 5). The *paternalistic* objective is grounded in the claim that *"it is legitimate for choice architects to try to influence people's behavior in order to make their lives longer, healthier, and better"* (p. 5). *Nudges* are defined by choice architects trying to motivate people to engage in certain actions and behavior. The role of choice architects is closely related to the role of meta-designers who create contexts in which users can provide content. By providing rich seeds [Fischer & Ostwald, 2002], they impose structures that affect the choices and actions of users, making a certain level of paternalism inevitable. The approach provides evidence and arguments about the importance of good defaults especially for activities that users consider personally of minimal relevance: users welcome a default rule and predefined functionality making life simpler and easier and protecting them from participation overload (see Section 5.2) and against their own mistakes.

Social Production. Benkler [Benkler, 2006] provides an elaborate framework and arguments that the most important aspect of the networked information economy is the possibility for reversing the *control focus* of the industrial information economy by enriching individual autonomy. This objective will be achieved by creating environments built less around control and more around facilitating action. He differentiates between passive (e.g.: television) and active (e.g.: open source, Wikipedia, Second Life) media (see Section 4.1). In active media, users are restricted to the role of consumers limited to selecting finished goods they can consume from a pre-defined range of options whereas in active media users are treated as active, creative human beings, capable of solving their own problems and building their own fantasies, alone and in affiliation with others.

Democratizing Innovation. Von Hippel [von Hippel, 2005] provides evidence from a broad range of different domains that users (supported by improvements in computer and communication technology)

increasingly can develop their own new artifacts and services. His case studies demonstrate that users (acting as professional amateurs - "pro-ams" -[Leadbeater & Miller, 2004]) who innovate can develop exactly what they want, rather than relying on designers of manufacturers to act as their agents or scribes. Additionally, individual users (acting as power users, local developers, and gardeners [Nardi, 1993]) do not have to develop everything they need on their own: they can benefit from innovations developed and freely shared by others.

Wikinomics. Tapscott and Williams [Tapscott & Williams, 2006] in their book "Wikinomics: How Mass Collaboration Changes Everything" explore what the Web 2.0 [O'Reilly, 2005] and mass collaboration [Cress et al., 2016] means for business and technology. They describe and analyze a number of success stories (including Wikipedia, open source, and LEGO) and introduce a number of concepts such as *"prosumers"* (indicating that users today often being "producers" in one context and "consumers" in another one). While the book analyzes success stories based on wiki-based environments, it does not mention that many efforts engaging users in participation (including their own effort that readers edit their book or write a chapter of it) did not succeed providing evidence for the empirical finding that "most wikis are dead at arrival".

Remix. Lessig [Lessig, 2008] in his book "Remix: Making Art and Commerce Thrive in the Hybrid Economy" analyzes participatory cultures (as promoted and supported by meta-design) from an *intellectual property perspective*. He distinguishes between two cultures: (1) a RO ("Read/Only") culture dominated by consumption, and (2) a RW (Read/Write) culture in which all people contribute to the re-creation and evolution of an existing culture by remixing existing components to create new ones. He discusses specifically the importance of "amateur creativity" [Leadbeater & Miller, 2004] in a RW culture (resembling the creativity of end users in an EUD culture) and how to avoid that this creativity is restricted by copyright regulation.

Technology Mediated Social Participation (TMSP). TMSP (http://tmsp.umd.edu) represents a movement (sponsored by the U.S. National Science Foundation) aiming to develop a scientific research agenda and educational recommendations for creating a cohesive community that generates the foundational science, engineering, and graduate training necessary for a new era of social participation technologies by empowering individuals to become active in local and global communities with a focus on exploring questions of how to motivate participation, increase social trust, and promote collaboration [Shneiderman, 2009].

Maker Movement. A basic belief and objective underlying the community of Makers is that the movement will end the monopoly of mass manufacturing just as the Internet ended the monopoly of mass media [Anderson, 2012]. It creates a culture that represents a technology-based extension (with 3D printers, laser cutters, microcontrollers, etc.) of the "Do-it-Yourself (DIY)" culture (as it has existed in numerous other domains such as home improvement activities). It emphasizes learning-through-doing in social environments by highlighting informal, networked, peer-led, and shared learning motivated by interest and fun.

2.3   *Methodologies and Models Extending the Meta-Design Framework*

The meta-design framework has inspired some methodologies for modeling and developing systems for EUD. Two of such extensions will be briefly described.

Software Shaping Workshop (SSW). The SSW is a design methodology based on the meta-design framework to model EUD-enabling systems [Costabile et al., 2006b; Costabile et al., 2007]. The idea

underlying this methodology is that software environments should be designed in analogy with artisan workshops, where traditional artisans, such as blacksmiths and joiners, extract the necessary tools to perform their activities from a repository, put them on a bench to do their work and finally set back in the repository those ones not useful anymore. In this way, artisans shape their work environments to their needs by using all and only the tools needed in a *specific* situation. By analogy, a SSW is designed as a virtual workshop, in which end users find a set of virtual tools useful to carry out their activities and shape their environment and tools by adapting them to their current needs, without the burden of using a traditional programming language. In SSW, end users manipulate objects and tools through a suitable domain-oriented visual language, and unwittingly create software programs [Costabile et al., 2008b], through which they later perform the necessary computations. In the SSW approach, users play two distinct roles, which should be supported by two types of SSWs. The former is that of end users who perform their work activities; the latter refers to domain experts, who are called on to design the SSW for end users in collaboration with other experts, e.g. software engineers, graphic designers, and HCI experts. End users will use *application SSWs*; whilst, the workshops used by domain experts to perform their design activities are called *system workshops.* The other members of the design team are supported by system workshops as well; all application and system workshops are customized to the culture and skills of their users. The designed interactive system results in a hierarchical network of SSWs, each specific for a community of users [Costabile et al., 2007]. The network encompasses three levels: 1) the *meta-design level*, where software engineers shape the tools and the system workshops to be used in the next level; 2) the *design level*, where HCI experts and domain experts use their system workshops to design, implement, and validate the application workshops devoted to end users; and 3) the *use level*, where end users of the different sub-communities use their application workshops and cooperate to achieve a task.

The SSW methodology encourages software designers to become *meta-designers* by involving all stakeholders in system design. In SSW, all stakeholders can make contributions that will be available to the other stakeholders for evaluation and feedback, in order to eventually converge to a common design. In light of these considerations, meta-design has been conceived in [Costabile et al., 2005] as "a technique, which provides the stakeholders in the design team with suitable languages and tools to foster their personal and common reasoning about the development of interactive software systems to support user work". This definition complements that of Fischer and Giaccardi, who conceive meta-design as a conceptual framework for defining and creating socio-technical infrastructures in which new forms of collaborative design can take place [Fischer & Giaccardi, 2006].

Hive-Mind Space (HMS) Model. The HMS model [Zhu, 2012; Zhu et al., 2011a; Zhu et al., 2010] is an evolution of the SSW methodology specifically oriented to support collaborative and creative design activities of multidisciplinary design teams. Hive Mind models in general focus on the collective intelligence (the hive mind) of people collaborating to pursue a common goal. They rely on the metaphor that people may collaborate within a community as a swarm of bees [Kelly, 1995], where each member of the community interacts locally, according to local rules, with a limited number of other community members, and the global behavior of the community emerges from local interactions. The HMS blends the general Hive Mind models and the SSW approach to support collaborative design and to foster creativity among design teams. The HMS model considers group activities, collective intelligence, and social creativity; whilst, from the SSW approach, the HMS model retains the three-level structure of the SSW network and enriches the workshops with tools for communication with other members of the same community and with other communities involved in the design collaboration. To this end, the HMS model introduces a central communication channel, called *digital boundary zone,* that allows the exchange and

management of so-called *digital boundary objects* [Zhu et al., 2010] consisting of software artifacts to represent what stakeholders mean during a collaboration activity. The HMS model supports a *Community of Interest* (CoIs) [Fischer, 2001] composed of a set of *Communities of Practice* (CoPs) [Wenger, 1998; Wenger et al., 2002]. Indeed, the HMS model, as well as the SSW approach, offers different workshops for various CoPs involved in collaborative design, each one localized to the CoP's culture, role and platform in use. Furthermore, the architecture proposed for the HMS model has an open under-development structure: further levels could be added to the network and at each level new CoPs can collaborate if needed. In order to evaluate the HMS model and provide some concrete guidelines for its implementation, the MikiWiki meta-design environment has been developed [Zhu et al., 2011b]. It is a structured programmable wiki that encompasses a hierarchical page organization made of pages and folder pages. Communication features are made available in MikiWiki as underdesigned "nuggets" (e.g. *chat*, *comment*, *wall*, and *notify* nuggets), which also represent the seeds [Fischer et al., 2001] for promoting system appropriation and modification. Users can easily start using and remixing existing nuggets, while power users may modify them, thus introducing new behaviors. MikiWiki has been applied in a variety of case studies, including the support of co-located meetings for the collaborative design of original mobile applications, such as the creativity barometer [Zhu & Herrmann, 2013].

The above conceptualizations define and support the role of meta-designers as professionals (1) using their own creativity to produce socio-technical environments in which other people can be creative and (2) defining the technical and social conditions for broad participation in design activities, which are as important as creating the software artifacts themselves.

3   Exploring Applications in Different Domains from a Meta-Design Perspective

This section presents some applications in different domains for which a meta-design perspective has been adopted. It is based on specific case studies (discussed in more detail in other publications) illustrating how meta-design has allowed modeling problems in innovative ways and putting end users in charge with the help of socio-technical mechanisms enabling EUD activities. They are all examples of system design to support human-problem interaction, rather than human-computer interaction (see design guidelines listed in Section 6). However, it is worth noticing that tools developed in the case studies were not deployed, but remained at an academic proofs-of-concept level; therefore, no consideration about consequences of long-term participation within related communities will be provided.

3.1   E-government

Meta-design and EUD techniques have been applied in the e-government domain pursuing two main objectives: (1) supporting municipality clerks in performing content authoring tasks by paying attention to the accessibility of the underlying web-oriented code [Fogli, 2009; Fogli et al., 2010]; and (2) supporting the same users in the creation of online e-government services devoted to citizens [Fogli & Parasiliti Provenza, 2011; Fogli & Parasiliti Provenza, 2012].

