Data & Knowledge Engineering 143 (2023) 102108

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

GIRE: Gender-Inclusive Requirements Engineering

Inês Nunes *, Ana Moreira, João Araujo

NOVA LINCS, NOVA University of Lisbon, Portugal

ARTICLE INFO

Keywords:
Gender issues
Gender-inclusive
Conceptual modeling
Requirements engineering
Software development

ABSTRACT

Gender inclusion is fundamental to a prosperous society, but inequality and exclusion persist in various sectors of it. One of them is the ICT field, which is still struggling to represent the diversity of those it serves. The lack of diversity and power imbalance in software development affects the produced systems, that, instead of advancing gender inclusion, create new barriers in achieving it. Although considered neutral, software does not equally serve everyone who depends on it, favoring characteristics that are statistically more observed in those that are represented during development. As software development teams are predominantly male, it is not surprising that existing systems favor characteristics that are statistically more observed in men over characteristics observed in other genders. As technologies rapidly evolve and revolutionize the way we live, addressing this problem becomes urgent to ensure that these systems benefit everyone, regardless of their gender. As a first step towards this goal, we performed a systematic mapping study on gender issues in software engineering whose results indicated that gender impacts development and systems, but there are limited approaches for addressing it in Requirements Engineering. This study served as the foundation for proposing a conceptual model for gender-inclusive requirements. Its main objective is to facilitate discussion and analysis of gender and related concepts in the elicitation process to include them in the specification of requirements. In this paper, we extend this work by illustrating the concepts with an example, by presenting a process for using the knowledge of the model and a prototype tool that implements it, and by discussing an evaluation with 31 participants of the conceptual model's usefulness, difficulty of understanding, strengths and weaknesses, use and recommendation, and finally, its components. The results were positive as both novices and experts in conceptual modeling considered the model useful, provided comprehensive feedback on its strengths but also suggestions for improvement, and most answered positively to the questions about whether they would use and recommend it.

1. Introduction

Gender issues refer to "all aspects and concerns related to women's and men's lives and situation in society, to the way they interrelate, their differences in access to and use of resources, their activities, and how they react to changes, interventions and policies" [1].[1] Over the last decades, research and assessment have made it possible to understand where and when gender issues occur specifically, enabling the development of measures that contribute to significant progress for equal rights, responsibilities, and opportunities [3]. Yet, intersecting forms of discrimination are still being perpetuated, directly and indirectly, by social norms, practices, and gender-based stereotypes [4]. Such is the case of the ICT field, which is struggling to represent the diversity of those

* Corresponding author.
E-mail addresses: ir.nunes@campus.fct.unl.pt (I. Nunes), amm@fct.unl.pt (A. Moreira), joao.araujo@fct.unl.pt (J. Araujo).

[1] In this paper, we use "women" to refer to cisgender and transgender women, and "men" to refer to cisgender and transgender men, as recommended in [2].

https://doi.org/10.1016/j.datak.2022.102108
Received 16 January 2022; Received in revised form 18 October 2022; Accepted 13 November 2022
Available online 17 November 2022

it serves. Concerning gender, the field is still male-dominated: women comprise only 3% of ICT graduates worldwide [5], hold just 24% of all digital sector jobs [6], and represent only between 2% to 4% of open source developers [7]. Moreover, men are 15% more likely than women to be senior software developers, and two and four times more likely to be managers and executives, respectively [6]. Unfortunately, gender-based prejudice and hostility persists for minorities when joining the field [8], who still encounter inequalities in the workplace that reinforce the existent gender bias that excludes them [6].

The lack of diversity and gender imbalance in positions of power in the software industry suggest that software products might not be inclusive. Therefore, our aim is to study how gender issues have been addressed in software engineering. The first step towards fulfilling this goal was to conduct a systematic mapping study on gender issues in software engineering. The results confirmed the hypothesis that existing systems favor the characteristics statistically observed in those involved in development, ignoring and even discriminating against users whose characteristics are underrepresented in the process [9]. Indeed, gender plays a significant role in the users' attitudes towards software, and software systems provide better support for the characteristics statistically more observed in men over characteristics observed in other genders [8,10,11]. These gender issues in software systems have negative and harmful consequences for those who interact with them. However, software is typically assumed to be gender-neutral [12], and thus, employing design techniques such as the I-methodology, which models users' behaviors based on that of development teams, results in software systems that match the preferences, technical capabilities, and learning style of men [13,14].

For instance, problem-solving software is designed with the expectation that users will learn and engage with new features through tinkering. Yet, female users are statistically less likely to do so compared to male users [9]. In open source communities, women feel discouraged to participate due to the competition, negativity, and sexism in the environment [15]. If the users' gender is not visible, the acceptance rate of pull requests from women is higher than those from men, but if gender is visible, the acceptance rate is lower than those from men [16]. These gender issues have several negative effects on women's participation in these communities, such as anticipating less success, less interest, and showing lower levels of confidence in their abilities [15], which are currently not being addressed. Nevertheless, when gender is explicitly addressed, misconceptions and assumptions about underrepresented groups as homogeneous can inadvertently integrate harmful stereotypes into the software. In particular, the 'shrink it and pink it' practice [17] creates pink, simplified technologies, such as fashion and wedding video games [18], that impose and reinforce traditional notions of femininity [13]. Moreover, stereotypical masculine visual elements in interfaces, which are typically considered to be neutral, not only negatively impact female users' sense of belonging, but also the majority of potential users, including men, who are also more positively affected by the gender-inclusive design [19]. Therefore, our goal is to address gender issues in the requirements engineering process to ensure gender-inclusive system design decisions that will lead to software that supports all users equally, regardless of gender.

From the systematic mapping study we gathered a comprehensive overview of the state of the art on gender issues in software engineering. Overall, gender issues arise early in development, but there are few approaches for addressing them, and most are focused on the design phase. In addition, the impact of gender on systems and users is well documented, but this knowledge is scattered across distinct resources. A common structured vocabulary is necessary for analyzing and addressing gender and related concepts in development, particularly in requirements. This gap motivated the collection of key concepts from the mapping study for developing a common definition for gender-inclusive requirements. We used the collected knowledge as the groundwork for proposing a conceptual model for gender-inclusive requirements. The model is composed of four main components, *Gender, Sociocultural Context, Human Actor,* and *Software System*. Its goal is to support requirements engineers in eliciting gender-inclusive requirements by providing a taxonomy of gender concepts that impact software, and thus, facilitating the discussion of such concepts to include them in the specification of requirements. As a first step towards this goal, we created a document for each concept with a description of associated potential gender issues, recommendations to overcome them, and a question-based checklist to support their application. Then, we developed a three-step process to use the conceptual model through the documents in the elicitation activity. It consists of selecting the intended concepts to analyze, selecting the questions, and finally, gathering them in one document for semi-structured interviews, questionnaires, or creativity techniques. Lastly, as a proof of concept, we developed a prototype tool that implements this process. The prototype tool itself was developed using the conceptual model to be gender-inclusive. The conceptual model is then evaluated by thirty one participants regarding its usefulness, difficulty of understanding, strengths and weaknesses, use and recommendation, and finally, the completeness of each component. Although the majority regarded the model as moderately difficult to understand, they considered it very useful and its components complete. Also, most participants would use and recommend it. Overall, the results were positive and encouraging as participants provided valuable insights and constructive feedback for future improvement.

This paper is based on [20] and extends our work in [21] where we presented a systematic mapping study on gender issues in software engineering, whose results showed the limited existence of approaches for addressing gender in Requirements Engineering. To bridge this gap, we proposed a conceptual model for gender-inclusive requirements grounded on the findings of the mapping study. In this paper, we further examine the conceptual model concepts and relationships, use an example based on the car-sharing system to illustrate the concepts, discuss a process for using the proposed conceptual model and a prototype tool that implements it as a proof of concept, and present an evaluation of the conceptual model and its four main components with 31 participants. This evaluation focused on the usefulness, understandability, completeness, strengths and weaknesses of the model, as well as the participants' recommendation for, use. The results were promising and the received feedback from the open questions provided valuable suggestions for further improvement.

The remaining of the paper is structured as follows. Section 2 describes the systematic mapping study on gender issues in software engineering. Section 3 presents the proposed conceptual model for gender-inclusive requirements based on the study and describes its main concepts and relations using an illustrative example, the car sharing system. Then it presents the process for integrating the conceptual model in elicitation activities and discusses, as a proof of concept, the prototype tool that implements this process. Section 4 presents the evaluation of the proposed conceptual model and discusses the results and threats to validity. Finally, Section 5 concludes this paper and points directions for future work.

2. Gender issues in software engineering

A systematic mapping study [22] was performed to understand how gender issues have been addressed in software engineering. The research method consists of three main phases: planning, conducting, and reporting.

