Abstract — User diversity and co-evolution of users and systems are two important phenomena usually observed in the design and use of IT artifacts. In recent years, End-User Development (EUD) has been proposed to take into account these phenomena, by providing mechanisms that support people, who are not software professionals, to modify, adapt, and even create IT artifacts according to their specific evolving needs. This is particularly true in the case of sentient multimedia systems, in which the system is called on to interact with multiple sensors and multiple human actors. However, to motivate and sustain these people, a culture of participation is necessary, as well as proper meta-design activities that may promote and maintain it. To this aim, this paper first proposes a model for describing interaction and co-evolution in sentient multimedia systems enhanced by EUD features. Then it presents four main roles involved in interaction and co-evolution, including that of maieuta-designer, as the “social counterpart” of the meta-designer. Finally, it describes how the maieuta-designer is in charge of carrying out all those activities that are necessary to cultivate a culture of participation, by means of proper ways that are briefly introduced in the paper.

Keywords — End-User Development; cultures of participation; co-evolution; meta-designer; maieuta-designer

1. INTRODUCTION

Sentient multimedia systems are distributed systems that actively interact with the environment through the exchange of multimedia information with many kinds of information sources, such as sensors, robots, actuators, websites and others. End users also belong to such sources of information, since they are called on to communicate and express their feelings, evolving needs and requests to this web of computational nodes. Accordingly, the overall system has to take into account this information flow coming from humans. If we take this stance, a sentient multimedia system can also be seen as a socio-technical system, which encompasses people (rather than just users) that are bound together by social ties and personal relations of acquaintance and that are also linked with each other and with personal devices and other machines. The latter are able to perceive the environment in which people interact, also by considering the capability of people to feel a situation besides perceiving it (e.g., through the issue of a preference, "likes", and emoticons in tweets), and interpret situations to give people multimedia and multichannel means to act accordingly.

In many IT domains, IT artifacts are usually developed as commodities rather than as ad-hoc projects, that is more for uniform populations of consumers rather than for members of different communities that exhibit local needs and perform situated practices to achieve their goals [1]. Moreover, each end-user community is often characterized by user diversity, due to users’ different physical and/or cognitive abilities, past experiences, roles, responsibilities and work contexts. To this end, in today's competitive global market, the adoption of product configuration software has recently helped to increasingly speed up the understanding of the customers' needs for a successful design and implementation of customized products [2-5]. In fact, product configuration is the activity of customizing a product, in order to better meet the needs of a particular end user more quickly. However, fulfilling the needs of end users is a “moving target” [6], since they evolve (e.g., regarding their proficiency of use, skills, expectations, needs, wishes and domain knowledge) by using software systems, and they can also change their practices (to accommodate the new artifact [7]). Acknowledging this twofold evolution (i.e., of users and their tasks) entails the requirement that IT artifacts should be designed to be very flexible, in order to be easily adapted to the specific needs of the user communities and, hopefully, to be personalized by the individual users to better fit their own evolving needs. This overall phenomenon has been called co-evolution of users and systems, to denote the variety of situations where users and their systems must co-evolve in a continuously self-adapting mutual fit [8].

End-User Development (EUD) has been proposed as one possible solution to cope with the challenges posed by user diversity and co-evolution, since it encompasses techniques that allow end users to modify and extend their own IT artifacts without necessarily delegating these modifications to software professional developers. Taking co-evolution seriously sheds light on the fact that continuously relying on professional actors for these interventions would not be feasible in the long run. Indeed, one kind of unintended consequence related to IT artifact deployment, which is reported most frequently in the specialist literature, regards the never-ending request for modifications, corrections and evolution of artifacts by the users [9]. In other words, in a EUD perspective, software systems are viewed as “continuously evolving sociotechnical
systems driven by design activities of both professional software engineers and users” [10].

For the particular human-oriented extension of the traditional definition of sentient multimedia systems (see above) proposed by us, we uphold that an EUD approach is necessary also for the design and continuous evolution of this novel class of applications. In particular, through the use of EUD methods and techniques, a sentient multimedia system can resemble a living system, with some degree of intelligence, that reacts to the end users’ evolution through the consequent self-adaptation and in turn favors the adaptation of end users to the evolution of the system. In this way, the word “sentient” would also imply “alive” and “intelligent”, as this kind of emergent behavior is what characterizes many complex systems, like socio-technical systems, that do have feedback loops between perception and action.