In the first case, a Content Management System (CMS) was extended to allow end users creating accessible web content (e.g., tables that could be easily accessed by visually impaired people) without being aware of performing software development, that is, creating proper HTML code. The extended CMS allowed users to accomplish tasks by simply editing content or selecting some content from available choices; the system then generated the correct HTML code by exploiting the content provided by the user. In the case of e-government service creation, a meta-design approach structured in two main phases was adopted: (1) a bottom-up activity was carried out, starting from the analysis of current services made available by the

municipality, with the aim of defining a meta-model of e-government services; and (2) an EUD environment that allowed civil servants to create instances of the meta-model was developed; this environment allowed creating XML documents, without being aware of that, and these documents were automatically interpreted to generate web applications that implemented e-government services (see Figure 2).

Both objectives were achieved after the observation of the daily tasks of end users (civil servants) and their usual approach to the use of computer systems; in this way, a fill-in form interaction style was provided in both solutions, given that administration tasks often consist in the compilation of paper-based forms. In the case of service creation, the interaction style was combined with a wizard design pattern that reflected the structure of the service to be created. Indeed, according to libertarian paternalism (Section 2.2), the civil servants should not have had so much freedom (and consequent responsibility) to modify the layout of the service pages or the structure of the service. In this way, service analysis and model-based design of EUD techniques remained in control of the software developers, as well as the consequent development of the fill-in forms that allowed civil servants to create online services.

In both cases, the rational for participation of civil servants consisted in becoming more independent from IT people for editing web site content and defining e-government services respectively. An EUD approach to this field was indeed been recognized by the domain experts as a way to improve work practice, in terms of effectiveness and efficiency, especially in those small or rural government agencies that cannot afford the budget necessary to employ software professionals or pay for software consultants. In this sense, EUD in the e-government domain, can be conceived as a social production framework [Benkler, 2006] (Section 2.2), which contributes to enrich individual autonomy by making people capable of solving their own problems.
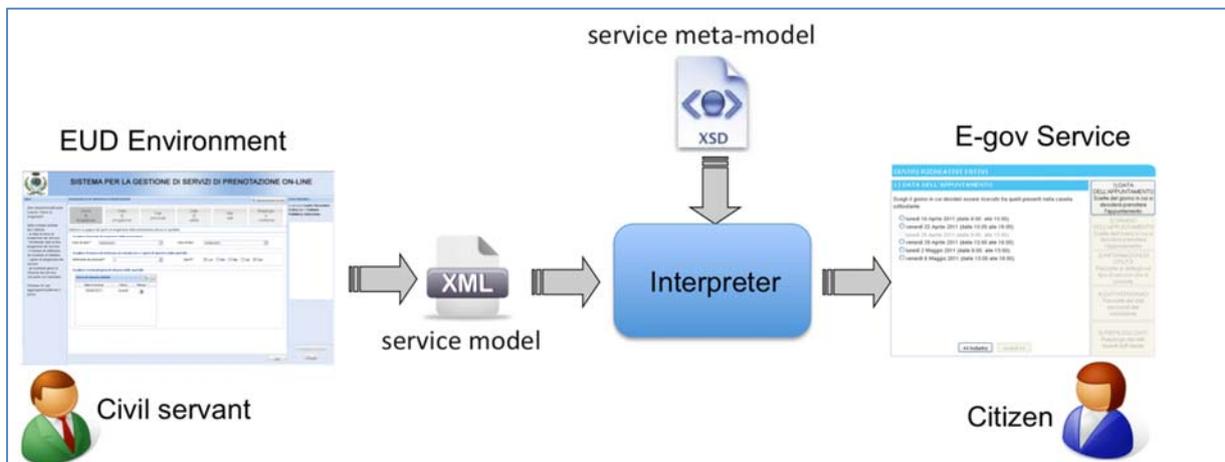
Figure 2: EUD approach to e-government service creation.

3.2   Mash-ups

From the end users' perspective, the development of web-based interactive systems is a demanding task. Perfectly in line with a democratized innovation [von Hippel, 2005], a common technique addressing this problem are mash-ups, i.e. the creation of Web applications through the composition of available Web services, without requiring skills in computer programming. Cappiello and colleagues present DashMash [Cappiello et al., 2011], an end-user oriented platform enabling inexperienced users to compose their own

mash-ups, in the form of dashboards exploiting company-internal services operating on data warehouses and public APIs.

The work in [Ghiani et al., 2011] proposes an approach, based on direct manipulation, which allows end users to create mash-ups by using web components extracted from existing web applications, such as Amazon or eBay. Other EUD tools for mash-ups are based on annotation features (e.g., [Avola et al., 2011; Dittrich et al., 2011]).

A recent mash-up platform, EFESTO, enables end users to create interactive workspaces by exploiting visual composition paradigms that accommodate the end-user mental model. With EFESTO end users create "live mash-ups" where information is dynamically extracted from heterogeneous data sources and visualized and manipulated into visual templates [Desolda et al., 2016]. Besides the composition paradigm for end users, one of the most relevant features of EFESTO is the possibility to exploit the data available in the Linked Open Data cloud. In fact, this mash-up platform allows end users to extend a Web service with the so-called "polymorphic data source" built on top of the Linked Open Data cloud. It is called polymorphic because it provides mutable information with respect to the data sources of which it is composed [Desolda, 2015].

Following a RW culture, instead of a RO one, mash-up platforms can be regarded as EUD environments able to foster user creativity in defining tools for personalized search and data analysis, and, at the same time, transform end users from consumers of a variety of Web services into producers of Web applications suitable to their work practice or personal needs.

3.3 Electronic Patient Records

Patient records are official artifacts with which medical and paramedical personnel preserve the memory or knowledge of facts and events that occurred in a hospital ward [Berg, 1999]. The patient record is a many-sided document: it is available to several different people, with different skills, background and expertise. They are not only physicians and nurses, but also patients and their relatives; thus patient records must have the ability to speak different "voices" to convey different meanings according to people using it [Cabitza & Simone, 2009]. A patient record is composed of a number of modules, each one containing specific patient data; hospital personnel in different wards are usually only interested in a subset of such modules. The employees use the modules to accomplish their specific tasks: for example, the reception staff records personal data at the acceptance of patients into the hospital; physicians examine other modules to make a diagnosis; nurses record medications and patients' parameters; and so on.

The development of the Electronic Patient Record (EPR) must take into account the various stakeholders involved in the EPR management and their different needs and personal (visual) languages. In the study reported in [Costabile et al., 2009], five different stakeholders have been identified: 1) *practice managers*, who decide the modules to be taken into account for the hospital; 2) *head physicians*, who are responsible for the specific EPR (subset of modules) for the ward; 3) *physicians*, using the EPR into their ward; 4) *nurses*, who fill the EPR; and (5) *administrative staff* who manages patient admission and billing. This is a typical situation that can be found in any hospital. In particular, the head physician has the responsibility of the definition of the EPR to be adopted in her/his ward, and currently must transfer her/his EPR specification to IT personnel or software consultants for successive implementation.

The SSW methodology described in Section 2.3 has been applied for the development of a novel concept of EPR, tailored to the ward's needs and to the different stakeholders' preferences and practices [Costabile et al., 2008a; Costabile et al., 2007]. In particular, the hierarchical and interconnected structure of SSWs has

allowed implementing the concept of libertarian paternalism (Section 2.2). At the *meta-design level* foreseen by the SSW methodology a team composed of software engineers, HCI experts and physicians designed the software environments for the different stakeholders, as well as the data modules, which are the basic components of the EPR. At the design level, software environments allowing each head physician to design the EPR for her/his ward by directly manipulating data modules in her/his software environment have been created, without depending anymore on "high-tech scribes", but sharing control on the system with them (see design guidelines in Section 6). In this case, physicians and nurses of a specific ward are the end users, while the head physician is the end-user developer in charge of creating the EPR for them.

3.4   *Supporting People with Cognitive Disabilities*

People with cognitive disabilities represent a *"universe of one"* problem [Carmien & Fischer, 2008; Carmien & Fischer, 2010]. They often will have several different disabilities and each specific combination of cognitive, motoric, sensory, and psychological impairments together define a need for deeply customized assistive technology such that a solution for one person will rarely work for another. The "*universe of one*" conceptualization includes the empirical finding that (1) *unexpected islands of abilities* exist: users can have unexpected skills and abilities that can be leveraged to ensure a better possibility of task accomplishment; and (2) *unexpected deficits of abilities* exist often occurring in otherwise high functioning individuals. Accessing and addressing these unexpected variations in skills and needs, particularly with respect to creating task support, requires an intimate knowledge of the user that *only caregivers* can provide.

The fundamental challenge derived from supporting the "universe of one" requirement is that it demands highly specific systems that we tried to achieve with a *meta-design approach.*

The *Memory Aiding Prompting System* (MAPS) [Carmien, 2006] represents a socio-technical environment based on a meta-design framework by providing the caregivers the design power to modify and evolve the technical systems according to the needs of individual users. To accommodate unexpected issues at use time, systems need to be *underdesigned* [Brand, 1995] by providing a context and a background against which situated cases can be interpreted thereby allowing the "owners of problems" to create the solutions themselves at use time.

Supporting people with cognitive disabilities represents a *multi-tiered proxy design problem*, since the end users (the persons with cognitive disabilities) cannot act as end-user developers, but only their caregivers can exercise this role. Some problems are characterized by the presence of end users that may not be able to express their needs, requiring additional stakeholders to articulate such needs and act as end-user developers on behalf of them and for them (see the proposed taxonomy of EUD activities in Section 4.3).

The challenge of MAPS was to design tools flexible enough to adapt to the unique needs of people with cognitive disabilities. The system was developed as a platform able to provide a prompting system for individuals with cognitive disabilities, along with an editing tool that allowed caregivers to design prompting scripts (Figure 3). It was aimed to support the independence and safety of people with cognitive disabilities in their daily activities, such as going to a grocery store or taking a bus. Participation was motivated by the fact that creating a specific environment by caregivers helped the people with cognitive abilities. Furthermore, like in other applications previously described, participation was facilitated by domain-oriented interaction support: indeed, designing a system for a unique use could be learnt and done with a reasonable time investment, thus coping with the trade-off between cultures of participation and participation overload (see Section 5.2).