2.1. Planning and conducting

Planning phase. Planning involves the formulation of the research question, the definition of the search and study selection strategies, the establishment of the quality assessment criteria, and lastly, the specification of the data collection and extraction strategy. The research question is:

- **RQ.** *How have gender issues been addressed in software engineering?*

For a more focused and detailed search, this research question was decomposed into four sub questions:

- **RQ1.** How are gender and gender issues understood in software engineering?
- **RQ2.** What are the software application domains where gender issues were addressed?
- **RQ3.** What gender issues impact software users?
- **RQ4.** What approaches have been proposed to address gender issues in software development?

Conducting phase. The conducting phase follows the research protocol defined in the previous phase and it consists of applying the search process, selecting the relevant studies according to the inclusion and exclusion criteria and the quality assessment, and performing the data extraction and synthesis.

The **search strategy** aims to find the most complete and consistent set of studies according to the research purpose [23]. The search string was built with the key terms of the research questions:

("gender issues" OR "gender diversity" OR "gender bias" OR "gender stereotype" OR "gender inclusive" OR "gender equality" OR "gender gap" OR "gender difference") AND ("software engineering" OR "requirements engineering")

The search string was run automatically on July 11, 2020, on Google Scholar, yielding 729 candidate studies that were saved in a bibliography tool. Snowballing was used on the final set of selected papers to find applicable studies.

The **inclusion and exclusion criteria** were defined during the planning phase to avoid bias when evaluating and selecting the relevant studies that answer the research questions. The evaluation was performed in two iterations: (i) analyzing and reading the title and the abstract of the 729 studies, and excluding the studies that met at least one of the exclusion criteria, which decreased the number of studies to 60; (ii) reading the 60 studies in full, resulting in 31 selected studies. The **quality criteria** focused on the number of citations considering the year of publication and the CORE Rank. Snowballing was applied to the 31 studies and 5 more were selected for data extraction, totaling 36 studies.

A **data extraction** template records the content of each selected study to ensure that the selected papers are subjected to the same extraction criteria. The template was designed as a form and it consisted of a section for the meta-data of the study and a data extraction section where the information required to answer the research questions was synthesized. The data included for the gender conceptualization (RQ1), the application domain (RQ2), a description of the gender issues identified (RQ3), and the respective proposed approach to address them (RQ4).

2.2. Reporting the results

RQ1. How are gender and gender issues understood in software engineering? To answer this question we extracted information concerning the gender definition from the selected studies. We devised four categories of gender conceptualizations and classified the studies according to them: *Binary (categorical view of gender), Binary and Social Context, Social Construct*, and *Intersectional*.

The studies of the first category assumed a binary view of gender [8,10,24–35], focusing on individuals who either identify themselves as female or male, and assuming one's biological sex corresponds to one's gender. Most studies did not discuss their model of gender and used the concept as a statistical variable to analyze differences between women and men in the use, preferences, adoption, or interaction with software. The studies from the second category understood gender as a self-described attribute influenced by social roles [11,12,15,18,19,36–43]. However, they addressed gender as binary. In the third category, the studies conceptualize gender as a social construct and discuss the limitations of assuming gender binary approaches [9,17,44–48]. As a social construct, gender is a spectrum in which individuals align themselves by expressing their gender identity, independently of biological sex. Thus, differences in attitudes towards software are related to differences in gender expressions that result from internalized gendered behavior from a given social context. Yet, these differences in needs and expectations are not being considered and software is statistically privileging male users, placing female individuals at a disadvantage when using them [12,24,27,28,32,39,41]. Lastly, the fourth category presents "intersectionality" where approaches seek to explain how gender intersects with other social identities (e.g., race, culture), and how these intersections form multiple, layered identities [13,49]. Hence, social and cultural context are fundamental gender factors to consider when developing software. In summary, the following three main gender issues were identified.

Software is gender-neutral. Predominantly male technology development teams assume software technologies as exclusively technical and gender-neutral. Mostly, they assume their own needs and values are universally applicable. Also, there is a lack of knowledge from those involved in software development about the concept of gender and its impact, predominating the idea that these are unrelated, or that gender does not impact software.

Binary conceptualization of gender. If gender is addressed, it is typically simplified to a statistical binary category, sometimes associated with biological sex. This highlights the existence of gender differences. However, it does not fit them in a social context to understand why they exist, and therefore, does not provide any information about how to solve them. Moreover, a strict female/male binary approach can embed gender stereotypes in the software.

Lack of approaches in RE. When awareness is raised regarding gender issues in the software development process, there are few concrete approaches to identify and address them in the early stages, specifically in the requirements phase. In fact, we did not find any requirements engineering methods or tools to tackle gender issues from a non-binary perspective nor common and practical definitions for gender requirements. In the absence of a concrete method to address gender in requirements, software systems are failing to include diverse perspectives and are statistically privileging male users, disadvantaging and discriminating against users who identify with other characteristics.

RQ2. What are the software application domains where gender issues were addressed? The results revealed a broad set of software application domains where gender issues were addressed. We found several different and specific application domains, and therefore we aggregated the selected studies according to one of the following criteria: the article was classified as the more general domain from which they are part of, or, in case there was a significant number of articles from a more specific domain, we aggregated them together as a separate domain from the general one. From the selected studies, *Human Computer Interaction* was the domain with more papers [9,12,13,17,37,42,44,46–48] (10 out of 36), providing evidence of gender differences in the interaction and performance in computing tasks and presented a concrete focus on the inclusion of gender perspective in the technology design. The second application domain was *End User Applications* [24–29,33], where gender issues were found in problem-solving software, with 7 studies. The third was *Open Source Software* [8,15,41,45] which contributed with 4 studies, mainly in participation and contribution to project development. Furthermore, the application domains *Web Applications* [10,19,43], *Mobile Applications* [11,30,32] and *Game Development* [18,35,36] included the same number of articles, with 3 out of 36, followed by *Machine Learning* with 2 studies [38,39], sharing the common goal of preventing algorithms from perpetuating gender stereotypes. The application domains in which gender issues have been least addressed are *Social Networking Services* [31], *Requirements Engineering* [34] and *Smart Cities, Mobility & IoT* [40] with 1 study each. With the exception of the *Machine Learning* domain, where gender issues refer to bias in datasets and the creation of algorithms, empirical research on significant gender differences in attitudes towards software was presented across software domains.

RQ3. What gender issues impact software users? We found a set of studies that described empirical evidence on gender differences in attitudes towards software that are not accounted for, and thus, negatively impact its users. However, most studies only provided results for female and male users (see RQ1). Therefore, we present the differences as they are stated in the studies, but acknowledging the limitations of this approach. These gender issues in the software disadvantage and negatively impact users who show different characteristics from those that the software privileges [9]. From the selected studies, we collected the set of characteristics that influence attitudes towards software, as shown in Table 1.

Statistically, female users are motivated to use technology as a tool to accomplish a goal, exhibit a comprehensive processing style, have low self-efficacy, show aversion to risk, and are process-oriented learners [9]. Male users see technology as a source of entertainment, process information in a selective and heuristic manner, exhibit high self-efficacy, are risk-tolerant, and learn by tinkering [9].

Gender issues arise as software is designed around the way male users tend to use software systems. For instance, female users may choose not to try an unfamiliar system or feature if the perceived risks are high in relation to the possible benefits. Yet, software requires a certain level of risk-taking from its users to be fully used, disadvantaging those who are more risk-averse [9]. Furthermore, the characteristics are interrelated and mutually influenced. Self-Efficacy is positively correlated to Perceived Ease of Use for female users [29] and related to Willingness to Learn [24] and Tinkering [25]. Motivation for using the system is also related to the Willingness to Learn different features [29], which may be lower for female users [28], and to the interaction Environment, where female users are more motivated by a collaborative Environment, rather than a competitive one [26], and gamification had no effect on their motivation or performance [41]. The perceived Cost–Benefit of learning may also be higher than a male user's perceived cost to learn the same feature [33].

Moreover, perceived Ease of Use is more important for female users, while male users are more influenced by perceived Usefulness [40]. Regarding Visual Design, Sense of Belonging and consequent Motivation for using the software system are impacted negatively when user interfaces are driven by gender-biased design choices in aesthetics, images, and language [19]. Although female users were more affected by the stereotypical images, male users were also affected, and both benefited from the gender-inclusive design [19]. Visual Design is also more important and impacts levels of Credibility for female users [32]. Moreover, they are statistically more likely to process information comprehensively, and thus, are more accurate in assimilating and decoding verbal and nonverbal Cues [19,43].

If the Language or Communication Style of an interface or online community is not gender-inclusive (e.g., masculine gender-exclusive language, such as using "guys" when referring to a group or "he" as default, "boy's club", sexist language) their Sense of Belonging, Willingness, Self-efficacy, and Motivation to engage decreases [15,19]. The online programming community StackOverflow negatively affects women's participation through its male-oriented design, culture of criticism, and unwelcoming, competitive, and hostile environment [15,41].

Furthermore, users with female characteristics are more concerned about Privacy [11] and are more socially oriented [30] therefore prefer technology that enables them to be available and connect to others, specially with people they already know [31].

Table 1
Characteristics that influence attitudes towards software.