However, this far-reaching objective also requires the “cultivation” of a culture of participation, in order to motivate and sustain end users in their contribution to system evolution. This avoids the risk of participation inequality [11] and of replicating the current gap between IT professionals and end users at the shop floor level. The meta-design framework, which aims to help “users to become co-designers at use time”, has been proposed to this latter aim [12]. However, this framework seems to have neglected some important aspects that might make it more operative in real settings, like its relationship with activities promoting a culture of participation. Therefore, in this paper, we would like to investigate how to extend the original proposal of meta-design with mechanisms that are more specifically aimed at cultivating a culture of participation and thus enabling a suitable environment for the sustainable co-evolution of users and their systems.

To this end, we draw on our research experience in a variety of application domains (e.g., medicine, mechanical engineering, e-government and others) and on the analysis of the existing literature about a variety of EUD projects. We first propose a model for interaction and co-evolution that aims at clarifying the dynamics occurring in EUD settings among the different professionals involved, their tasks and the systems they use. In particular, we expand the technical activities a meta-designer should perform to support system adaptation and growth over time, and then provide indications about the activities of his/her “social counterpart”, namely the maieuta-designer, who is in charge of activating all the necessary social mechanisms that may sustain the co-evolution phenomenon.

The paper is organized as follows: Section II presents the related works; Section III describes the proposed model for interaction and co-evolution between users and systems, by clarifying in particular the role played by each different professional; Section IV expands the role of the maieuta-designer and proposes a framework for supporting his/her activities and Section V concludes the paper.

II. RELATED WORKS

Since the eighties, the human-computer interaction (HCI) literature has proposed different techniques for the design of interactive systems. They start from user-centered methods [13] – including field studies, interviews, task analysis, usability testing – and move on to participatory design techniques [14], where users are directly involved in the creation of interaction scenarios [15] and/or static and semi-static prototypes [16].

However, while HCI scholars have been considering user-centered and participatory design approaches as consolidated and successful practices for interactive system development, only in recent years the need for continuous system development with the participation of end users also at use time has received adequate attention. Consequently, end-user programming (EUP) techniques have been embedded in commercial software, such as macro recording in word processors, formula composition in spread sheets or filter definition in e-mail clients. EUP is defined in [17] as “programming to achieve the result of a program primarily for personal, rather than public use.” However, as highlighted in [18], the problem with end-user programming is that the programs created by end users are often of too low quality in terms of efficiency and maintainability. To cope with this problem, the End-User Software Engineering (EUSE) research area has emerged, which studies EUP practices and proposes new kinds of technologies that help end users improve software quality [17]. Software engineering activities, such as specification, reuse, testing, and debugging, are the primary focus of this research area, and therefore attention is put on the software code created by end users.

Recently, the term ‘end-user programming’ has been gradually replaced with the term ‘end-user development’ [19], in order to give user involvement in system design a broader perspective, with respect to mere code development for personal goals. Indeed, EUD denotes any kind of active participation of end users in the software design and development process, ranging from requirement specification through domain-specific modelling (cfr. [20]) to more advanced activities, such as system personalization and modification, or even creation of new software artifacts [21]. Therefore, contrary to participatory design, EUD research advocates end user participation not only during the design phase, but also during system usage. Both research lines are currently very active: on the one hand, new methods and techniques are being studied to better capture and satisfy user requirements; on the other, a variety of mechanisms are being proposed to allow run-time system modifications with increasing complexity and expression power [22] and possible creation of new software artifacts to be used by (or also by) other people [23].

This has led to the conception of a new design paradigm for systems that support EUD activities, namely meta-design. In [10], meta-design is regarded as a framework for creating “sociotechnical environments that empower domain experts to engage actively in the continuous development of systems”. Conversely, in [8] Costabile et al. view meta-design as “a design paradigm that includes end users as active members of the design team and provides all the stakeholders in the team with suitable languages and tools to foster their personal and common reasoning about the development of interactive software systems that support end users’ work”. The two proposals focus on different but complementary aspects of meta-design: the former stresses the social as well as the technical nature of the software environment that should foster
users’ involvement; the latter highlights the importance of defining proper languages and tools to adequately support different stakeholders’ participation in system development.