MAPS included: (1) an editor to enable the caregiver (usually a family member) to edit, store, and reuse multimedia scripts for prompting instructions to support different daily tasks (i.e., sequences of video and verbal instructions); (2) a shared information space for storing script images and sounds, user and task modeling metadata, and a repository of tested scripts to be used as templates by other caregivers using the editor; and (3) a PDA-based device that prompted instructions to support the persons with cognitive disabilities in the accomplishment of their daily tasks.
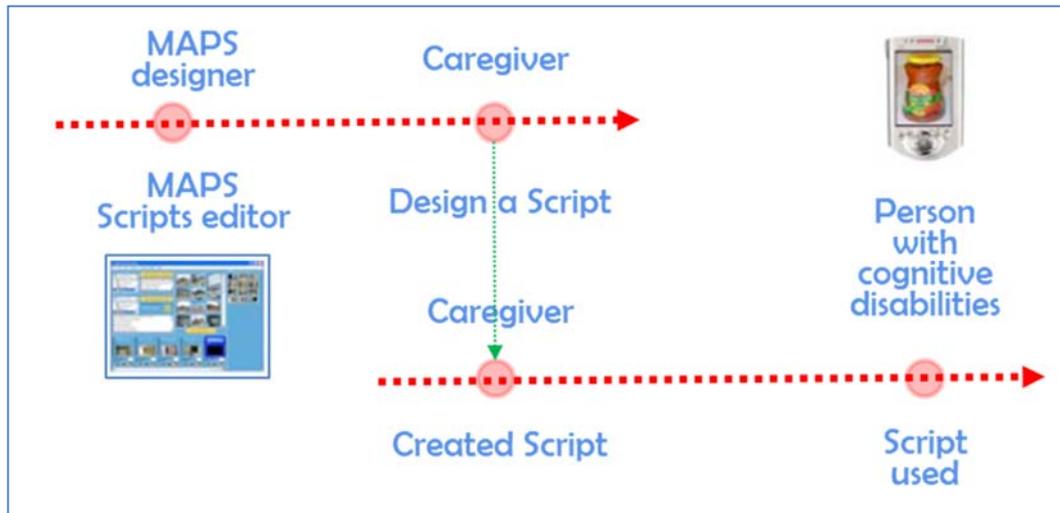

Figure 3: MAPS: empowering caregivers to act as designers

Multi-tiered proxy design problems push further in the direction of adopting meta-design approaches, since all involved stakeholders must be provided with suitable languages and tools to foster their participation in the development of software and hardware systems that support end users [Costabile et al., 2007].

3.5 *Physical rehabilitation*

The PhD research of Daniel Tetteroo [Tetteroo, 2013; Tetteroo, 2015] explored the design, development, and deployment of an end user extensible physical rehabilitation technology (called TagTrainer). The thesis provides a socio-technical perspective on the merits and issues related to the deployment of an EUD system in the context of physical rehabilitation therapy requiring personalized exercises, due to the high diversity in patients and their corresponding treatment needs.

TagTrainer is a tangible, interactive training platform for arm-hand rehabilitation exercises focused on relearning daily activities, such as manipulating cutlery and cups, in patients who experienced a stroke [Tetteroo et al., 2014]. It consists of four parts: 1) one or more interactive boards that can give audio-visual feedback and are able to detect RFID-tagged objects; 2) a collection of objects with RFID-tags attached to them; 3) the TagTrainer Exercise Creator, which supports therapists in creating and modifying exercises to be executed on the board; and 4) the TagTrainer Patient Interface, which allows therapists to manage personalized exercise programs by providing patients with feedback about their progress. From a preliminary study, it emerged that therapists are not information workers and usually do not rely on ICT for delivering treatment to patients; however, a cultural transformation could be fostered through

TagTrainer, which allowed therapists to become end-user developers, without the need to learn any programming language.

The PhD thesis describes the user-centered and participatory design process adopted for TagTrainer development; and it presents four studies in which TagTrainer was deployed in the context of rehabilitation clinics. The aim of these studies was to evaluate the acceptance of TagTrainer, to probe the feasibility of therapists as end-user developers of training exercises (supported by a meta-design environment based on a closely related architecture as illustrated in Figure 3 for MAPS), and to identify factors that influence the uptake of EUD practices. In particular, it has been observed how therapists varied in engagement as exercise creators: indeed, they played different roles, either (re-)using existing exercises or creating new ones, depending on their attitudes, age, and experience with information technology, as anticipated by cultures of participation theory introduced in [Fischer, 2011] (see Section 4.1). The research effort centered on TagTrainer identified some key challenges for enabling EUD practices (see [Tetteroo & Markopoulos, in print]) in clinical settings. This by aligning with the organization model, guiding end-user developers to ensure usability and software quality of their creations (see design tradeoffs discussed in Section 5.4), and providing features for retrieval and sharing of solutions created by end-user developers [Tetteroo et al., 2014].

3.6   *Virtual worlds*

Research conducted by Benjamin Koehne [Koehne et al., 2011] (closely related to the research by Mørch and colleagues [Caruso et al., 2015; Mørch et al., in print]) employed meta-design based theories in *virtual worlds* specifically by contrasting massively-multiplayer online role-playing games such as "Lord of the Rings Online" with open-ended virtual worlds such as "Second Life". The research employed ethnographic methods to explore the following research objectives:

- develop *additional examples* of meta-design for worlds that have no laws and boundaries;
- support the *empowerment of end users* that are not initially interested or motivated to conduct design practice;
- assess the *duality* between virtual worlds and meta-design, i.e.: how does meta-design affects practices in *virtual worlds* and vice versa; and
- analyze the *support for meta-design* in both unique environments, focusing on the benefits and shortcomings of the gaming-oriented and the open-ended environment under study.

Some of the findings of this research can be summarized as follows:

- Virtual worlds offer an opportunity to study the effects of collaboration on the way casual users move through rich ecologies of participation (see Section 4.2). Technical scaffolding systems alone are not sufficient. Instead, social community components need to make collaboration tools more accessible and attractive for casual users.
- Current open-ended virtual worlds (such as Second Life) provide means for extensions through source code modification which only technical people will be able to do. Additional mechanisms supporting meta-design would empower end users to extend these systems with additional capabilities.

4   New Conceptual Developments

This section explores some of the concepts related to the meta-design paradigm that emerged or were refined in the last decade. Table 2 briefly summarizes such concepts, while the next subsections discuss them in more detail.

Table 2: Concepts related to meta-design

Concept	Description
Cultures of participation	A shift from a consumer culture to cultures of participation, in which all people are provided with the socio-technical means for participation, has been observed in commercial systems and scientific works.
Rich ecologies of participation	Beyond the roles of consumer and designer, other roles of end users have been identified in literature; this led to identify richer ecologies of participation in software development.
Taxonomy of EUD activities	Different types of EUD activities have been identified and classified as individual EUD, public inward EUD and public outward EUD.
Co-evolution model	A model describing the interaction and co-evolution of users and systems is proposed; it takes into consideration all the different types of EUD foreseen in the EUD taxonomy.

4.1 Cultures of Participation

The rise in *social computing* (based on social production and mass collaboration) has facilitated a shift from *consumer cultures* (specialized in producing finished artifacts to be consumed passively) to *cultures of participation* (in which all people are provided with the means to participate and to contribute actively in personally meaningful problems) [Fischer, 2011; Jenkins, 2009].

Cultures of participation are facilitated and supported by a variety of different technological environments (such as: the participatory Web ("Web 2.0") [O'Reilly, 2005], table-top computing and domain-oriented design environments [Arias et al., 2016]); all of them contributing in different ways to the aims of engaging diverse audiences, enhancing creativity, sharing information, and fostering the collaboration among users acting as active contributors and designers. They democratize design and innovation [von Hippel, 2005] by shifting power and control towards users, supporting them to act as both designers and consumers ("prosumers") [Tapscott & Williams, 2006] and allowing systems to be shaped through real-time use.

The following design requirements derived from the meta-design framework support cultures of participation as follows:

- *Making changes must seem possible:* Contributors should not be intimidated and should not have the impression that they are incapable of making changes; the more users become convinced that changes are not as difficult as they think they are, the more they may be willing to participate.
- *Changes must be technically feasible:* If a system is closed, then contributors cannot make any changes; as a necessary prerequisite, there needs to be possibilities and mechanisms for extension.
- *Benefits must be perceived:* Contributors have to believe that what they get in return justifies the investment they make. The benefits perceived may vary and can include: professional benefits (helping for one's own work), social benefits (increased status in a community, possibilities for jobs), and personal benefits (engaging in fun activities).
- *The environments must support tasks that people engage in:* The best environments will not succeed if they are focused on activities that people do rarely or consider of marginal value.

- *Low barriers must exist to sharing changes:* Evolutionary growth is greatly accelerated in systems in which participants can share changes and keep track of multiple versions easily. If sharing is difficult, it creates an unnecessary burden that participants are unwilling to overcome.
- *Defining the role of meta-designers:* Meta-designers should use their own creativity in developing socio-technical environments in which other people can be put in charge. They must be willing to share control of how systems will be used, which content will be contained, and which functionality will be supported.

Cultures of participation support users as active contributors who can transcend the functionality and content of existing systems. By facilitating these possibilities, *control* is distributed among all stakeholders in the design process. There is evidence that shared control will lead to more innovation [von Hippel, 2005]: *"Users that innovate can develop exactly what they want, rather than relying on manufacturers to act as their (often very imperfect) agents."* (A similar argument surfaced in the interview with the geo-scientist described earlier). Cultures of participation erode monopoly positions held by professions, educational institutions, experts, and high-tech scribes [Fischer, 2002]. Drawbacks and trade-offs associated with cultures of participation are discussed in Section 5.2.