Characteristic	Description	Concept
Risk	Perception of possible outcomes when using a software [9]	Perceptions
Financial Cost	Perception regarding the financial cost of a software [30]	Perceptions
Ease of Use	Perception that using the technology will be effort-free [29]	Perceptions
Usefulness	Perception that using the technology will provide utility [29]	Perceptions
Credibility	Level of credibility attributed to a software [32]	Beliefs
Trust	Belief in the reliability and trustworthiness of a software [30]	Beliefs
Privacy	Concerns and behavior for privacy when using a software [11]	Beliefs
Cost-benefit	Earned benefit compared to the cost of trying a software [33]	Beliefs
Self-efficacy	Belief in the ability to use software in varied situations [9]	Beliefs
Sense of Belonging	Feeling of fitting in with an online culture or community [19]	Beliefs
Linguistic &Communic. Style	Linguistic and communication styles in an online community/website interface/software [19]	Preferences
Visual Design	Aesthetics of the software interface, including imagery,colorfulness, complexity, and fonts [32]	Preferences
Cue Detection	Cue detection in interface design, language, and communitynorms of a particular software [32]	Skills
InformationProcessing Style	Strategies for processing new information and solvingproblems in a software task [9]	Skills
Awareness	Previous experiences and knowledge about a software [15]	Skills
Willingnessto Learn	Desire to acquire knowledge about a software [24]	Motivations
Motivation	Reasons behind one's behaviors towards software [9]	Motivations
Tinkering	Exploratory behavior when using a software [9]	Motivations
Time Commitment	Time one has available for using a software [36]	Responsibilities
RoutineIntegration	Software compatibility with one's habits, behavior, patterns,and environments [40]	Responsibilities
SocialInteraction	Type and quantity of social interaction provided by anonline community/software [30]	Environment
Environment	Conditions for interacting with the software [26]	Environment
Community	Size, culture, and environment of an online community [15]	Environment
Access	Access to technological resources [40]	Access

They also have less Access to digital resources [40], less previous technological experience [15], are more sensitive on cost of purchasing [30], and have less Time available to interact with the system and can be discouraged if it compromises too much time to learn [35,36]. Thus, the compatibility of the system with their everyday Routine is essential for its use [40].

RQ4. What approaches have been proposed to address gender issues in software development? Few existing works propose approaches to address gender issues during the system's development. One of these works is GenderMag [9], a method for detecting and fixing gender inclusivity bugs in software through a systematic process that evaluates its features. The process is based on a set of personas that represent a range of distinct users structured around five underlying gender differences in problem solving skills: motivation, information processing style, computer self-efficacy, risk aversion, and tinkering. Results showed that applying GenderMag improved the software's inclusiveness and eliminated the gender gap in the software's design [37].

IT & me [46] uses gender-sensitive personas in the design process of software to ensure the diversity of female perspectives. Personas were developed with an agile, iterative approach model that involved potential users, preventing the integration of stereotypical gender assumptions in the software platform design [46]. In [40], a conceptual model that relates gender differences in daily mobility patterns and social roles with user acceptance of smart mobility technologies is proposed with the aim of considering the needs of all users during development. Another study [33] proposed an end-user profile formation approach, 'RULES', to be used in behavioral modeling implementations, consisting of five behavioral attributes influenced by the user's gender, namely risk, usefulness, and ease of use perception, learning willingness, and self-efficacy. Yet another work [18] proposed an integrative approach to understand and evaluate gender inclusiveness in game development through a framework that guides the design of gender inclusiveness in games.

Lastly, [34] uses a modified coherence method to overcome gender differences in communication patterns by adding a structure to requirements elicitation and allowing clear information transfer in interviews between mixed gender users and analysts.

2.3. Threats to validity

In this study, we took actions throughout the planning stage of the systematic mapping study to mitigate several potential threats and consequently increase the correctness and completeness of the results. However, the research practices and methods used, like all studies, have limitations and implications that constitute threats to the validity of the resulting findings. We followed the proposed guidelines in [50] for lessening the following threats:

- *Construct validity*. The completeness and correctness of the conceptual model depends on how well the mapping study was conducted. One threat concerns the search string not including all the relevant keywords. This was mitigated by validating it within the authors and an external reviewer.
- *Internal validity*. To minimize the threat of not including all the relevant information for this study, we attempted to gather all the keywords' synonyms that represent the research questions and tested several search strings based on the number of relevant results they retrieved, and chose the most optimal. However, since the number of studies was high, we could have missed some relevant work due to exhaustion.

- *External validity*. To ensure we did not miss relevant research work regarding our purpose, we opted to perform the automatic search in Google Scholar, which provides a very large set of publications in Software Engineering, to ensure we cover the maximum results possible. Additionally, we used the Snowballing approach. However, we did not consider Non-English or unavailable studies.
- *Conclusion validity*. We saved the retrieved studies in the Google Scholar's bibliography management tool at the time of the review, and used the data extraction form to structure the data necessary to answer each RQ.

2.4. Related work

Stumpf et al. performed a conceptual review on gender-inclusive HCI design that provides an overview of the motivations, the state of the art, and possible future work for this area [51]. It presents economic, ethical, and political motivations for addressing gender in HCI, a review of gender differences in cognitive and behavioral styles that influence the interaction with technology and differentiates them from gender stereotypes. Díaz et al. observed differences between women and men in effort and accuracy of elicited requirements, and concluded that mixed teams yield the most optimal results [52]. In addition, Aksekili et al. showed that organizational support and policies for women's advancement improves team work quality and performance of agile software development teams [53]. The work presented here addresses the relationship between gender concepts and the software development process and how it impacts the developed software and how users interact with it, based on the systematic mapping study we conducted.

3. A conceptual model for gender-inclusive requirements

This section discusses the conceptual model for gender-inclusive requirements and its development process, followed by a description of its four main components with a brief illustrative example. Finally, we present a prototype tool as a proof of a concept for using the model.

3.1. Gender-inclusive conceptual model

The proposed conceptual model for gender-inclusive requirements was constructed as the UML class diagram, shown in Fig. 1. For clarity, we omitted from this diagram the attributes (defining characteristics) of each class, but available in Table 1.

The development of the conceptual model followed an incremental three-step process based on the methodologies proposed in [54,55]: specification, knowledge acquisition, and conceptualization. In the **specification** step, we defined the domain and purpose of the conceptual model and established its scope. The domain of the conceptual model concerns the intersection of gender with software technologies, exploring the mutually influencing relationship between gender and software development and software itself. Its purpose is to find and mitigate gender issues from the beginning and throughout the software development life cycle and support requirements engineers in the elicitation of gender-inclusive requirements that, if satisfied, will make software systems inclusive for every user. In the **knowledge acquisition** step, we gathered the required knowledge for building the model. We resorted to the findings of the systematic mapping study to identify and define the concepts and relationships that compose the conceptual model. We used the domain classification criteria to define its scope of application and the categories *Social Construct* and *Intersectional* for its gender conceptualization. Finally, in the **conceptualization** step, we classified and structured the previously collected domain knowledge to create the conceptual model. The last two steps were carried out simultaneously, alternating between the two according to the development progress. Moreover, we followed a middle-out approach when defining the concepts and their relations, as recommended in [54]. The end result is a conceptual model composed of four main components: *Gender*, *Sociocultural Context*, *Human Actor*, and *Software System*.

Next, we describe the concepts of each component and illustrate them using the *car-sharing System* [56]. This system allows users to rent cars via a web or mobile application, and charges based on the rental period, distance, and car model. The cars can be provided by individuals or companies, to which an identifier is assigned at the time of registration that will serve to locate them in the web and mobile application [56]. Also, the car should be parked anywhere within the operational area of the car-sharing service [56].

Gender. Gender is conceptualized both as a structural feature of society and as a complex part of an individual's *Identity*. The *Structural Gender* represents the normative societal attributes and opportunities based on the perceived *Biological Sex* of the *Human Actor*, including *Gender Norms* and *Gender Roles* [47]. In the car-sharing example, these represent the gender division of labor that translates into different needs and motivations for using this alternative mobility system. Furthermore, *Structural Gender* is defined and reinforced in a particular *Sociocultural Context* through social expectations and interactions, creating gendered power asymmetries in society and establishing gendered systems that perpetuate inequality. For instance, uneven socioeconomic conditions may result in less access and use of software technologies, and consequently, in less technical knowledge and experience levels [6,15]. In this context, *Gender Stereotypes* emerge and are internalized by *Human Actors* that perpetuate them by conforming to the expected behavior and reinforcing it to others [38,45]. Hence, *Structural Gender* influences the *Individual Gender*. Regarding the car-sharing system, the influence of *Structural Gender* in the *Individual Gender* results in differences in individual daily routines and mobility patterns that can be grouped by gender [40].