In both cases, meta-design is regarded in the wider perspective of cultures of participation [12, 24]. Indeed, cultures of participation open up new opportunities and challenges for the design of innovative interactive systems, whose users “are provided with the means to participate and to contribute actively in personally meaningful problems” [12]. With respect to EUSE, cultures of participation thus pay more attention to interaction design and conceptual modelling of EUD environments, cooperation among users and impact on work organizations. Particularly, beyond meta-design, Fischer proposes two other components for establishing a culture of participation [12] that we think are fundamental also in the conception and design of sentient multimedia systems: 1) social creativity, in order to allow all possible contributions to be considered to solve a complex problem and to support people interacting with each other and through shared hardware and software artefacts; 2) richer ecologies of participation, in order to obtain different levels of participation on the basis of the different roles that people and their devices can play or would like to play.

The importance of cultures of participation has been demonstrated in the literature with reference to a variety of application domains [12]. However, one aspect has been neglected till now, that is how to enable users to appropriate such a culture, in order to obtain software artifacts that successfully evolve in user’s hands and co-evolve with users’ tasks, abilities, skills, and preferences. This is particularly true for sentient multimedia systems, as they are bound to the environment and the people living in it, and thus this requires users to be even more engaged in their shaping over time. In our view, this requires proper mechanisms and tools that allow the nurturing and cultivation of cultures of participation within organizational settings. To this end, in the following we propose an analysis of the activities a meta-designer should perform throughout the software life cycle. In particular, we suggest refining the meta-designer role, by splitting and specializing it into two main sub-roles – a more technical role and a more social one – which, according to the situation, domain, and budget constraints, can be played by the same person or by different professionals.

Figure 1 also highlights the two roles involved in this traditional view on interaction and co-evolution: the end user and the software developer. The end user is considered a passive user of an interactive system and consumer of its products and services. The software developer is the creator of the system and, during its life cycle, he/she may be called on to modify and extend it for adaptation to the emerging needs and requests of end users.

**B. Four Main Roles in EUD practice**

In EUD literature, a new role is considered: the meta-designer. Thus, the two main roles usually highlighted, other than the software developer, are: end user and meta-designer. End users are increasingly required to act as active contributors at use time, thus becoming “producers” of contents and functionalities, like in Wikipedia, Scratch, SketchUp, and many others [12]. In literature such an “active” end user is called in different ways: “power user” [30], “local developer” [31], “gardener” [32], “end-user developer” [33], “bricoleur” [34].
To disentangle this variability of names, we have proposed to refer to such a figure with the term *domain developer* [23]. This term has been chosen because this person is always an expert of the domain in which he/she works and her/his main goal is more the development of the capabilities available in her/his setting than just software code (software is never an end in itself, but always a means). Thus, the domain developer subsumes all those roles denoting people in charge of carrying out software development activities (namely, ‘actual’ EUD activities) without being professional software developers. In some cases, end user and domain developer are roles played by the same person, as in the case described in [10], where a geoscientist decided to spend three months in acquiring programming knowledge, in order to be able to develop software for himself to analyze the data he collected. However, in the majority of situations they are played by different people, who may also belong to different communities, like in multi-tiered proxy design problems [33], i.e. in all those situations where end users cannot or are not willing to act as developers. A typical example is the case of e-government, where citizens using e-government services constitute the community of end users, whilst civil servants called on to develop services for citizens belong to the community of experts in government issues [35], and thus may become domain developers. Another example is the system described in [36], which provides an editing tool that allows caregivers (domain developers) to customize a simple, wireless prompting system for individuals with cognitive disabilities (end users).

The role of meta-designer, on the other hand, is intended for all professionals who are in charge of creating “socio-technical environments that empower users to engage actively in the continuous development of systems rather than being restricted to the use of existing systems” [37]. In other words, a meta-designer “creates 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” [12].