4.2   *Rich Ecologies of Participation*

Users and developers are commonly considered two distinct groups of people. Nowadays, with the Web 2.0 and the widespread use of web-based software systems, the sharp distinction between users and developers is quickly disappearing since users are more and more involved in the development of interactive (web-based) systems. An example is given by Google Sites and the many other similar platforms that today allow even naïve users to have an active role in the development of web sites suited to their needs. This results in a continuum ranging between end users as passive consumers to meta-designers [Fischer & Giaccardi, 2006]. In some cases, the same individuals play different roles: sometimes they are and want to be consumers, in other situations they prefer to be designers. Therefore, the terms "consumer" and "designer" cannot be considered as attributes of a person, but as roles in a specific context. More generally, several virtual organizations of end users exist in which richer ecologies of participants can develop according to their own needs [Preece & Shneiderman, 2009]. A deeper understanding of these ecologies leads to identify further roles beyond the traditional ones: *professional amateurs* [Leadbeater & Miller, 2004], *prosumers* [Tapscott & Williams, 2006], *power users, local developers,* and *gardeners* [Nardi, 1993], *bricolant-bricoleur* [Cabitza & Simone, 2015]. Such roles need to be exploited to create multi-faceted computational environments [Myers et al., 2006] tailored to the interests, needs and expertise of different stakeholders (see for instance the SSW methodology and HMS model discussed in Section 2.3), in order to also support *migration paths* among the different roles [Fischer et al., 2008].

The meta-designer role is usually intended for those professionals who are in charge of creating *"open systems at design time that can be modified by their users, acting as co-designers, requiring and supporting more complex interactions at use time"* [Fischer & Herrmann, 2011]. The work of Cabitza et al. [Cabitza et al., 2014b] extended this definition by introducing a distinct role for her/his social counterpart, that is the role of the *maieuta designer*[1]. On the one hand, the meta-designer is regarded more as a technical role: he/she is in charge of designing the EUD environment and all those tools by which end users could carry out their EUD activities. On the other hand, the maieuta designer can be considered as someone in charge of designing

---

[1] From "maieutic", the adjective relating to the method used by Socrates of eliciting knowledge in the mind of a person by interrogation and insistence on close and logical reasoning (http://dictionary.com).

the EUD-enabling environment by creating the *social conditions* for end users to become developers of their own system. These social conditions include: (1) sustaining end users to appropriate the design culture and the technical notions necessary for system development; (2) involving as many end users as possible in the process of continuous refinement of the system, by stimulating their participation and providing tools supporting their collaboration, and (3) facilitating the migration from the role of passive user to that of end-user developer. For these reasons, such a designer has been called a "maieuta", that is, someone who is able to apply the Socratic method of making people acquire notions, motivations, and self-confidence to undertake challenging tasks.

The maieuta designers are the persons who guarantee the long-term sustainability of an EUD project. Indeed, they should be identified within a community as someone who could make all community members become progressively independent from the IT professionals. One of their main tasks is to design (or better "co-design") initiatives in which to promote the EUD project and transfer to the community members the underlying values and concepts (i.e., empowerment, co-production, appropriation, cultures of participation, etc.). For example, the maieuta designer can devise simple mechanisms to foster participation and build a real culture of participation by creating proper motivation strategies, e.g. by exploiting gamification mechanisms [Benzi et al., 2015], and by creating collaboration infrastructures (e.g. by setting up social media associated with the EUD project to stimulate contributions and moderate communication among community members).

4.3   EUD Taxonomy

The new developments that occurred in the EUD field in the last ten years have led research scholars to analyze the new concepts, roles, and artifacts developed around EUD. To this aim, an EUD taxonomy has been proposed in [Cabitza et al., 2014a]. In this taxonomy, a classification of EUD into *individual EUD* and *public EUD* is proposed (see Figure 4). Individual EUD encompasses all those activities that are concerned with the creation, modification or extension of a software artifact for personal use only (therefore, individual EUD overlaps with End-User Programming [Myers et al., 2006]). Typical examples of individual EUD are spreadsheet programming for macro creation or modification, and scripting environments, like R and MATLAB, for statistical computing and data visualization (used by biologists, geologists and other scientists to analyze and display their data autonomously).

Public EUD denotes all those situations where end users either create or configure software artifacts that are used by *other* people belonging to the same community (because they are colleagues and co-workers) or belonging to a different community (because they work in a different department). In both cases, public EUD means that the outcome of the EUD activity is aimed at being shared and publicly available to others than the end user involved in the programming activity. The main difference between public and individual EUD is then the explicit intention behind the development effort: either making something intended to be shared or not.

Public EUD can be further specialized into *inward EUD* and *outward EUD*. In the former case, the people carrying out the EUD activity work for a community they also belong to, as in the case of Electronic Patient Records mentioned before [Costabile et al., 2008a; Costabile et al., 2007]. In inward EUD, activities are intended to support members of small teams and groups of people sharing sets of conventions, assumptions, and practices, i.e., communities of practice [Wenger, 1998; Wenger et al., 2002]. In this case, one member or a group of members of the community carries out the EUD activities, possibly engaging a conversation with software professionals over time, according to a *mutual development* approach [Andersen & Mørch, 2009]: they work for the proficiency of the community itself, given their (often tacit) knowledge

of the characteristics and skills of its members. In the outward EUD case, the EUD activity is aimed at building and improving tools that have to be used across different communities or, even, in other communities. Therefore, at least two communities (forming a community of interest [Fischer, 2001]), are involved and there is no guarantee that those who carry out EUD activities will also take advantage of the product of these activities. For example, in the e-government case (see Section 3.1) [Fogli & Parasiliti Provenza, 2011; Fogli & Parasiliti Provenza, 2012], the civil servants are in charge of creating e-government services for the citizens, whereas in the Memory Aiding Prompting System (see Section 3.2) [Carmien & Fischer, 2010], caregivers develop and customize prompting systems for persons with cognitive disabilities. Therefore, in public outward EUD, the quality of the software artifacts created by the end-user developers is more important than in individual and public inward EUD (see design tradeoffs in Section 5.5).

The objective of deepening the meaning implicit in the taxonomy is twofold. On the one hand, it suggests that there exist different "types" of end-user developments, and thus different meta-design frameworks, methods, and techniques should be considered for sustaining the activities of the end-user developers. On the other hand, it focuses on public EUD that is more and more pervading our daily life, but that has not received so far enough attention by the EUD community (this is true especially for outward EUD).
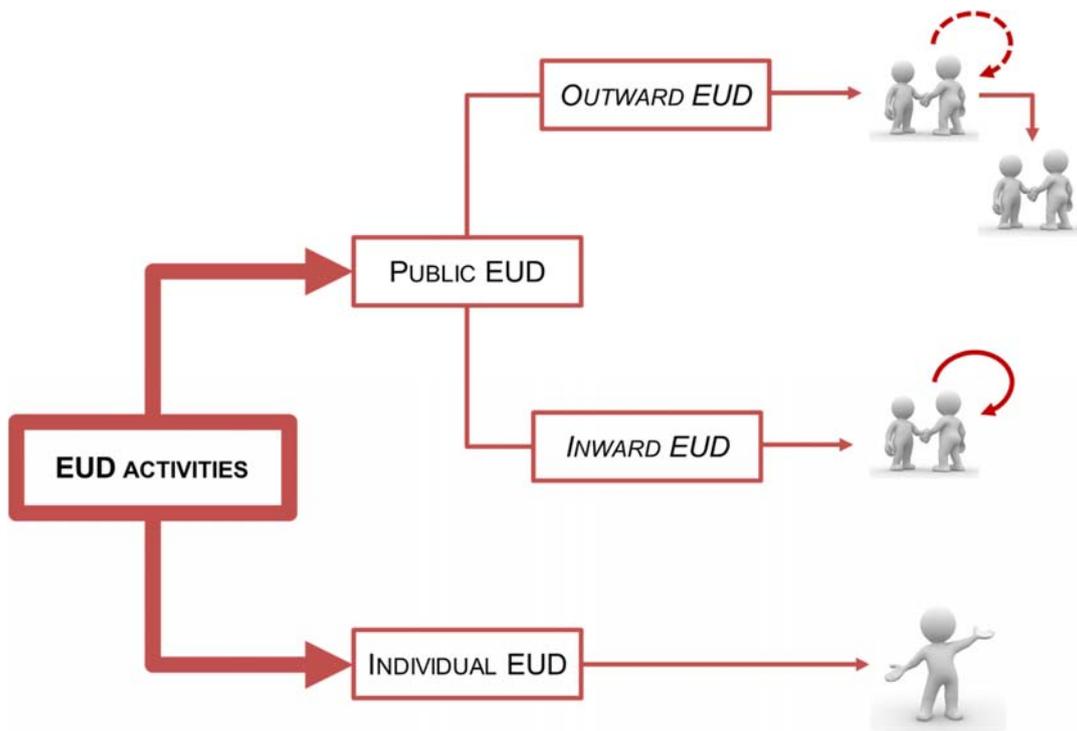
Figure 4: A Taxonomy of EUD Activities

4.4   Co-evolution Model

EUD encompasses techniques and applications that empower end users to develop and adapt systems creating foundations for the co-evolution of users and systems [Costabile et al., 2007]. To model this phenomenon, the *Interaction and Co-Evolution (ICE)* model (proposed in [Costabile et al., 2006a]) encompasses three cycles: the user-system interaction cycle, the task-artifact co-evolution cycle, and the

organization-technology co-evolution cycle. The inner cycle emphasizes that two different interpretation processes occur inside the human and the machine, which may become the source of usability problems and are related to the communication gap existing between users and designers at design time. The task-artifact co-evolution cycle recalls a well-known phenomenon described by Carroll and Rosson [Carroll & Rosson, 1992], namely that the software artifacts created to support some user's tasks usually suggest new possible tasks and that, to support these new tasks, new artifacts must be created. The outer cycle regards the co-evolution phenomenon according to a wider view: since technological advances provide designers with new possibilities for improving interactive systems once they are already in use, these possibilities may change users' work habits, thus making their social and work organization evolve itself with technology.

The ICE model is suitable to individual EUD, whilst an extended model, ICE², has been proposed in [Fogli & Piccinno, 2013a] to deal with public inward and outward EUD. Here, since end-user developers develop for others, they need to interact easily with an EUD environment to create, modify, or adapt software systems devoted to end users. Therefore, the ICE² model encompasses the end-user developer role, and three additional cycles model the mutual influence that systems and technology have with end-user developers and respective organizations.