The *Individual Gender* represents the *Human Actor*'s self-identified gender [47], which is one part of their *Identity* [49]. We represent it through the individual's experience of gender, their *Gender Identity*, which is independent of their *Biological Sex*, and

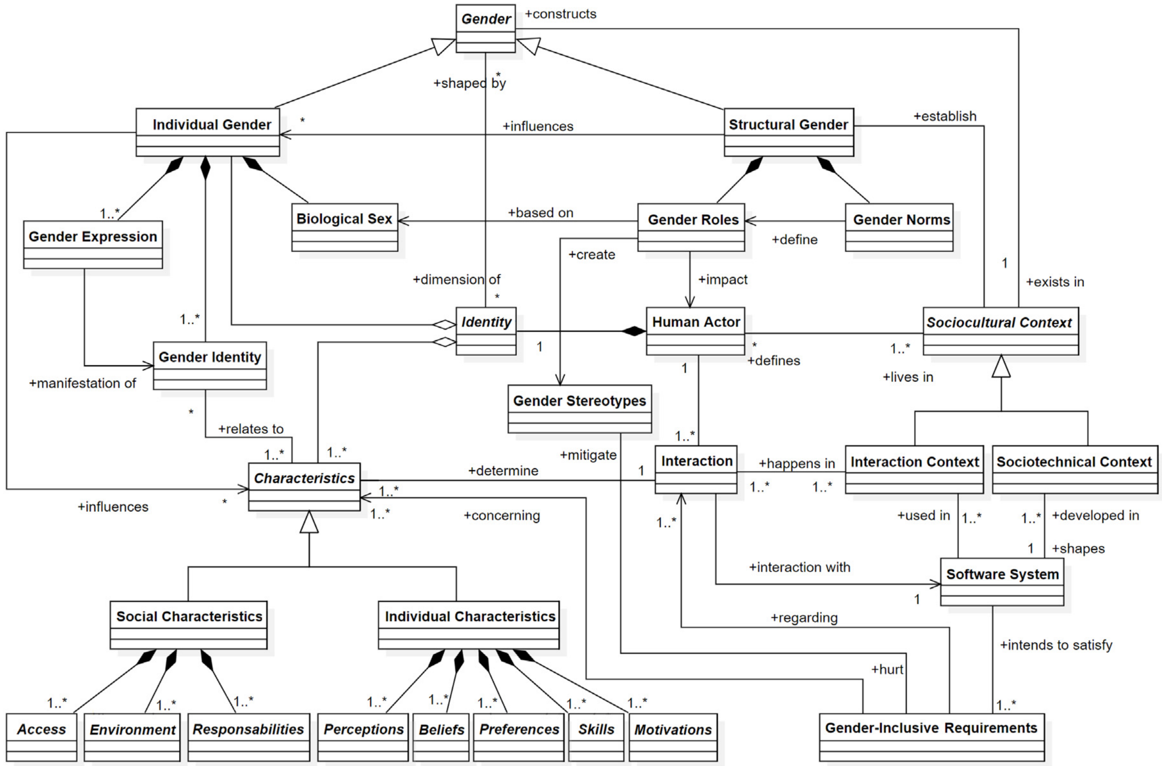

Fig. 1. Conceptual model of gender-inclusive requirements.

their *Gender Expression*, the manifestation of their *Gender Identity*. Both *Gender Identity* and *Gender Expression* are conceptualized as a continuum where the *Human Actors* can align themselves to describe how gender shapes diverse behaviors and attitudes towards software technologies and go beyond the gender binary [48]. Furthermore, the *Human Actors* can vary their *Gender Identities* according to the context of the *Interaction*, which allows the construction of fluid and complex identities [17].

Sociocultural context. This represents the social and cultural factors and events of a particular time period, where *Gender* is constructed and where *Human Actors* live in and are defined by. *Gender* is constructed within a *Sociocultural context* and institutionalized through socialization processes and systems, represented by *Structural Gender*. Both *Individual Gender* and *Structural Gender* have an influence on the software development process and on the elicited requirements of system to-be, represented by the relationships between these two concepts and the concept of *Sociotechnical Context*.

The software development process, especially the requirements engineering phase, is shaped by a range of social factors and contextual influences, such as the needs and values, represented in the model as *Characteristics*, of development teams, owners, individuals with decision-making power, organizations, and communities. These determine what is elicited and prioritized into requirements for the system, and also what is excluded. However, their preferred needs and values do not reflect those of the software users because this context is defined by a very narrow subset of the population: it is overwhelmingly male and with high levels of education and income [12,39]. Both open-source and closed-source software systems are developed and maintained in male-dominated environments where acknowledging different contexts is overlooked under the assumption of technical neutrality and thus, embody and reflect the masculine culture and identity of its creators [42,47]. In contrast, the *Human Actors* who interact with the *Software System* can be highly demographically diverse and subject to the impact of *Gender* in different ways. Thus, they have diverse experiences, opportunities, roles, responsibilities, and levels of access and decision-making. These differences in behavior, preferences, and needs imply distinct attitudes towards software technologies, represented in the model as the *Interaction Context*. Since the *Software System* is the product of the values and priorities of its *Sociotechnical context*, the historical patterns of gender discrimination, the consequent lack of gender diversity, and limited decision-making power from underrepresented *Human Actors* restricts the discussion of different viewpoints.

Therefore, we emphasize the distinction between the two contexts to provide a clear understanding of how ignoring the *Sociotechnical context* of software during development may unintentionally lead to systems that replicate the existing structures of gender inequality in society. This asymmetry is particularly noticeable in the car-sharing system. Due to the ubiquitous nature of the system, the *Interaction context* can be complex, unexpected, and highly gender-diverse. The gender gap between the *Sociotechnical context* and the *Interaction context* will be significant. The *Socio-technical context* is decisive for both the elicited information for the definition of the *Gender-Inclusive Requirements*, and also for the car-sharing *Software System* that will be developed based on those requirements. Thus, requirements engineers must carefully plan the elicitation process to mitigate gender issues such as unequal status and decision-making power in the development context that may result in masculine perspectives being prioritized [12] and the possibility of *male hegemonic identity* where minorities adopt the mainstream culture of the company for integration and avoid introducing gender perspectives in group sessions [42]. Diverse groups of stakeholders should be consulted, including those who have substantially less power or influence, and specifically elicit information from minorities through individual interviews or group sessions where they are the majority. When addressing gender specifically, inclusive language should be adopted, for instance, by avoiding binary language such as ''both genders'' or describing users as opposites to each other, but rather using they as default pronouns. A *Sociotechnical context* where gender inclusion is intended and addressed is the first step towards eliciting the information required for the formulation of the car-sharing system's *Gender-Inclusive Requirements*.

Then, the information for formulating *Gender-Inclusive Requirements* must be elicited through the understanding and specification of the context of technology use, the *Interaction Context*, which is itself deeply interwoven with understanding people's identities and everyday practices. These are represented in the model through the concepts of *Human actor*, *Identity*, and *Characteristics*, which we discuss next.

Human actor. This represents the individuals who are involved with the Software System. The *Human Actor* lives in a particular *Sociocultural context* and has a unique *Identity* that is characterized by their *Individual Gender* and *Characteristics*. In the car-sharing system, they represent the individuals that provide their cars and those who rent them. The *Human Actors* are subject to the structural gendered systems of the *Sociocultural context* they live in and participate in the social construction of gender by both embodying the *Gender Roles* from their social context and perceiving others based on the same social norms. Hence, their *Identity* is also shaped by *Gender*, and consequently, their *Characteristics* as well.

The *Characteristics* of the *Human Actor* represent the individual and social traits of a *Human Actor* that determine their attitudes towards software technologies. These personal attributes were empirically found to determine the *Interaction* with the *Software System* and are influenced by the user's *Gender* (see Table 1). We organized them as either *Individual*, representing the personal attributes of a *Human Actor*, or *Social*, representing those established in relation to others. This classification facilitates the selection of the most suitable *Characteristics* for the type of *Software System*. *Individual Characteristics* include *Perceptions* (Perceived Risk, Perceived Financial Cost, Perceived Ease of Use, and Perceived Usefulness), *Beliefs* (Credibility, Trust, Privacy, Cost-benefit, Self-efficacy, and Sense of belonging), *Preferences* (Linguistic and Communication Styles and Visual Design), *Skills* (Cue Detection, Information Processing Style, and Awareness), and *Motivations* (Willingness to learn, Motivation, and Tinkering). *Social Characteristics* include *Responsibilities* (Time commitment and Routine integration), *Social Environment* (Social Interaction, Environment, and Community), and *Access* (Access to technological resources). The majority of the studies from where these attributes were collected were previously classified as *Binary* or *Binary, Social Construct* (see Section 2). However, because we conceptualize *Gender Identity* as a spectrum, we do not expect a *Human Actor* to be consistent with one gender for all the *Characteristics* but rather vary because these result from complex and contextual individual experiences and social attributes, opportunities, and relationships. For example, a *Human Actor* can be characterized with an information processing style statistically more prevalent among female users while showing attitudes towards risk more observed in male users, regardless of their gender identity. Requirements engineers should specify the appropriate *Characteristics* for the system. The car-sharing system is intended to be used in the everyday personal and professional life of its users, and for example, *Routine Integration* was selected. This *Characteristic* supports the uncovering of gender specific travel behavior by emphasizing the differences in everyday-life situations. For instance, statistically, women manage professional, family, and housekeeping responsibilities as well as leisure time activities [57], and thus, their routines are more complex in activities, requiring the integration of the system into their daily activity patterns for using it [40]. Requirements engineers should act as facilitators to ensure these *Characteristics* are used to elicit information that would otherwise be missed, rather than categorizing users. The *Characteristics* should be used as a component of the users' *Identity* to encourage those involved in the development process to engage with diversity and think of more complex and realistic users in semi-structured interviews, questionnaires, or creativity techniques.