Given these definitions, which do not completely clarify the activities a meta-designer should actually perform, we also consider the role of *maieuta-designer* (the term is pronounced just like that of meta-designer, but with a *ju* in the middle: *meta* designer vs. *maieuta* designer). Like the meta-designer, also the maieuta-designer can be considered as someone in charge of designing the EUD-enabling environment, by which domain developers can build and adapt the artifacts to be used by end users. The role of the maieuta-designer encompasses activities that are involved in the task of supporting the meta-task of the domain developers, namely creating the socio-technical preconditions for: a) having the domain experts appropriate the design culture and technical notions necessary for the meta-task of artifact development and b) involving as many end users as possible in the process of continuous
refinement of the artifact, by improving participation and “produsage” [38]. For this reason we call such a designer a “maieuta”. This is partly in analogy with the Socratic method of making people acquire notions, motivations and self-confidence to undertake challenging tasks and partly in clear assonance with the term meta-designer, of which it is a specialization more oriented to the social aspects of EUD practice than to the technical ones [34].

C. Co-Evolution of End Users and Systems through EUD

The four roles described above – end user, domain developer, meta-designer and maieuta-designer – interact with each other and with the IT artifact and EUD tools, and each contributes to the co-evolution phenomenon. Fig. 2 presents an extended version of the Interaction and Co-evolution (ICE) model previously described, which encompasses all the four roles.

In EUD practice, the traditional co-evolution process (left-hand side of the figure) is sustained by the right-hand side co-evolution process, which involves the domain developer (see Fig. 2). Indeed, requests for system evolution coming from end users reach domain developers, who may directly operate on the system through EUD tools or, if necessary, may in turn ask meta-designers for the evolution of their own tools (by means of the most internal cycle, i.e., the interaction cycle). In particular, the task-artifact cycle that involves end users affects the meta-task-artifact cycle of domain developers. For example, in the Electronic Patient Record (EPR) project described in [39], ward physicians can be aware that new data are needed (for example, in the case of specific pathologies), but they cannot find the related module in the EPR. Thus, they have to ask for a new specific module from the head physician, who, using EUD tools, will evolve the current EPR accordingly or request meta-designers to create the new type of module.

The most external cycle is also in this case a long-term cycle that regards the mutual influence between the technology used for artifact implementation and the organization of the community of the domain developers (technology-domain developer organization cycle). Technology advances give meta-designers new possibilities for improving interactive systems used by domain developers to evolve the system, resulting in new interaction possibilities that might also change users’ work habits.

Thanks to the powerfulness of the current technology, even accessible through the Internet, users have the possibility of increasingly taking an active role in the development of software tools suited to their needs. This has led from a strong dichotomy between the end user and the domain developer to a continuum of roles that constitutes a rich ecology of participants [12, 40] with different skills towards development, responsibility, appropriation and contribution in the whole ecosystem. This is true also in the case in which the same person is and wants to be just a “consumer” (i.e., only a user) in some situations and a “producer” (i.e., domain developer) in others. Therefore, “end user/domain developer” is not an attribute of a person, but a role assumed in a specific context. As we will discuss in the following, the maieuta-designer is in charge of fostering and favoring the migration path from the role of end user to that of domain developer.

The artifact in Fig. 2 is an intermediary object between the two co-evolution processes, which can be regarded as being composed of two parts: 1) the software system devoted to the end user and 2) the EUD tools (including the EUD environment and/or software components as building blocks of the software system being developed) that are used by the domain developer to generate and/or adapt the software system for end users.

An intermediary object is usually an object that is shared, exchanged and circulated among members of networks and communities to mediate their interactions [41]. In [23], we have further distinguished intermediary objects as boundary objects and knowledge artifacts to better characterize their role in EUD contexts. The notion of boundary objects has been introduced by Bowker and Star [42] to account for those artifacts that enable a sort of standardized and effectively simplified communication and coordination between members of different communities of practice. For example, in the e-government project described in [43], a civil servant may generate the XML description of an e-government service through a suitable EUD environment, and this description gives rise to the automatic creation of the web pages to be used by citizens who will apply for the service. Therefore, the XML description and the corresponding web pages can be regarded as a boundary object between the civil servant and the citizen communities. Knowledge artifacts, on the other hand, are artifacts that enable and support learning and innovation within a specific community of practice (what in [44] has been called a “knowing community”), namely processes of knowledge acquisition, accumulation and sharing among its members [45]. For example, the Electronic Patient Record [39, 46, 47] represents a knowledge artifact used within a hospital ward and among different wards for accumulating and sharing knowledge about patients.