Figure 5 illustrates the ICE² model presented in [Fogli & Piccinno, 2013a]. The left-hand side of the figure corresponds to the ICE model previously mentioned; it describes a process that is influenced by the specular process involving end-user developers, which is described by the right-hand side of the figure. The artifact can be regarded as a boundary object [Star, 1989] between the community of end users and that of end-user developers. It consists of the software system devoted to the end users and of the EUD tools used by end-user developers to generate and/or adapt the software system for end users. Different kinds of interaction between the two co-evolution processes occur at use time. They are discussed in [Fogli & Piccinno, 2013a] with the help of some case studies.
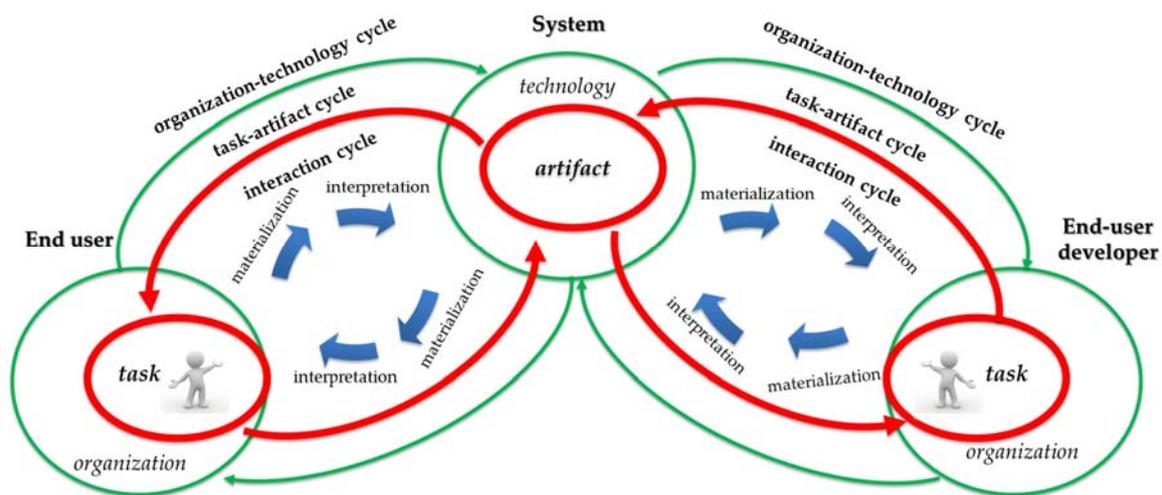

Figure 5: The ICE² model

5 Identifying Design Drawbacks and Trade-Offs Associated with Meta-Design

This section will examine some of the most important design trade-offs associated with meta-design. They are first summarized in Table 3 and then analyzed in the following sub-sections.

Table 3: Design trade-offs related to meta-design scenarios.

Trade-off	Description
Standardization vs. Improvisation	Tension between improvisation that encompasses innovation and creativity and the need for providing standard applications easy to distribute and maintain.
Cultures of participation vs. Information, participation and collaboration overload	Culture of participation can cause 1) information overload (by generating more information), 2) participation overload (by engaging people to act as active contributors), and 3) collaboration overload (by requiring coordination activities between the numerous contributors).
End-user driven evolution vs. Lack of continuity and synergy	End user-driven evolution is no guarantee for success because: (1) there is a lack of continuity over time, and (2) professional developers and users did not collaborate.
Usability of EUD products vs. flexibility of EUD tools	Guaranteeing the usability of the software artifacts created by end-user developers should be counterbalanced by a lack of flexibility in their creation/adaptation possibilities.
Utility vs. usability of EUD products	Some EUD projects should privilege utility whilst other should focus more on the usability of the results of EUD activities (i.e. EUD products); the "type" of EUD (see EUD taxonomy) may determine the choices in setting up the socio-technical conditions for EUD.

5.1 Standardization versus Improvisation

Meta-design creates inherent tensions, for example, between standardization and improvisation. The SAP Info (July 2003, page 33) argues to reduce the number of customer modifications ([Fischer & Giaccardi, 2006], p.446): "every customer modification implies costs because it has to be maintained by the customer. Each time a support package is imported there is a risk that the customer modification may have to be adjusted or re-implemented. To reduce the costs of such on-going maintenance of customer-specific changes, one of the key targets during an upgrade should be to return to the SAP standard wherever this is possible." Finding the right balance between standardization (which can suppress innovation and creativity) and improvisation (which can lead to a Babel of different and incompatible versions) has been noted as a challenge in open-source environments [Raymond & Young, 2001], in which forking has often led developers in different directions.

5.2 Transcending Consumer Cultures versus Information and Participation Overload

More and more information is available in the current digital society, coming from social networks, smart sensors and actuators, personal mobile systems, and web-based applications (e.g.: there are approx. 1.5 Million Apps available for iPhones and Android phones). Better support environments (such as: search engines, recommender systems, aggregators systems, and context-aware applications based on user and task models) are needed to cope with this *information overload* [Fischer, 2012].

Cultures of participation (see Section 4.1) have as a downside that they are contributing to a *participation overload* problem. People are more and more required to personally take care of their bank accounts, travel arrangements, retirement plans, etc. All these activities are manifestations of the DIY society. E-participation, e-democracy, wikis, open source software and EUD environments represent other situations where end users are asked to participate with their opinions, votes, interests, knowledge and expertise. In addition, users are asked to participate in peer-support communities, collaboratories, and crowdsourcing environments.

Participation overload is one of the most serious problems for meta-design. The following design trade-offs should be taken into account:

▪ for *personally irrelevant problems*, individuals should not be forced to act as an active contributors in cases in which they want to be consumers; people do not enjoy freedom of choice, and specifically in complex and unfamiliar domains, active choosing can be a burden, not a benefit (in the libertarian paternalism framework (see Section 2.2) the paternalism dimensions should be emphasized);

▪ for *personally meaningful problems*, individuals should not be restricted to act as consumers in situations where they want to be active contributors and decision makers. In such situations, environments should support engagement, promote learning, and participation. The rationale for this is provided by the following observation: *"The experience of having participated in a problem makes a difference to those who are affected by the solution. People are more likely to like a solution if they have been involved in its generation; even though it might not make sense otherwise"* [Rittel, 1984].

To cope with the participation overload problem, existing *methods* such as rich seeds, reuse, redesign, and remixing need to be further improved and extended. S*upport environments,* such as construction kits and domain-oriented design environments providing high-level building blocks and allowing users to express themselves in their own language should be studied and developed.

5.3   Lack of Continuity and Lack of Synergy

The Oregon Experiment, a housing experiment at the University of Oregon, instantiating the concept of end user-driven evolution, served as an interesting case study that end user-driven evolution is no guarantee for success [Alexander, 1975]. The analysis of its unsustainability indicated two major reasons: (1) there was a lack of continuity over time, and (2) professional developers and users did not collaborate, so there was a lack of synergy. These findings led to postulate the need for methods and techniques for maintaining high the interest in the user-driven evolution activities during the time and making developers and users engage in intense collaborations. The first aspect regards *long-term sustainability*, i.e., the need for taking high the interest of users when they are involved as contributors in the evolution, even after that any expert has left them alone.

With design rationale captured, communication enhanced, and end-user modifiability supported, developers have a rich source of information to evolve the system in the way users really need it. This leads to address a new issue that is the *perception of relative advantage*: for an innovation to have an impact on the daily life of its users it is important that these latter ones perceive the new thing as giving them a clear advantage with respect to the traditional counterparts whatever these are [Emani et al., 2012].

As has been outlined in Section 4.2, in a richer ecology of participants, the role of the maieuta designer is to make the community around an EUD system progressively more independent from the professionals [Cabitza et al., 2014b]. In other words, the maieuta designer is the person who is in charge of designing the EUD-enabling environment by creating the *social conditions* for end users to become developers of their

own system and guaranteeing the long-term sustainability of the end-user driven evolution of the system at hand.

The more and more involvement of end users in the evolution of the system brings different stakeholders, including consultants, designers internal to the organization and end users, to collaborate among themselves to shape the system. This means that such stakeholders need to face fundamental challenges in learning how to communicate and in building a shared understanding. Such a *lack of synergy* emerged also because during time, users and environments evolve. Collaborative design has emerged as a response to the needs felt by various organizations of adapting software to specific environments and users. Visual mediation mechanisms for collaborative design, development and evolution of software have been proposed to provide a means to improve communication and cooperation among all stakeholders involved in the design process [Ardito et al., 2011].

5.4   Usability of EUD Products versus Flexibility of EUD Tools

Several companies are more and more requiring information systems that are flexible enough to be adapted to the variety of their users, e.g. employees, business analysts, customers, and providers [Dörner et al., 2007]. EUD methods have been proposed as a solution for developing flexible systems, which can be adapted to the different needs directly on behalf of end users. However, such end users have no or few competencies in information technologies and often are not willing to acquire them; therefore, the software engineering community has raised many doubts on the effectiveness of the EUD approach, by underlying the important role played by skilled, professional software developers to guarantee software correctness, efficiency, maintainability, and security [Harrison, 2004]. For these reasons, the End-User Software Engineering (EUSE) research addresses the issue of software quality in EUD [Burnett & Scaffidi, 2013]. Literature works in this area propose methods oriented to non-professional software developers to carry out requirement analysis and specification, system design and reuse, code debugging, verification and testing. However, there is a further and important issue related to EUD, that is the usability of the software artifacts developed by end users [Fogli & Piccinno, 2013b]. EUSE mainly considers EUD as an activity targeted to create programs for personal rather than public use, thus distinguishing it from professional programming [Ko et al., 2011]; in this view, usability may not be an issue. However, if we consider public outward EUD (see Figure 4), where end-user developers create and adapt programs for others (sometimes belonging to another community), usability of the results of the EUD activity could become a problem. This is due to the fact that end-user developers are neither professional developers nor HCI experts, and may have a vague understanding of the usability concept. The idea proposed in [Fogli & Piccinno, 2013a; Fogli & Piccinno, 2013b] to cope with this problem does not conceive EUD as direct creation of code on behalf of end-user developers, but as the instantiation of a meta-model that represents a domain-dependent class of software applications for end users. Therefore, usability of the resulting environments is achieved through a meta-design activity, carried out by a multi-disciplinary design team including domain experts, which is aimed at defining the conceptual model of the resulting applications, the meta-model subsuming them and proper EUD tools for instantiating the meta-model in a easy way by end-user developers. To guarantee usability, the types of available EUD activities should be restricted and end-user developers' creativity could be limited as a consequence. In summary, a trade-off would emerge between the flexibility of EUD tools and the usability of systems resulting from EUD activities.