This conceptualization of the *Human Actor* follows the 'quality of pluralism' from [44], that seeks to recognize the complex and unique identities of users across *Sociocultural contexts* to foster engagement with diversity and challenge the homogeneous points of view that underpin assumptions made about users in the early stages of development. Using personal attributes that are part of the individuals' identities preserves their individuality and complexity and it allows addressing gender issues without categorizing users. Thus, we can consider diverse groups of *Human Actors* and account for behavioral diversity among them during *Interactions*. These can happen in various *Interaction Contexts*, which can be distinct from the *Sociotechnical Context* where the *Software System* was developed and where the *Human Actor* may or may not have participated. The *Human Actor* expects the *Software System* will be inclusive and support their *Individual Characteristics* and *Social Characteristics* for a successful *Interaction*.

Software system. This represents the *Software System* the *Human Actor* interacts with in an *Interaction Context*. It captures a software system as the product of human-based decisions in a *Sociotechnical Context* influenced by *Gender*. However, this context is typically not representative of the diverse *Characteristics* of its users. Therefore, an analysis of whose *Characteristics* are being prioritized and which ones are being neglected is required to assess whether the decisions being made discriminate against users that are underrepresented during the software development process. For instance, discussing *Routine Integration* in the car-sharing system allows the rectification of gender-based issues, and thus, contributes to building a software system that will benefit all its users, regardless of gender differences in the human actors' life and work responsibilities. To achieve this, the *Software System* intends to satisfy *Gender-Inclusive Requirements*.

The *Gender-Inclusive Requirements* concern the multiple and diverse *Characteristics* of *Human Actors*. In turn, the *Human Actor* expects the *Software System* to be inclusive and support their *Individual Characteristics* and *Social Characteristics* for a successful *Interaction*. Hence, the information for the formulation of these requirements must concern not only the users and their characteristics, but also the contextualization of users as individuals in a social environment, the *Interaction Context*.

The *Gender-Inclusive Requirements* are not intended to be fixed and separate, but rather capture the needs and perspectives of diverse users and integrate them into the functional and non-functional requirements according to the *Software System* domain. In the car-sharing system, addressing the concept of *Individual Gender* could formulate a requirement such as *"the system must provide users with the agency to define their gender identity (change names, pronouns at any time)"*. From the *Characteristics*, *Routine Integration* and *Motivation* could formulate *"the system should provide the car locations evenly distributed, including near schools and hospitals"*. Moreover, *Trust* and *Security* could elicit information to define *"the system should provide users with the option to save trustworthy cars"* and *Language & Communication Style* could, for instance, enable the formulation of *"the system should provide a mechanism for reporting reviews that contain sexist, misogynist or targeted language"*.

In conclusion, the proposed model describes gender and related concepts that impact software development and its users to support the elicitation of information that will lead to gender-inclusive requirements. Towards this goal, we created a process for integrating the knowledge of the model in elicitation activities, which we describe next.

3.2. GIRE: A proof of concept

As proof of concept for a process that would support the integration of the model in requirements elicitation, we described a document that would be available for each concept with the following structure:

- **Description.** Concept definition to help its identification and memorization quickly without resorting to additional documentation.
- **Goal.** Description of the concept's purpose. For requirements engineers, it allows a more informed and comprehensive choice about its integration (or not) in the elicitation. For stakeholders, it helps discern its intent to understand the type of information to share.
- **Gender Issues.** Description of gender issues that may arise for the system and its users if the concept is not addressed. These help requirements engineers identify possible gender issues in the information provided by stakeholders, and also allow the assessment of their own assumptions.
- **Recommendations.** Description of recommendations for overcoming the gender issues mentioned before. These intend to provide requirements engineers with the knowledge to mitigate gendered information and suggest solutions.
- **Questions.** Customizable question-based checklist to support the application of the above recommendations. These are not exhaustive but rather the starting point for a discussion. They are structured to focus on the software system and what it can provide to meet the goals of the respective concept. However, they can be adapted to reflect the objectives of the elicitation technique in which they will be used. For instance, the following question for the *Motivation* concept: *"Does the system improve all users' motivation to learn new features equally?"* can be restructured to *"How would you learn a new feature in this system?"* or *"Do you feel more motivated to learn a new feature for what it can accomplish for you or more for enjoyment?"*.

The documents' main goal would be to assist software and requirements engineers in capturing the model's knowledge, facilitate communication, and improve decision-making during elicitation. To demonstrate this, we created a preliminary version of the Gender-Inclusive Requirements Engineering (GIRE) process. GIRE consists of three steps: select the concepts to be addressed, select the intended questions for each concept, and lastly, create a document with the complete selected information. The final document is intended to be used in semi-structured interviews, questionnaires, brainstorming or creativity techniques.

Then, we created a proof-of-concept for a web-based tool, the GIRE tool, to illustrate how the conceptual model could be used through the implementation of the GIRE process. Such tool would consist of three main tasks that represent the three steps of the process: select the concepts to be addressed, select the intended questions for each concept, and finally, provide details, such as author and participants' names, for the complete document that will support the elicitation activity. To illustrate the feasibility of the GIRE tool, we implemented a prototype for it using React, an open-source JavaScript library for building user interfaces.[2] Moreover, the GIRE tool itself intends to be a gender-inclusive tool. Accordingly, we developed the features of the prototype tool using the conceptual model, and thus using the prototype tool also as a brief example. Given the application domain of the tool, we selected the characteristics related to end-user development [9], which we will address throughout this description of the prototype: *Motivation, Self-efficacy, Information Processing Style, Tinkering*, and *Risk Perception*. Nevertheless, the prototype is not meant to be extensive, but rather simulate how the tool would function. Therefore, the application contains a single interactive path, while the others were kept static. For instance, to start the interactive three-step process, the workspace option should be] selected in the first, opening page.

Once in the personal workspace, the option to create a new document should be selected, showing the page that corresponds to the first task. In it, the GIRE tool presents the complete conceptual model so that process-oriented users can see the full model and understand what concepts they want to select, as shown in Fig. 2. In addition, for tinkering users, each concept is clickable so they can interactively explore the conceptual model and feel curious about its details.

The details for each concept include those presented in the document, namely its definition, potential gender issues, recommendations for addressing them, and finally a customizable question-based checklist for assisting in the implementation of the

[2] The prototype can be found in the following link: https://github.com/inesnunes/GIREtool

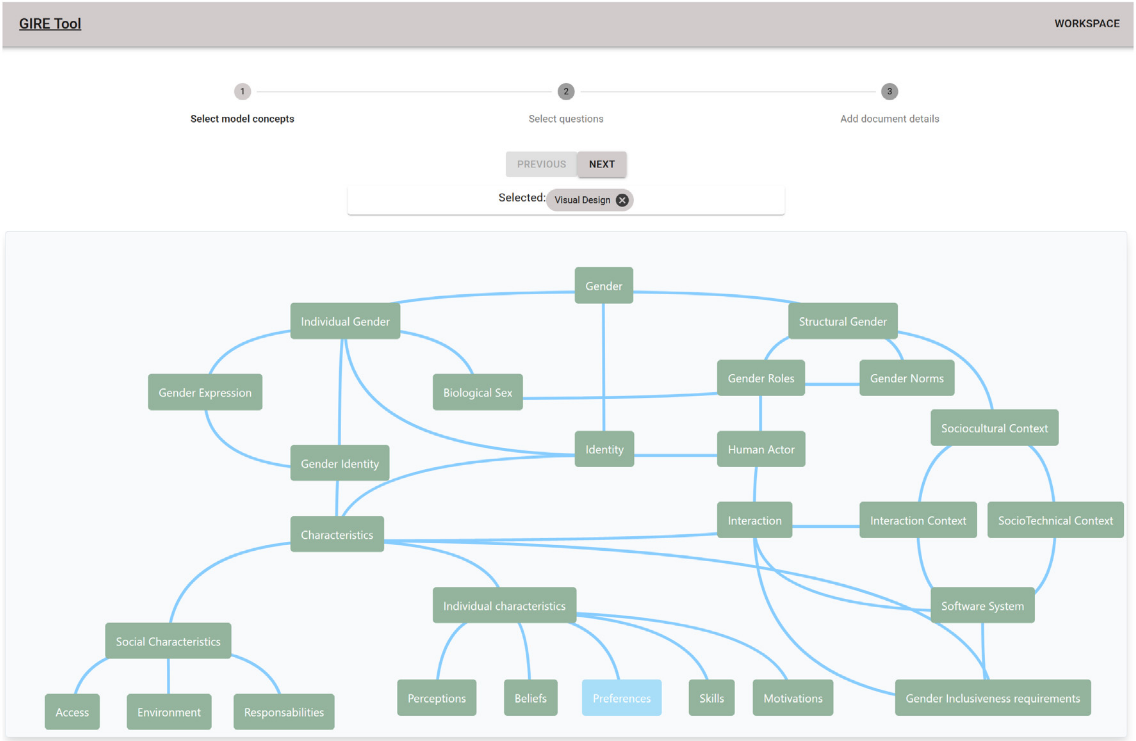

Fig. 2. Concept selection in the GIRE prototype tool with "Visual Design" selected.

recommendations. The full details are only shown if that option is selected, allowing support for varying degrees of self-efficacy users by providing the full details if required to complete the process or skip them, as they are optional, and complete the task effectively. For the interactive path, the option *Preferences*, which contains the concept of *Visual Design* should be added to the document.