D. Supporting Co-Evolution in EUD settings: the Meta- and Maieuta-Designer roles

Fisher et al. have proposed a set of meta-design guidelines [10], namely indications at a high level of abstraction, on how to carry out a meta-design project. Assuming the validity of such guidelines, we propose here to make a further step, by identifying some more operational indications, in order to make the meta-design activities concrete. In particular, splitting the social activities and technical activities that the meta-designer and maieuta-designer are called on to carry out respectively is done to take into account the socio-technical gap, that is the divide between what is known that should be supported socially and what can actually be technically supported [48].

To this aim, we suggest that the meta-designer be in charge of designing and providing the most effective EUD tools that may sustain the co-evolution between end users, domain developers and IT artifacts. In this way, the meta-designer is not just a software developer playing the role described in Figure 1 (system development and its possible evolution over time, on the basis of the users’ requests); but he/she must be endowed with a set of skills that allows him/her to understand the users’ potential for participation in software co-creation, in order to support them with suitable technical tools. Thus he/she must have competencies in human-computer interaction,
computer-supported cooperative work, interaction design, knowledge management, multimedia and even semiotics [49].

A variety of EUD solutions have been proposed in the literature over the years, from script creation [36], to component-based approaches [50, 51] and from meta-model instantiation [52, 53] to visual programming [8, 54, 55]. The meta-designer is thus called on to choose the most suitable paradigm for the case in hand, and tail it to the application domain, namely to its habits and users’ characteristics and preferences. Furthermore, a meta-designer is requested to develop the infrastructures for communication among end users and between end users and domain developers [56].

The maieuta-designer, on the other hand, is not only what elsewhere has been defined as a facilitator [57], that is, the role responsible for facilitating the adoption of an IT artifact within a certain community. Rather, the maieuta-designer is the person who is supposed to facilitate appropriation [58], i.e., the process by which end users migrate from their initial role to that of domain developer along the continuum of roles discussed in [40] or, at least, to enable and empower end users to appropriate and contribute to their IT artifacts. Therefore, the end users can commit themselves to improving the artifacts as a way to make them more effective and their work more efficient. Whenever an end user is not capable of, or not interested in, “evolving” into the role of domain developer, the maieuta-designer might favor her/his participation in system evolution, e.g., by simply guiding her/him to report perceived shortcomings and system faults, and suggesting due modifications and appreciated improvements. Moreover, the maieuta designer is also in charge of reducing the socio-technical gap, by creating the conditions and making feasible the use of the IT artifacts and lowering the tendency of the users to just give up using them, especially if they do not have any technical support from IT professional people.

The clarification provided here of the roles of meta-designer and maieuta-designer is based on a critical reflection on the authors’ experience in a variety of EUD projects. Actually, in these projects they have often been engaged in playing both roles, at different stages of the system life cycle. As meta-designers, they have studied and proposed different interaction metaphors and environments, to support domain developers in carrying out EUD activities; whereas, as maieuta-designers they have realized that, besides asking users the right questions as any good business analyst does, it is even more important that the users themselves are induced to think about their answers and build their own awareness on how to deal with issues through the system. Thus, the aim of this paper is to shed light on the techniques that could be adopted to sustain system appropriation, by underlying the need for transferring this capability to someone inside the organization who could reiterate the awareness process and make it sustainable.

IV. HELPING END USERS HELP THEMSELVES

This section provides some further hints on how the tasks of the maieuta-designer could be performed.

A. Identifying the maieuta-designers and their tasks

As has been outlined above, the concept of maieuta-designer means identifying someone who could make the community gathering around a EUD platform progressively more independent of the IT professionals. To some extent, he/she is the person who guarantees the long-term sustainability of the EUD project. Therefore, this can be an IT professional with an educational curriculum that is quite different from the one traditionally proposed for the common software analyst or engineer. This agenda would encompass, for instance, teaching the basics of social informatics and some qualitative research methods adapted to the IT domain [59], like focus groups [60], insights on current theories on IT impact and risk management [61], as well as notions of socially-informed history of technological evolution [62].