5.5   Utility versus Usability of EUD Products

Another design trade-off that has been observed in different contexts is that between *utility* and *usability* of software applications. In [Grudin, 1992], Grudin highlighted that in in-house and internal system development emphasis is usually put on utility since software artifacts are built according to the functional needs of the company; whilst, in commercial projects, usability is more important, since one of the priorities is to facilitate system acceptance by users and thus promote the diffusion of the system.

This trade-off is revived in meta-design and is concerned with utility and usability of EUD products. Recalling the EUD taxonomy described in Section 4.3, if one considers individual EUD and public inward EUD, the activities of end-user development encompass system adaptation and extension to increase effectiveness of the individual user and/or of his/her community; therefore, emphasis in these cases should be mainly put on utility. On the other hand, in public outward EUD, end-user developers create or adapt a software artifact by constantly taking into account the requests of the end users belonging to a different community.

As a consequence the socio-technical conditions that meta-designers (and maieuta designers) must create are different for these two situations. To support end-user developers in individual EUD and public inward EUD, meta-design must focus on the design of EUD tools and infrastructures for communication within the community [Dittrich et al., in print]; furthermore, proper training of the end-user developers must be taken into consideration, both in terms of programming methods and languages and of software engineering methods for guaranteeing software quality (as EUSE prescribes). On the other hand, in public outward EUD, not only tools supporting end-user developers must fit their characteristics, skills and background, but also the artifacts created for end users by the end-user developers must be usable. Thus, in this case, EUD techniques must be based on domain-specific concepts inspired to daily work practices and enriched with proper mechanisms for making artifact creation easier and code generation transparent, as well as for guaranteeing the creation of usable artifacts. Therefore, in this case, end-user developers should be supported no more with training in programming, but rather with user-friendly and visually engaging EUD systems, along with motivation strategies to foster end-user developers' participation in effectively doing EUD.

6   Design Guidelines

The above mentioned conceptual developments and analyzed trade-offs and drawbacks led the research on EUD, meta-design, and co-design to look for design guidelines for domain experts and for designing in use.

Among them, the set of guidelines for *domain experts* (representing end users who are experts in domains other than software design) proposed in [Fischer et al., 2009] include:

- *Support human-problem interaction*. Domain experts are interested in solving their problems, rather than in interacting with computers, therefore design must support human-problem interaction, rather than human-computer interaction. This can be achieved by increasing domain specificity, as in the case of domain-oriented design environments [Arias et al., 2016] and the various Software Shaping Workshops (see Section 2.3) [Costabile et al., 2006b; Costabile et al., 2007].

- *Underdesign for emergent behavior*. *Underdesign* [Brand, 1995] relates to meta-design, in that it creates design spaces where users can create solutions suitable to their contingent needs allowing them to explore problems and solutions not envisioned at design time. Systems should be underdesigned so

that users do not treat them as finished products but view them as continuous beta versions that are open to incorporate emergent design behaviors during use.

- *Enable legitimate peripheral participation*. Newcomers to a community must be able to engage in *legitimate peripheral participation* [Lave & Wenger, 1991] through transparent policies and procedures for incorporating user contributions into the software systems. To this aim, the system architecture must support rich ecologies of participation (see Section 4.2), in order to support newcomers in progressively difficult and independent tasks, so that they can start participating peripherally and move on gradually to take charge of more difficult tasks.

- *Share control*. Meta-designers must share control on the system with the participating users. Users can play different roles, depending on their level of involvement, and thus have their own responsibility and authority. When users change their roles in the community by making frequent and substantial contributions [Fischer et al., 2008], they should be granted with more authority in the decision-making process that shapes the system. This helps sustain user participation and system evolution: users become stakeholders, acquire ownership in the system, and will likely make further contributions; granting authority attracts (new) users who want to influence system development and encourages them to contribute.

- *Reward and recognize contributions*. Fostering user participation in system use and development requires paying attention to users' motivations. Human beings do not act only for material gain but also for psychological well-being, social integration and connectedness, social capital, recognition, and reputation [Benkler & Nissenbaum, 2006]. Motivation is derived from users' satisfaction in their involvement by shaping the software system to solve their problems, and can be positively reinforced and amplified when the community's social structure and conventions reward and recognize the contributions of users.

- *Foster reflective communities*. The knowledge relevant to a complex design problem is usually distributed among many domain experts. Fostering reflective communities becomes a fundamental goal of meta-design [Arias et al., 2016] and can be achieved by creating proper mechanisms for collaboration among domain experts, who may bring controversial perspectives to the problem solution. This requires facilitating a shared understanding among domain experts, by allowing them to bring their different knowledge sources and equally contribute to the creation of new insights, new ideas, and new artifacts.

The set of guidelines for designing in use [Maceli & Atwood, 2011] are derived from the literature on *co-design* (in which designers and users collaboratively are shaping a system over time) and include:

- *Connect with other people with similar needs, both nearby and far away.* This principle would like to encourage designers to focus on how users can use the system to connect with other people, and how they might extend the system to satisfy this requirement.

- *Reach out and converse with other people in real-time, while they are using the system*. This principle emphasizes how users can have live experiences and conversations with other people, who could be not only other users within or outside their community, but also designers or users acting as designers. This principle also suggests paying attention to the emergent use of chat and microblogging tools to facilitate backchannel conversations.

- *Combine it with other tools and systems they use regularly*. The idea suggested by this principle is regarding a system as only one piece of a larger, evolving puzzle and not assume it to be something on which the user is totally focused for all the time. Users interact with several tools and systems on a daily basis and often at the same time. The possibility offered by mashups (see Section 3.2) to combine

different data sources and programming interfaces to create novel tools is suggested as a possible way to address this need.

- *Begin using it quickly, without a lot of help or instruction*. This principle is related to the general and traditional theme of system usability; it is aimed at envisioning ways in which users could begin using a system quickly, by potentially becoming able to act as designers with a limited effort.
- *Tailor it to their personalized needs.* This principle regards tailorability as fundamental to support users to act as designers. Both adaptivity (the system may tailor itself on the basis of recurrent user interactions) and adaptability (the user consciously performs tailoring actions) are considered successful solutions to provide users with the necessary tools for system modification when new needs arise during its use.

With respect to the meta-design guidelines mentioned before, these principles are especially aimed at providing a frame for users and designers to communicate changes across the system lifecycle to foster co-evolution of system and users.

7 Conclusion

Providing all citizens with the means to become co-creators of new ideas, knowledge, and products in personally meaningful activities presents one of the most exciting innovations and transformations with profound implications in the years to come. This objective characterizes *the vision behind EUD as a cultural transformation*, which complements and transcends the traditional technological perspective of EUD [Burnett & Scaffidi, 2013] mainly oriented to engage and support people in programming activities.

To make this vision a reality, the EUD research community needs to establish (1) new theoretical frameworks (the chapter by Clarisse S. de Souza in this volume argues that Semiotic Engineering can provide a unified theoretical framing for various EUD-related topics [de Souza, in print]), (2) new discourses and shared languages about concepts, assumptions, values, stories, metaphors, design approaches, and (3) new learning theories, such as those aimed at promoting computational thinking [Brennan & Resnick, 2012; Kafai, 2016]. End users (by claiming ownership in personally meaningful problems) should be empowered to design, build, and evolve their own artifacts and meta-designers should create environments to foster *cultures of participation.* These objectives will support all citizens to situate computation in new cultural and material contexts with the support of socio-technical environments that democratize design.

New information and communication technologies have been heralded as the major driving forces behind innovations in working, learning, and collaborating. But many approaches have had only a minor impact by being conceptualized primarily as technology-centered developments. *Technology alone does not determine social structure: it creates feasibility spaces for new social and cultural practice.* Changes in complex environments are not only dictated by technology; rather, they are the result of an incremental shift in human behavior and social organization and, as such, require the *co-design of social and technical systems*.

In an EUD culture supported by meta-design, individuals acting as designers will acquire a new mindset: they are no longer passive receivers of knowledge, but instead are active researchers, designers, and communicators of knowledge. Knowledge is no longer handed down from above, but instead is constructed collaboratively in the contexts of work.

Meta-design provides the enabling conditions for putting owners of problems in charge by defining the technical and social conditions for broad participation in design activities. It addresses the challenges of fostering new mindsets, new sources of creativity, and cultural changes to create foundations for innovative

societies. The foremost objective of meta-design is empowering humans (albeit not all of them, not at all times, not in all contexts) to be and act as designers in personally meaningful activities [Fischer, 2011].

8   Acknowledgements

9   References

Alexander, C. (1975). *The Oregon Experiment*: Oxford University Press.

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.

Ardito, C., Barricelli, B. R., Buono, P., Costabile, M. F., Piccinno, A., Valtolina, S., & Zhu, L. (2011). Visual Mediation Mechanisms for Collaborative Design and Development. In C. Stephanidis (Ed.), *Universal Access in Human-Computer Interaction. Design for All and eInclusion* (Vol. 6765, pp. 3-11). Berlin / Heidelberg: Springer. doi: 10.1007/978-3-642-21672-5_1.

Arias, E. G., Eden, H., & Fischer, G. (2016). *The Envisionment and Discovery Collaboratory (EDC): Explorations in Human-Centered Informatics*. San Rafael, California: Morgan & Claypool Publishers.

Avola, D., Bottoni, P., & Genzone, R. (2011). Light-Weight Composition of Personal Documents from Distributed Information. In M. F. Costabile, Y. Dittrich, G. Fischer & A. Piccinno (Eds.), *End-User Development* (Vol. 6654, pp. 221-226): Springer Berlin Heidelberg. doi: 10.1007/978-3-642-21530-8_17.