Furthermore, the GIRE prototype always shows a top bar that indicates the task in which the user is in each moment. This can make users with lower self-efficacy feel confident about what they are doing as they can see in what task they are currently on, what they have done, and what tasks follow. This also supports risk-averse users. Then, the page for selecting the questions only shows them and the selecting options so task-oriented users can focus on that task only and feel motivated to finish it. Here, the prototype shows three questions associated with the selected concept, *Visual Design*, that can be added to the final document. To continue the interactive path, at least one question must be selected — for example "Can the visual design be discriminating against any user of any gender identity (e.g., does it include stereotypical masculine elements)?" — otherwise the next button is kept disabled.

In the last task, the prototype tool presents a form for naming the document, the author and the participants in the elicitation process. After confirmation, the input information from the complete process is stored in the workspace, which can be deleted, edited, or exported into the final document. If the latter is selected, the prototype tool automatically generates the document, as a docx file, with the author and participants details, each concepts' descriptions, recommendations, the selected questions in plain text in English each with a checkbox, and space for requirements engineers to add information, such as write down answers, reminders, and ideas.

4. Evaluation of the conceptual model

A preliminary evaluation of the conceptual model consisted of a static validation [50], which involves presenting and analyzing a proposed solution without requiring it to be used [50]. Its purpose was to collect perspectives, from novices and experts in conceptual modeling, on the completeness of each of the four components of the conceptual model, the usefulness and difficulty of understanding of the model, its overall positive and negative points, and finally, if they would use and/or recommend it. Therefore, the proposed model was presented through a presentation guide to the participants who then answered a questionnaire.[3] The invitation for the evaluation was sent to the participants via e-mail, which contained a private link to the guide and the questionnaire.

[3] The presentation guide and the questionnaire form are available in the following link: https://drive.google.com/drive/folders/1bNYzaca5PPBTWTA7FvVAmlJ6WgngsYdS?usp=sharing

4.1. Guide and questionnaire

To introduce the conceptual model to the participants, we created a presentation guide, with a total of 20 pages of explanatory text in English and images of the conceptual model and its components. To provide participants with the context of the work, the guide included the motivation for the development of the model, the main gender issues that we intend to address with it, and the goals of the work. Then, an explanation for each component as well as the definitions for all the concepts of that component were provided. Finally, it presented an explanation of the overall model. Reading it was required before answering the questionnaire.

The questionnaire, asking participants to evaluate the conceptual model and its components, was structured in nine sections:

- **Introduction** Introduces the context and the objectives of the evaluation questionnaire, informs that the answers are anonymous, provides the link to the presentation guide and informs of its reading necessity to answer the questions that follow.
- **Pre-survey** Collected demographic information and opinions on gender inclusion in software development. Although gender inclusion has been receiving increasing attention in software engineering, the field is still introducing the topic. Therefore, gathering perspectives on gender inclusion is relatively significant to understand the extent to which the participant values this topic, but does not find the model useful, or does not consider it a relevant matter and thus neither the model.
- **Overview of the conceptual model** An informative section with an image of the conceptual model with its four components highlighted and a brief description of its main objective.
- **Four sections for each component of the model** Evaluates each component's completeness by asking if the component expresses what it intends to express with the multiple choice options of "Yes" or "No". If answered negatively, the participant could provide their reason in an open field question.
- **Overall opinion of the conceptual model** Evaluates the conceptual model's usefulness and difficulty of understanding through a likert scale of options, from one to five, where one is "Not useful" and "Not difficult" and five is "Very useful" and "Very difficult" respectively. It evaluates its positive and negative aspects with two open field questions, and finally, if the participant would use or recommend it, with three options: "Yes", "No", and "Perhaps".
- **Guide evaluation** Assess the quality of the guide by asking two multiple choice options of "Yes" or "No": if the provided guide was useful and if it was easy to understand.

Even though the information and the questions were written in English, the answers could be given in English or Portuguese. Nevertheless, all participants chose to answer in English.

4.2. Participants demographic

To evaluate the conceptual model, we considered two groups of participants: (1) experts and academics and (2) industry practitioners. The first group included both authors with expertise in computer science who have addressed gender in their work, and experts in conceptual modeling that could have addressed gender. We considered the list of authors from the main studies of the systematic mapping study (see Section 2). On the other hand, the second group included academic staff as well as industry practitioners from diverse areas. We included this group of participants to understand how the conceptual model was perceived by individuals of different backgrounds (e.g., as academic training and degrees). We invited the participants by sending an email to each one with an introductory formal text, a brief justification for why the person was receiving this email and the invitation to participate in the evaluation followed by the link to the form.

From a total of 31 participants, fifteen were women (48.4%), fifteen men (48.4%), and one non-binary (3.23%). Regarding the scientific domain of the participants' degree, we divided them into three groups: Computer Science with 71%, a degree that is not Computer Science but it is related to science or engineering with 12.9%, and finally the degrees related to Languages and Humanities, representing 16.1%. Furthermore, Fig. 3 shows the participants' academic qualifications, with 6.45% of the participants were Bachelor students, 22.6% completed Bachelor, 32.3% a Masters Degree, and 38.7% a PhD. Regarding their current occupation, 35.5% are students, 22.6% academics, 22.6% practitioners, 16.1% researchers, and 3.23% researcher and practitioner, as shown in Fig. 4.

A total of 9 (54.8%) participants had less than two years of experience in conceptual modeling. However, from the 9 participants who do not have a Computer Science degree, 7 (77.8%) selected this option. Furthermore, 19.4% of the participants had three to five years of experience and 12.9% had six to ten years. Finally, we had a total of 12.9% of participants with more than ten years of experience.

4.3. Perceptions on gender issues

First, we asked participants to rate their opinion on the importance of gender inclusion in software development in a likert scale as shown in Table 2. A total of 27 in 31 participants considered gender inclusion in software development very important or important, thus suggesting their interest in this topic.

Then, we sought to understand the extent to which the participant is unaware of existing gender issues in software development, and therefore, does not consider the conceptual model needed or relevant. The results are presented in Table 3.

The results also indicate that the majority, 29 participants, are aware of current gender issues in software development (see Table 3): Fifteen (48.4%) relate them with culture and education, and fourteen (45.2%) are aware of the statistical evidence that

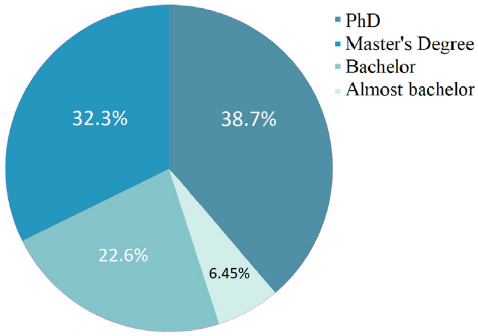

Fig. 3. Percentage by academic training.

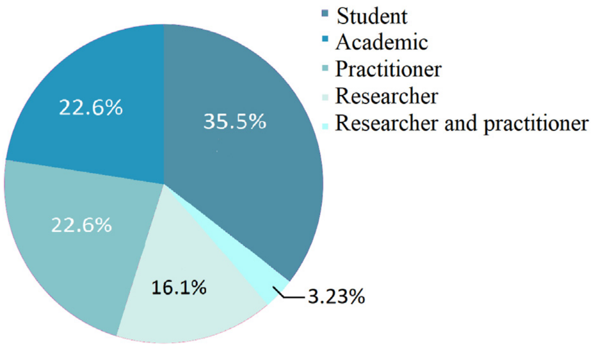

Fig. 4. Percentage by current occupation.

Table 2
Results for the importance of gender inclusion in software development.

Importance of gender inclusion in SD	Participants
5 (Very important)	18 (58%)
4 (Important)	9 (12.9%)
3 (Moderately important)	2 (6.45%)
2 (Slightly important)	0 (0.0%)
1 (No. important)	2 (6.45%)

Table 3
Results for the perspective on gender issues in software development.