The person playing this role must also train on the job one or more “insiders” of the community of end users that will continue his/her work of facilitation. The latter should be endorsed by the sponsors of the IT project and the organization’s top managers, and also be chosen on a voluntarily basis according to their ability and will to encourage colleagues to take part in the development process. The term “designer” is not out-of-place here for at least two reasons: first, the maieuta-designer would be a clear example of a critical designer, i.e., someone “who asks carefully crafted questions and makes [people] think” [63], instead of focusing on solving problems and finding answers. Moreover, one of the main tasks of the maieuta designer would be to “design” (or better “co-design”) initiatives in which to promote the EUD project, disseminate the underlying values and concepts (i.e., empowerment, co-production, appropriation, co-evolution, produsage, equipotentiality [38], etc.) and enrol the most expert and enthusiastic end users. Then the maieuta-designer should give due visibility of the end users’ contributions, and devise simple mechanisms to foster participation and build a real culture of participation. This can be done in many ways: for instance by applying blended gamification, within a competition among colleagues, possibly associated with some reward or compensation policy, e.g., a mechanism by which “the more contributions produced, the higher the rank achieved”. Moreover, this can be done by setting up a social media associated with the IT project, e.g., a forum, a blog, a wiki, or something that integrates all of these simpler components, in which to ask for content and contributions and moderate communication within these ad-hoc means. In so doing, such a Web resource would flank the EUD platform as an additional “resource for action” [64] and a virtual meeting place where tasks are coordinated upon the EUD artifacts and the related procedures, FAQs and use instructions are documented/discussed.

B. The Google example

Over the years, Google software developers have produced a set of apps and gadgets, technologies for voice recognition, home appliances and entertaining devices. Many of them are smartphone-based and thus accessible and widely used by end users, as well as most of the technologies developed so far by Google. However, even though the specific functionality of each application is usually known, the possibility of connecting or using together different applications, as parts of a unique ubiquitous system, is rarely considered by end users. Indeed, Google apps and technologies are thought to also interact with each other, in order to provide a more pervasive and ubiquitous
experience. For example, Chromecast dongle can be connected to the TV and exchange data with your tablet or phone, while Google Now can exchange data with Google platform, in order to suggest personalized hints. Actually, all these applications can be considered as parts of a pervasive multimedia system surrounding us.

Since people might not be immediately aware of this possibility, Google has set up a team whose aim is to advertise the “hidden” sentient multimedia system already available in our pocket. The team has thus created the Google House Project, an itinerant house toured through New York, London, Paris, Hamburg, Dubai and many other important cities all around the world. The house is split into five rooms: kitchen, living-room, study, travel-room and fashion-room, with a sixth environment outside to try out Google Glass, the company’s latest wearable device (Fig. 3). The majority of technologies available in the house are smartphone-based, linked up to huge displays scattered around the home via Wi-Fi.

In a such setting, tablets or phones can be used in the kitchen to look for recipes, by using voice commands, or run a Google-owned YouTube channel with millions of subscribers while cooking, or have the system convert doses from grams to ounces. In the living-room you can use Google Nav to plan your day’s journey or use Google Chromecast dongle. The Chromecast dongle is popped into the back of the television and then connects wirelessly with any tablet or smartphone to stream your favorite movie or, even, use the new photo app Auto Awesome. This application is able to automatically exclude duplicates and duds from a gallery of holiday pictures and take the smiles from a series of photos of the same people, in order to pool them into one, perfect shot. In the fashion room, your daughter can use Hangout to talk with her friends about choices of dress or to access Google Trends to identify the most popular fashion styles while getting dressed.