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.

Benkler, Y. (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven: Yale University Press.

Benkler, Y., & Nissenbaum, H. (2006). Commons-based Peer Production and Virtue. *Political Philosophy, 14*(4), 394-419

Benzi, F., Cabitza, F., Fogli, D., Lanzilotti, R., & Piccinno, A. (2015). Gamification Techniques for Rule Management in Ambient Intelligence. In B. De Ruyter, A. Kameas, P. Chatzimisios & I. Mavrommati (Eds.), *Ambient Intelligence* (Vol. 9425, pp. 353-356): Springer International Publishing. doi: 10.1007/978-3-319-26005-1_25.

Berg, M. (1999). Accumulating and Coordinating: Occasions for Information Technologies in Medical Work. *Computer Supported Cooperative Work, 8*(4), 373-401. doi: 10.1023/A:1008757115404

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

Brand, S. (1995). *How Buildings Learn: What Happens After They're Built*. New York: Penguin Books.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Proc. of Annual Meeting of the American Educational Research Association (AERA'12), Vancouver, Canada.

Burnett, M. (2009). What is End-User Software Engineering and Why Does It Matter? In V. Pipek, M. B. Rossen, B. deRuyter & V. Wulf (Eds.), *End-User Development* (pp. 15-28). Heidelberg: Springer.

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. (2016). Rule-based tools for the configuration of ambient intelligence systems: a comparative user study. *MULTIMEDIA TOOLS AND APPLICATIONS*, 1-21. doi: 10.1007/s11042-016-3511-2

Cabitza, F., Fogli, D., & Piccinno, A. (2014a). "Each to His Own": Distinguishing Activities, Roles and Artifacts in EUD Practices. In L. Caporarello, B. Di Martino & M. Martinez (Eds.), *Smart Organizations and Smart Artifacts* (Vol. 7, pp. 193-205). Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-07040-7_19.

Cabitza, F., Fogli, D., & Piccinno, A. (2014b). Fostering participation and co-evolution in sentient multimedia systems. *Journal of Visual Languages and Computing, 25*(6), 684-694. doi: 10.1016/j.jvlc.2014.10.014

Cabitza, F., & Simone, C. (2009). LWOAD: A Specification Language to Enable the End-User Develoment of Coordinative Functionalities. In V. Pipek, M. B. Rosson, B. de Ruyter & V. Wulf (Eds.), *End-User Development* (Vol. 5435, pp. 146-165). Berlin / Heidelberg: Springer. doi: 10.1007/978-3-642-00427-8.

Cabitza, F., & Simone, C. (2015). Building Socially Embedded Technologies: Implications About Design. In V. Wulf, K. Schmidt & D. Randall (Eds.), *Designing Socially Embedded Technologies in the Real-World* (pp. 217-270): Springer London. doi: 10.1007/978-1-4471-6720-4_11.

Cabitza, F., & Simone, C. (in print). Malleability in the hands of end users. In F. Paterno & V. Wulf (Eds.), *New Perspectives in End User Development*: Springer Int.

Cappiello, C., Matera, M., Picozzi, M., Sprega, G., Barbagallo, D., & Francalanci, C. (2011). DashMash: A Mashup Environment for End User Development. In S. Auer, O. Díaz & G. A. Papadopoulos (Eds.), *Web Engineering* (Vol. 6757, pp. 152-166). Berlin / Heidelberg: Springer. doi: 10.1007/978-3-642-22233-7_11.

Carmien, S. (2006). Socio-Technical Environments Supporting Distributed Cognition for Persons with Cognitive Disabilities (PhD Thesis, Computer Science Department, University of Colorado, Boulder). from http://www.scarmien.com/papers/dissertation_sm.pdf

Carmien, S., & Fischer, G. (2008). Design, Adoption, and Assessment of a Socio-Technical Environment Supporting Independence for Persons with Cognitive Disabilities *Proceedings of CHI 2008: ACM Conference on Human Factors in Computing Systems* (pp. (in press)). New York, NY, USA: ACM Press.

Carmien, S., & Fischer, G. (2010). Beyond Human-Computer Interaction: Meta-Design in Support of Human Problem-Domain Interaction. In M. Banich & D. Caccamise (Eds.), *Generalization of Knowledge: Multidisciplinary Perspectives* (pp. 331-349). New York: Psychology Press.

Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems, 10*(2), 181-212. doi: 10.1145/146802.146834

Caruso, V., Hartley, M. D., & Mørch, A. I. (2015). End-User Development in Second Life: Meta-design, Tailoring, and Appropriation. In P. Díaz, V. Pipek, C. Ardito, C. Jensen, I. Aedo & A. Boden (Eds.), *End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings* (pp. 92-108). Cham: Springer International Publishing. doi: 10.1007/978-3-319-18425-8_7.

Costabile, M. F., Fogli, D., Marcante, A., Mussio, P., Parasiliti Provenza, L., & Piccinno, A. (2008a). Designing Customized and Tailorable Visual Interactive Systems. *International Journal of Software Engineering and Knowledge Engineering, 18*(3), 305-325. doi: 10.1142/S0218194008003702

Costabile, M. F., Fogli, D., Marcante, A., & Piccinno, A. (2006a). *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., Fogli, D., Mussio, P., & Piccinno, A. (2005). *A meta-design approach to End-User Development.* Proc. of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Dallas, Texas, USA, September 20-24, 308-310. doi: 10.1109/VLHCC.2005.7.

Costabile, M. F., Fogli, D., Mussio, P., & Piccinno, A. (2006b). End-User Development: the Software Shaping Workshop Approach. In H. Lieberman, F. Paternò & V. Wulf (Eds.), *End User Development* (Vol. 9, pp. 183-205). Dordrecht, The Netherlands: Springer. doi: 10.1007/1-4020-5386-X_9.

Costabile, M. F., Fogli, D., Mussio, P., & Piccinno, A. (2007). Visual Interactive Systems for End-User Development: a Model-based Design Methodology. *IEEE Transactions on System Man and Cybernetics Part A-Systems and Humans, 37*(6), 1029-1046. doi: 10.1109/TSMCA.2007.904776

Costabile, M. F., Mussio, P., Parasiliti Provenza, L., & Piccinno, A. (2008b). *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., Piccinno, A., Ardito, C., Barricelli, B. R., & Lanzilotti, R. (2009). *End-User Development in the Medical Domain.* Proc. of 15th International Conference on Distributed Multimedia Systems (DMS), San Francisco Bay, CA, USA, September 10-12, 10-15.

Cress, U., Jeong, H., & Moskaliuk, J. (Eds.). (2016). *Mass Collaboration and Education.* Heidelberg: Springer.

de Souza, C. S. (in print). 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 Int.

Desolda, G. (2015). Enhancing Workspace Composition by Exploiting Linked Open Data as a Polymorphic Data Source. In E. Damiani, J. R. Howlett, C. L. Jain, L. Gallo & G. De Pietro (Eds.), *Intelligent Interactive Multimedia Systems and Services* (pp. 97-108). Cham: Springer International Publishing. doi: 10.1007/978-3-319-19830-9_9.

Desolda, G., Ardito, C., & Matera, M. (2016). EFESTO: A Platform for the End-User Development of Interactive Workspaces for Data Exploration. In F. Daniel & C. Pautasso (Eds.), *Rapid Mashup Development Tools: First International Rapid Mashup Challenge, RMC 2015, Rotterdam, The Netherlands, June 23, 2015, Revised Selected Papers* (pp. 63-81). Cham: Springer International Publishing. doi: 10.1007/978-3-319-28727-0_5.

Dittrich, Y., Bolmsten, J., & Ericksson, J. (in print). EUD and infrastructuring – sustaining organizational innovation capabilities. In F. Paterno & V. Wulf (Eds.), *New Perspectives in End User Development*: Springer Int.

Dittrich, Y., Madsen, P., & Rasmussen, R. (2011). Really Simple Mash-Ups. In M. F. Costabile, Y. Dittrich, G. Fischer & A. Piccinno (Eds.), *End-User Development* (Vol. 6654, pp. 227-232): Springer Berlin Heidelberg. doi: 10.1007/978-3-642-21530-8_18.

Dörner, C., Heß, J., & Pipek, V. (2007). *Improving Information Systems by End User Development: A Case Study.* Proc. of European Conference on Information Systems (ECIS), St. Gallen, Switzerland, Sept. 10-14, 783-794.

Emani, S., Yamin, C. K., Peters, E., Karson, A. S., Lipsitz, S. R., Wald, J. S., Williams, D. H., & Bates, D. W. (2012). Patient Perceptions of a Personal Health Record: A Test of the Diffusion of Innovation Model. *Journal of Medical Internet Research, 14*(6). doi: 10.2196/jmir.2278

Fischer, G. (2000). Symmetry of ignorance, social creativity, and meta-design. *Knowledge-Based Systems, 13*(7–8), 527-537. doi: 10.1016/S0950-7051(00)00065-4

Fischer, G. (2001). *Communities of Interest: Learning through the Interaction of Multiple Knowledge Systems.* Proc. of 24th Annual Information Systems Research Seminar In Scandinavia (IRIS'24), Ulvik, Norway, 1-14.

Fischer, G. (2002). Beyond 'Couch Potatoes': From Consumers to Designers and Active Contributors. *FirstMonday, 7*(12), http://firstmonday.org/issues/issue7_12/fischer/. doi: http://dx.doi.org/10.5210/fm.v7i12.1010

Fischer, G. (2011). Understanding, fostering, and supporting cultures of participation. *Interactions, 18*(3), 42-53. doi: 10.1145/1962438.1962450

Fischer, G. (2012). Context-Aware Systems: The 'Right' Information, at the 'Right' Time, in the 'Right' Place, in the 'Right' Way, to the 'Right' Person. In G. Tortora, S. Levialdi & M. Tucci (Eds.), *Proceedings of the Conference on Advanced Visual Interfaces (AVI 2012)* (pp. 287-294). Capri, Italy (May): ACM.