Perspectives on gender issues in SD	Participants
"Persistent structural inequalities with statistical evidence".	14 (45.2%)
"Inequalities are mainly due to culture and education".	15 (48.4%)
"There are no barriers because there is female presenceand inequalities are circumstantial".	1 (3.23%)
"I don't have an opinion".	1 (3.23%)

demonstrates persistent structural inequalities. We included these two distinct perspectives because the conceptual foundation for the model concerns the evidence of structural inequalities, possibly unfamiliar to participants who associate inequality with culture and education. Thus, we sought to understand if there is a relation between the selection of this option and difficulty in understanding the model.

4.4. Components

Overall, results demonstrate that the concepts and relationships in the components are enough to represent what they intend.

Gender

Asked if this component *"captures the concept of gender as perceived by individuals (self-identity) and society"*, 93.5% participants answered positively. Only two participants (6.45%), with less than two years of experience in conceptual modeling, answered "No".

However, they are bibliographical authors with significant knowledge and experience regarding this components' concepts and provided valuable feedback on how to improve it: *"There may be differences in perceived gender, gender expression, and gender roles. (...) Gender roles are not necessarily based on Biological Sex. "Men" and "women" may not be the only categories of these roles if we look beyond a Western context".* These suggestions open the way for further refinement of the concepts and the inclusion of additional ones related to a potential dimension of culture in the model.

Sociocultural context

Asked if this component is *"enough for capturing the relation of gender to the sociocultural contexts of software use and development"*, 87.1% participants answered positively. Among the four (12.9%) who answered negatively, three (9.7%) had less than two years of experience in conceptual modeling and one had more than ten years of experience. In general, they raised questions regarding the oversimplification of this component's concepts that, from their point of view, was not adequate given their relevance in the model: *"I don't really think it's binary; I even think that the "Sociocultural Context" class shouldn't be abstract. Have you ever thought about the possibility of having a third class being a specialization of the two (Interaction Context and SocioTechnical Context) with multiple inheritance. In other words, I think it's not that simple".* The provided feedback raised important issues and questions that should be further explored for improving this component.

Human actor

Asked if this component is *"enough for capturing a person with a unique identity of individual and social characteristics influenced by their gender"*, 80.6% answered positively. Although this component had an overall positive evaluation, it had the most negative responses among the four, with six "No": Four (13%) had less than two years of experience in conceptual modeling, one (6.4%) had six to ten years, and one had more than ten years of experience. The latter provided the following feedback *"I think the environment should be specialized in Family Environment, Workplace, and Social environment".* However, this information was not found in the selected studies of the mapping study, and thus, we did not have evidence that supported their inclusion in the model. Nevertheless, this feedback constitutes initial evidence that these concepts may be integrated, if confirmed and validated by more experts in future evaluations.

Software system

Asked if this component is *"enough for capturing a software system as the product of human-based decisions in a sociocultural context influenced by gender"*, 93.5% answered positively. From the two (6.45%) negative responses, one (3.225%) had more than ten years of experience and the other (3.225%) had less than two. Both mentioned that this component is simplified and does not describe reality: *"(...) it's just really a fundamental issue with me in how the question is asked and what the model should achieve. Maybe it makes more sense to frame it as an invitation to look at different aspects for reflection instead of describing individual experiences".* We agree that one of the goals of the model is to encourage reflection in requirements elicitation. However, as a conceptual model, it also aims to describe and represent the concepts we identified in the mapping study, and thus, this criticism raises valuable concerns for future improvements.

4.5. Conceptual model

After the model's components, we evaluated the complete conceptual model. The results were positive for both usefulness and difficulty in understanding from participants with fewer years of experience in conceptual modeling to experts. The results regarding use and recommendation were also positive, particularly among more experienced participants. Finally, feedback from the open questions was overall appreciative and provided valuable knowledge for further improvement.

Usefulness

Regarding usefulness, fourteen (45.2%) participants evaluated the model as "5 - Very Useful" and twelve (38.7%) as "4 - Useful", totaling 83.9% positive answers. Three (9.7%) participants considered it "3 - Moderately Useful" and two (6.4%) "2 - Slightly Useful". Yet, as shown in Fig. 5, four out of the five participants that gave a 2 or 3 rating evaluation have less experience in conceptual modeling. Moreover, four experts (three with 6–10 years and one with 10+ years of experience) considered the model very useful, while three (one with 6–10 years and two with 10+ years of experience) considered it useful. Finally, no participant answered "Not useful". We perceive very positive results from this analysis as the feedback suggests that both less experienced and expert participants found the model mostly useful.

Then, we sought to understand if participants who did not consider gender inclusion an important topic did not recognize the model's usefulness. Thus, we crossed the data on perceived importance of gender inclusion in software development with the answer on usefulness. The results are shown in Fig. 6, where 1 represents the answer "Not important", 3 "Moderately important", 4 "Important", and 5 "Very Important".

From the two participants who did not consider the model useful, one selected gender inclusion as "1 - Not Important" and the other as "5 - Very Important", suggesting the value in differentiating those that do not consider the topic relevant, and likewise

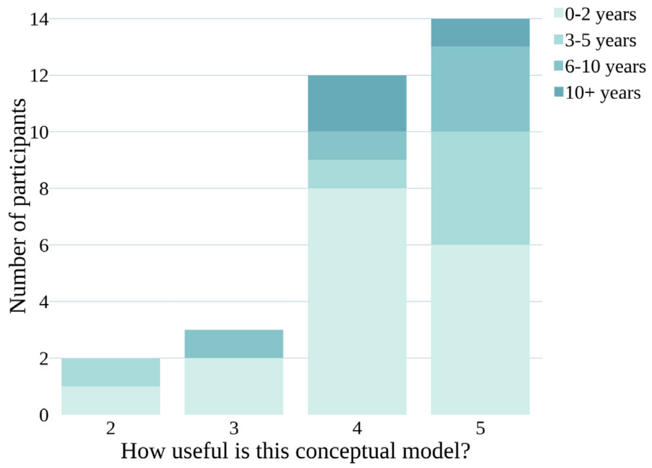

Fig. 5. Results for usefulness of the model by years of experience in conceptual modeling.

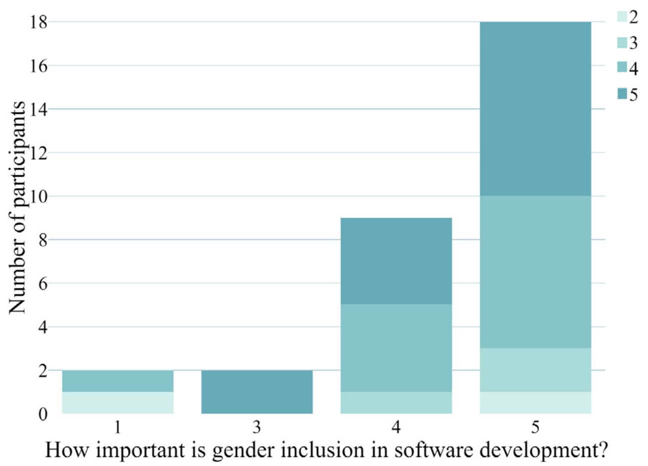

Fig. 6. Results for usefulness of the model by perceived importance of gender inclusion.

neither the model, or are critical of the model itself. Furthermore, the participants who considered gender inclusion a "3 - Moderately Important" topic all evaluated the model as "5 - Very Useful", suggesting that they may not consider it a priority or urgent topic, but still recognize the usefulness of the model. Overall, from the participants who considered the topic "Important" or "Very Important", there is a visible tendency of positive perspectives on the usefulness of the model. We consider this observation positive and valuable as participants who regard the topic as important held positive viewpoints of the model's usefulness.

Difficulty of understanding

After evaluating usefulness, we asked participants about difficulty of understanding the model. The results indicate that the majority considered it not difficult to moderately difficult to understand, with five (16.1%) participants answering "1 - Not Difficult", eight (25.8%) "2 - Slightly Difficult", and ten (32.26%) "3 - Moderately Difficult", as shown in Fig. 7. Participants with less experience in conceptual modeling had a range of perceived difficulty, however, this can be due to the fact that a participant may have little experience in conceptual modeling but have significant experience with gender-related knowledge, and therefore can understand the model with less difficulty.

To complement this discussion, we analyzed whether participants who did not have the knowledge on gender issues that constitutes the conceptual foundation of the model could still understand it. The results are shown in Fig. 8, where 1 represents the answer "I don't have an opinion", "2 - Inequalities are mainly due to culture and education", "3 - Persistent structural inequalities with statistical evidence" (the option that corresponds to the model's conceptual foundation), and "4 - There are no barriers because there is already female presence and inequalities are circumstantial".