In short, Google has created over the years a variety of applications and devices, which have been put in the hands of the users, in order to allow them to provide their feedback, try different usages, contribute their contents (e.g., in Google Maps or Google Sketchup), and eventually become part of the world-wide Google community. Then, Google developers (we can call them meta-designers in this case), have offered the possibility of combining these “pieces” of technology in different ways and at the user’s own pace to obtain a low cost, sentient multimedia system. However, most end users, even though they are perfectly able to exploit existing applications and devices in this new way, without the intervention of computer experts, were not aware of this possibility. Hence, Google had the idea of “advertising” it through a widely understandable example: a real and tangible house. Indeed, the team of researchers welcoming and hosting visitors to try out the Google experience (the maieuta-designers in this case), aimed to make users conscious of the possibility to create their own sentient multimedia system, and thus actually carry out a form of EUD. Furthermore, the Google House project allows end users to appreciate the different usages of each application: for example, the use of Hangout is shown in the fashion-room, but nothing prevents a scientist from deciding to use it in his study, to keep in touch with his foreign colleagues.

In other words, the task of maieuta-designers has been accomplished in this case by creating a real house and exploiting the resonance that the Google name may have all over the world. Like a bait-and-switch, this has enabled a limited number of users to actually visit the house and try its gadgets, but it has allowed millions of users to become aware of the nearly infinite scope of sentient multimedia systems and of the ever-expanding catalog of available applications which can be combined, according to one’s preferences.

Obviously, this approach can rarely be followed in medium and small organizations, and thus other ways should be identified to make users more confident in adopting tools and their capabilities, in order to put them in a position to exploit customization and extension possibilities, for a better fit with their needs. Furthermore, a true maieutic approach would also entail challenging real users, visiting the house, with Cedric Price’s provocative question: “if Google [Technology] is the answer, what is your question?” In looking for possible answers, the end users could also appropriate the underlying idea of such an augmented shelter and could make the best out of it in their situated lives and homes. For these reasons, the following sub-section delineates a more scalable approach that, capitalizing on the experience gained so far in different IT projects, is built around the idea of asking users questions.

---

1 http://www.thenational.ae/uae/technology/google-house-in-dubai-the-home-of-the-future-now
### TABLE I.

<table>
<thead>
<tr>
<th>Macro-areas of concern</th>
<th>Sample Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Psychological ownership and change management</td>
<td>What’s the system for you, and why has it been produced and its adoption encouraged within the organization? Do you think that communication within your team, or with the other teams, has changed lately, and if this is the case, has it been for the better or for the worse? How long have you been using the new system, do you think your workload has changed, that is increased, reduced, or is it just the same?</td>
</tr>
<tr>
<td>Work process redesign and adjustment (i.e., fit to task)</td>
<td>To what extent do you think you can exploit the system’s full potential? How much do you think the system fits your specific needs currently? Do you still use paper and office applications that you believe the new system will (or should) substitute sooner or later?</td>
</tr>
<tr>
<td>Usability shortcomings and room for improvement</td>
<td>Have you found using the system easy so far? Have you realized you have made errors in the process of either entering or retrieving information from the system?</td>
</tr>
<tr>
<td>Lack, redundancy or overload of data structures and functionalities</td>
<td>Do you think the system is requiring you to fill in too many data that are not really necessary to proceed in your tasks? Do you think the system provides too many functionalities among which you need to find the right one for your tasks?</td>
</tr>
<tr>
<td>Anomaly detection, bug reporting and evolution traceability</td>
<td>Have you found any errors or something you’ve considered a fault of the system while using it lately? Have you applied some effective solution or workaround to overcome a shortcoming related to the system lately? Do you think that the system has become more difficult to be used after some of its recent updates and new releases?</td>
</tr>
<tr>
<td>End-user deskilling and expertise preservation and enhancement</td>
<td>Have you lately experienced problems in the handing over of tasks or in the workflow (like unusual delays, common resources blocked by other teams and the like)? Do you think that using the new system may contribute to preserving or even enhancing your know-how about the work tasks?</td>
</tr>
<tr>
<td>Authorship- and privacy-related concerns</td>
<td>Do you think that sharing content and system modifications within your team prevents you from protecting ideas and information? Do you think that either the data you put into the system, or the actions you perform during its use could threaten your security and privacy?</td>
</tr>
<tr>
<td>End-user accountability and power issues (also empowerment)</td>
<td>Since the introduction of the new system, do you think that new people or roles have gained more visibility and power within your organization, at the expense of others? What’s the main obstacle that prevents you from participating more actively in the IT project (like time, skills, the colleagues already involved, a sense of pointlessness, ...)? What could really convince you to join the IT project, if anything (e.g., explicit acknowledgment by the top management, economic rewards, non-monetary compensations, benefits, social status, ...)?</td>
</tr>
</tbody>
</table>