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., & 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., Grudin, J., McCall, R., Ostwald, J., Redmiles, D., Reeves, B., & Shipman, F. (2001). Seeding, Evolutionary Growth and Reseeding: The Incremental Development of Collaborative Design Environments. In G. M. Olson, T. W. Malone & J. B. Smith (Eds.), *Coordination Theory and Collaboration Technology* (pp. 447-472). Mahwah, NJ: Lawrence Erlbaum Associates.

Fischer, G., & Herrmann, T. (2011). Socio-Technical Systems: A Meta-Design Perspective. *International Journal of Sociotechnology and Knowledge Development (IJSKD), 3*(1), 1-33. doi: 10.4018/jskd.2011010101

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.

Fischer, G., Piccinno, A., & Ye, Y. (2008). The Ecology of Participants in Co-Evolving Socio-Technical Environments. In P. Forbrig, Paternò, F. (Ed.), *Engineering Interactive Systems (Proceedings of 2nd Conference on Human-Centered Software Engineering)* (Vol. LNCS 5247, pp. 279-286). Heidelberg: Springer.

Fischer, G., & Scharff, E. (2000). *Meta-design: design for designers.* Proc. of 3rd conference on Designing interactive systems: processes, practices, methods, and techniques, New York City, New York, USA, 396-405. doi: 10.1145/347642.347798.

Fogli, D. (2009). End-User Development for E-Government Website Content Creation. In V. Pipek, M. B. Rosson, B. de Ruyter & V. Wulf (Eds.), *End-User Development* (Vol. 5435/2009, pp. 126-145). Berlin / Heidelberg: Springer. doi: 10.1007/978-3-642-00427-8.

Fogli, D., Colosio, S., & Sacco, M. (2010). Managing accessibility in local e-government websites through end-user development: a case study. *UNIVERSAL ACCESS IN THE INFORMATION SOCIETY, 9*(1), 35-50. doi: 10.1007/s10209-009-0158-z

Fogli, D., & Parasiliti Provenza, L. (2011). End-User Development of e-Government Services through Meta-modeling. In M. F. Costabile, Y. Dittrich, G. Fischer & A. Piccinno (Eds.), *End-User Development* (Vol. 6654, pp. 107-122). Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-21530-8_10.

Fogli, D., & Parasiliti Provenza, L. (2012). A meta-design approach to the development of e-government services. *Journal of Visual Languages and Computing, 23*(2), 47-62. doi: 10.1016/j.jvlc.2011.11.003

Fogli, D., & Piccinno, A. (2013a). 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.

Fogli, D., & Piccinno, A. (2013b). Enabling Domain Experts to Develop Usable Software Artifacts. In P. Spagnoletti (Ed.), *Organizational Change and Information Systems* (Vol. 2, pp. 419-428). Berlin Heidelberg: Springer. doi: 10.1007/978-3-642-37228-5_41.

Ghiani, G., Paternò, F., & Spano, L. D. (2011). Creating Mashups by Direct Manipulation of Existing Web Applications. In M. F. Costabile, Y. Dittrich, G. Fischer & A. Piccinno (Eds.), *End-User Development* (Vol. 6654, pp. 42-52). Berlin / Heidelberg: Springer. doi: 10.1007/978-3-642-21530-8_5.

Grudin, J. (1992). Utility and usability: research issues and development contexts. *Interacting with Computers, 4*(2), 209-217. doi: 10.1016/0953-5438(92)90005-Z

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

Henderson, A., & Kyng, M. (1991). There's No Place Like Home: Continuing Design in Use. In J. Greenbaum & M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems* (pp. 219-240). Hillsdale, NJ, USA: Lawrence Erlbaum Associates, Inc.

Illich, I. (1973). *Tools for Conviviality*. New York: Harper and Row.

Jenkins, H. (2009). *Confronting the Challenges of Participatory Cultures: Media Education for the 21st Century*. Cambridge, MA: MIT Press.

Kafai, Y. B. (2016). From computational thinking to computational participation in K--12 education. *Communications of the ACM, 59*(8), 26-27. doi: 10.1145/2955114

Kelly, K. (1995). *Out of control: the new biology of machines, social systems, and the economic world*. Boston, US: Addison-Wesley Longman Publishing Co., Inc.

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

Koehne, B., Redmiles, D., & Fischer, G. (2011). Extending the Meta-design Theory: Engaging Participants as Active Contributors in Virtual Worlds. In M. F. Costabile, Y. Dittrich, G. Fischer & A. Piccinno (Eds.), *End-User Development (Third International Symposium, Torre Canne, Italy, June)* (Vol. LNCS 6654, pp. 264-269). Heidelberg: Springer.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. New York: Cambridge University Press.

Leadbeater, C., & Miller, P. (2004). *The Pro-AM Revolution − How Enthusiasts are changing our Economy and Society*. London: Demos.

Lessig, L. (2008). *Remix: Making Art and Commerce thrive in the Hybrid Economy*. New York: Penguin Press.

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.

Ludwig, T., Dax, J., Pipek, V., & Wulf, V. (in print). A Practice-oriented Paradigm of End-User Development. In F. Paterno & V. Wulf (Eds.), *New Perspectives in End User Development*: Springer Int.

Maceli, M., & Atwood, M. E. (2011). From Human Crafters to Human Factors to Human Actors and Back Again: Bridging the Design Time – Use Time Divide. In M. F. Costabile, Y. Dittrich, G. Fischer & A. Piccinno (Eds.), *End-User Development* (Vol. 6654, pp. 76-91): Springer Berlin Heidelberg. doi: 10.1007/978-3-642-21530-8_8.

Mørch, A., Caruso, V., & Hartley, M. D. (in print). End-User Development and Learning in Second Life: The Evolving Artifacts Framework with Application. In F. Paterno & V. Wulf (Eds.), *New Perspectives in End User Development*: Springer Int.

Myers, B. A., Ko, A. J., & Burnett, M. M. (2006). Invited Research Overview: End-User Programming *Human Factors in Computing Systems, CHI'2006 (Montreal)* (pp. 75-80).

Nardi, B. A. (1993). *A Small Matter of Programming*. Cambridge, MA: The MIT Press.

National Research Council. (1999). *Being Fluent with Information Technology*. Washington, DC: National Academy Press.

Norman, D. A., & Draper, S. W. (1986). *User Centered System Design; New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

O'Reilly, T. (2005). What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

Preece, J., & Shneiderman, B. (2009). The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. *AIS Transactions on Human-Computer Interaction, 1*(1), 13-32

Raymond, E. S., & Young, B. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly & Associates.

Rittel, H. (1984). Second-Generation Design Methods. In N. Cross (Ed.), *Developments in Design Methodology* (pp. 317-327). New York: John Wiley & Sons.

Schuler, D., & Namioka, A. (1993). *Participatory Design: Principles and Practices*: Lawrence Erlbaum Associates, Inc.

Shneiderman, B. (2009). National Initiative for Social Participation. *Science, 323*(5920), 1426-1427

Star, S. L. (1989). The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving. In L. Gasser & M. N. Huhns (Eds.), *Distributed Artificial Intelligence* (Vol. II, pp. 37-54). San Mateo, CA: Morgan Kaufmann Publishers Inc.

Tapscott, D., & Williams, A. D. (2006). *Wikinomics: How Mass Collaboration Changes Everything*. New York, NY: Portofolio, Penguin Group.

Tetteroo, D. (2013). TagTrainer: a meta-design approach to interactive rehabilitation technology *End-User Development: Fourth International Symposium (Copenhagen, Denmark, June)*: 289-292.

Tetteroo, D. (2015). *End-user Adaptable Rehabilitation* (Ph.D. Dissertation), Eindhoven University of Technology.

Tetteroo, D., & Markopoulos, P. (in print). EUD survival 'in the wild': evaluation challenges for field deployments and how to address them. In F. Paterno & V. Wulf (Eds.), *New Perspectives in End User Development*: Springer Int.

Tetteroo, D., Seelen, H., Timmermans, A., & Markopoulos, P. (2014). Rehabilitation Therapists as Software Creators?: Introducing End-User Development in a Healthcare Setting. *International Journal of Sociotechnology and Knowledge Development (IJSKD), 6*(1), 36-50. doi: 10.4018/ijskd.2014010103

Thaler, R. H., & Sunstein, C. R. (2009). *Nudge – Improving Decisions about Health, Wealth, an Happiness*. London: Penguin Books.

von Hippel, E. (2005). *Democratizing Innovation*. Cambridge, MA: MIT Press.

Wenger, E. (1998). *Communities of Practice – Learning, Meaning, and Identity*. Cambridge, UK: Cambridge University Press.

Wenger, E., McDermott, R. A., & Snyder, W. (2002). *Cultivating Communities of Practice: A Guide to Managing Knowledge*. Boston, MA, USA: Harvard Business Press.

Ye, Y., & Fischer, G. (2007). Converging on a "Science of Design" through the Synthesis of Design Methodologies (CHI'2007 Workshop). from http://swiki.cs.colorado.edu:3232/CHI07Design/3

Zhu, L. (2012). *Hive-Mind Space: A Meta-design Approach for Cultivating and Supporting Collaborative Design*. (PhD), Università degli Studi di Milano, Milano.

Zhu, L., Barricelli, B. R., & Iacob, C. (2011a). A Meta-Design Model for Creative Distributed Collaborative Design. *International Journal of Distributed Systems and Technologies (IJDST), 2*(4), 1-16. doi: 10.4018/jdst.2011100101

Zhu, L., & Herrmann, T. (2013). Meta-design in Co-located Meetings. In Y. Dittrich, M. Burnett, A. Mørch & D. Redmiles (Eds.), *End-User Development: 4th International Symposium, IS-EUD 2013, Copenhagen, Denmark, June 10-13, 2013. Proceedings* (pp. 169-184). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-642-38706-7_13.

Zhu, L., Mussio, P., & Barricelli, B. R. (2010). *Hive-mind space model for creative, collaborative design*. Proc. of 1st DESIRE Network Conference on Creativity and Innovation in Design, Aarhus, Denmark, 121-130.

Zhu, L., Vaghi, I., & Barricelli, B. R. (2011b). *A Meta-Reflective Wiki for Collaborative Design*. Proc. of WikiSym, 53-62.