C. The maieutic approach

A maieutic approach is mainly characterized by the fact that it “brings others to conceive ‘thoughts or ideas’” with questioning [65], that is by helping others actively understand by themselves how they could make a worthy contribution to the project. To contextualize this approach in EUD settings, we propose a tentative list of items (see Table 1). Each item in the second column of Table 1 is a question that the maieuta-designer could ask (or speak about with) his/her colleagues to address a broader theme or topic regarding the process of digitization of the work setting or the related changes. This can be done in either small polls or surveys, administered through social media, or in informal but scheduled meetings with the members of a specific team at a time. It could also be done even in totally informal and impromptu talks at the coffee break or in similar situations [66]. In particular, the first question in the table (“What’s the system for you, and why has it been produced and its adoption encouraged within the organization?” [67]) is the most important one, as it refers both to the original Socratic main question (What’s this? Ti estì;) and to one of the most important matters of concern in requirement engineering.

Far from being comprehensive, the list of questions reported in Table 1 represents just a first contribution within a research strand. This could address more seriously how to contribute to fostering a culture of participation within organizational communities, especially in the context of a digitization project that is supported by EUD and meta-design techniques. Indeed, a more general framework of concerns, that a maieuta-designer should address, can be built from this preliminary list and the study of its impact on real settings (see the first column of Table 1). This framework to date includes eight macro-areas of concern that encompass: psychological ownership and change management [68, 69]; work process redesign and adjustment (i.e., fit to task); usability shortcomings and room for improvement; lack, redundancy or overload of data structures and functionalities; anomaly detection, bug reporting and evolution traceability; end-user deskilling and expertise preservation and enhancement; authorship- and privacy-related concerns; end-user accountability and power issues (also empowerment).

Our point is that there is a need to detect motivated people within organizations, and not only give to the domain developers a set of tools (i.e., the EUD environments), but also (and above all) assign to some people (i.e., the maieuta-designers) precise responsibilities and roles. It is particularly important to provide maieuta-designers with an indication of a set of possible actions to undertake and initiatives to foster, so that they can contribute to building a real culture of participation within their organization. All the actors involved may thus enjoy such a culture within the wider process of co-evolution.

V. Conclusions

In this paper we have proposed explicitly extending the notion of a sentient multimedia system to encompass socio-technical networks of humans and non-humans [70], where both kinds of agents are able to perceive and act in the environment, but only the former can feel it and make the
whole ensemble “sentient”. The most important feature of these networks is their ability to autonomously reconfigure their inner relationships (human-human, machine-machine, human-machine) and also to evolve, in the face of the continuous changes in the environment and the network itself. In this paper, we have focused on a particular design approach that does not only allow for continuous evolution, but even fosters it: meta-design. We have extended this framework, originally proposed within the EUD field, by enriching it with an additional role, the maieuta-designer, that could also be seen as a critical design device, and therefore as a way to make sentient multimedia systems able to wonder about and reflect upon themselves. As a critical design device, we suggest supporting the maieuta-designer with a list of questions, organized in turn according to a set of macro-areas of concerns, through which he/she can trigger reflection and cultivate a culture of participation. This can be a powerful way to make sentient multimedia systems more capable of interacting with an ever-changing environment and to be ready for the unexpected.

ACKNOWLEDGMENT

This work is partially supported by the Italian Ministry of University and Research (MIUR) under grant PON 02_00563_3470997 project VINCENTE - A Virtual collective INtelligence ENvironment to develop sustainable Technology Entrepreneurship ecosystems" and by the Italian Ministry of Economic Development (MISE) under grant PON Industria 2015 MI01_00294 “LOGIN – LOGistica INtegrata”. The authors wish to thank also Ms Lynn Magaret Rudd for her careful proofreading.

REFERENCES