Participants who chose the second option demonstrated knowledge on gender issues but not on all the concepts the model entails. This is visible in Fig. 8 as this group's perception of difficulty ranged from "2 - Slightly Difficult" to "4 - Difficult", with the majority

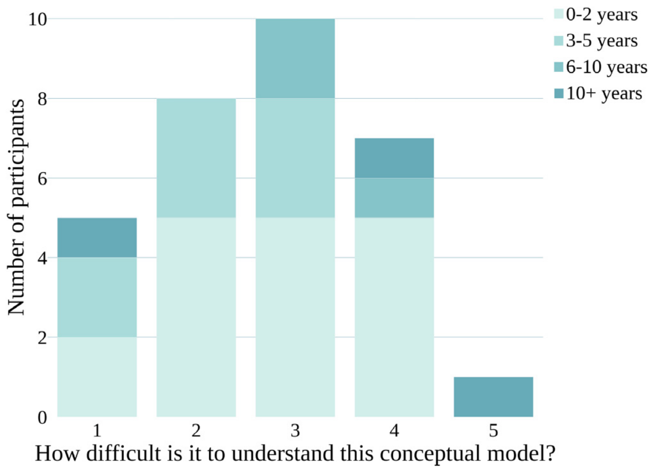

Fig. 7. Results for difficulty of understanding the model by years of experience.

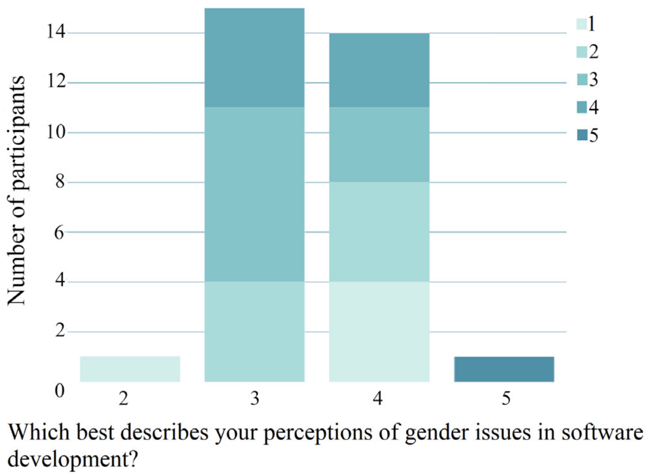

Fig. 8. Results for difficulty of understanding the model by knowledge on gender issues.

considering it "3 - Moderately Difficult". Furthermore, this analysis indicates that participants with a more thorough knowledge of gender issues in software development, option 3, understood the model with less difficulty. Finally, the participant that considered the model "5 - Very Difficult" to understand selected option 4. Although having more than ten years of experience, this participant may not be aware of gender issues, and thus, understanding the model may be more difficult at first.

Use and recommendation

When asked if they would use the model, seventeen (54.8%) participants answered "Yes" and twelve (38.7%) "Perhaps", as shown in Fig. 9. Considering the participants' demographics, we perceive 54.8% "Yes" responses as a positive result. Moreover, when analyzing the responses according to experience in conceptual modeling, the percentage of positive responses increased, with 75% of experts (more than 6 years of experience) answering "Yes". These participants have extensive knowledge of conceptual modeling, and thus, may have a more comprehensive perspective on the use and application of the model. Indeed, when asked if they would recommend the model, twenty five (80.6%) participants answered "Yes" and five (16.1%) "Perhaps", as shown in Fig. 10, in line with the previous observation.

Strongest points

Despite being an optional question, nearly all participants provided their perspectives when asked about the strongest points of the model, suggesting a likely interest and further analysis of the model. The feedback was positive and quite comprehensive as it provided important and diverse viewpoints on the value of the model.

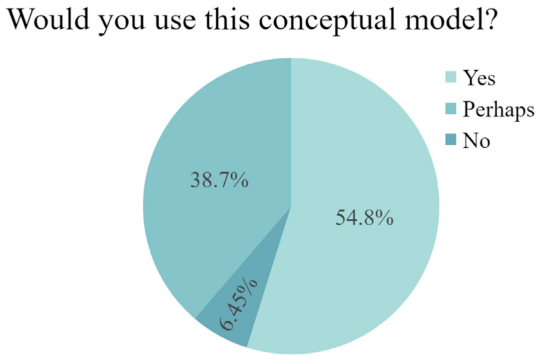

Fig. 9. Percentage of participants who would use the model.

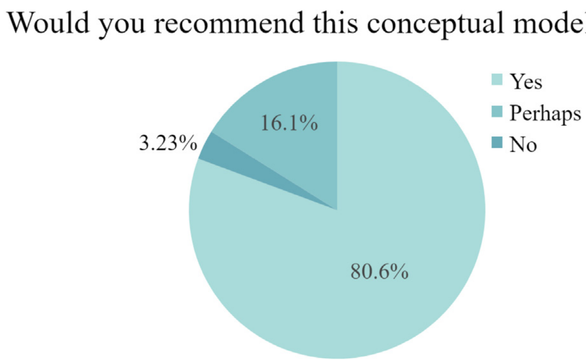

Fig. 10. Percentage of participants who would recommend the model.

For instance, participants mentioned why and in what contexts the model could be valuable for them: *"I think this model conceptualizes the key aspects of gender inclusiveness for software development very well. The compiled list of the Human Actor characteristics was a great resource to understand many different aspects to consider when striving to develop inclusive software"*. Moreover, they complimented the organization of the model and the adequacy and completeness of the concepts: *"I could see the use of this model as a translation tool to highlight the role of gender in software development and various aspects of gender to consider. I applaud the effort that clearly went into the model. I think the really key point is that context of development and context of use can be different"*.

This qualitative feedback is very encouraging and provides valuable information that complements and further explains the quantitative results.

Weakest points

As a qualitative question, the answers varied. However, we could group them into two main viewpoints. The first focused on the need to apply the model to a practical example as a proof of concept or integrate it into established practices to be adopted during software development: *"I would like to see a practical application of the model in an example case. It's hard for me to understand the impact of using the model without it. Although, as it stands, it's already an interesting point of discussion and a way to raise awareness for related issues"*. The comments point to an overall recognition of the conceptual model's potential for the development of inclusive software if it is further incorporated into current development practices.

The second group, mostly participants who did not have Computer Science as their degree's domain, mentioned that the model could be too complex to be fully understood: *"There is a lot of data to collect and model! I am not sure how easy it would be to solicit all of that information from users in a practical application. Similarly I am not sure how it would be applied in practice (but I am not a software developer)"*.

4.6. Guide evaluation

The results were overall unanimously positive regarding both usefulness, with 100% answering yes, and ease of understanding of the guide, with 90.3% "yes" and 9.7% "no".

4.7. Threats to validity

The threats to validity of this evaluation are discussed as follows.

Internal validity. We tried to make our guide as succinct and accessible as possible. However, it still may feel overwhelming for a participant who does not have enough knowledge on gender concepts and conceptual modeling to comprehend. Nevertheless, we had no evidence of this effect on participants as they considered the guide useful and easy to read. Moreover, we did not inform participants in the illustrative guide that the conceptual model was based on a systematic mapping study, which would have been a valuable information to contextualize the reasoning for the concepts, and thus, provide participants the opportunity to give a more informed answer. Therefore, it constitutes a possible threat to the validity of the results. Furthermore, the questionnaire was divided in nine sections, which could have exhausted the participant and affected the quality of their responses. to mitigate this problem we tried to maintain a low number of questions in each section, and questions themselves were almost all multiple choice while open field questions were optional. However, the majority of participants provided answers in optional and open-ended questions, suggesting this threat may have been mitigated.

External validity. It is a risk to generalize results from such sample size, and therefore, the number of participants can raise concerns regarding the validity of the evaluation. Despite the limited number of participants, our sample had a relatively high representativeness, as our participants included bibliographic authors with experience in gender concepts, experts in conceptual modeling, and both. However, some participants may not have the knowledge to provide a comprehensive analysis of the model. For instance, a participant with knowledge on gender issues may not be able to understand the model as it was constructed in a technical language. Moreover, participants with more experience in conceptual modeling may not be aware of gender-related concepts. To mitigate this threat, we created a guide that introduced participants to the model by providing a description of each of its concepts and context of existent gender issues that served as motivation for this work. As the participants considered the guide both useful and easy to understand, we are confident that this threat has not compromised the results. Yet, we did not ask participants directly about their experience with gender-based approaches in software engineering, which poses a threat to these results. Finally, both the guide and the form were in English so we could gather the opinions of international bibliographic authors. However, some of the participants to which we send an invitation and answered the form did not have English as mother-tongue, which could have impacted their understanding of the gender concepts presented in the guide and the questions in the form. Nevertheless, the participants provided quite comprehensive English responses to the open-ended questions and considered the guide both useful and easy to understand. Thus, we found no evidence of negative impact of this decision.

Construct validity. The questions of the form pose a threat to the validity to this study due to their construction. Particularly, if the questions were constructed in such a way that they represented what they intended to ask and if they were the right questions to assess the goals of this study. Furthermore, we cannot ensure the participants interpret the questions in the way we intended to. However, we constructed the questions in an iterative, careful process that attended to the wording and sentence simplicity and objectivity with the aim of removing ambiguity. Additionally, we constructed the form in distinct and structured sections to provide participants the context of which they were responding to.

Conclusion validity. We concluded a relationship between perspectives on gender issues and difficulty of understanding the conceptual model, and importance of gender inclusion and usefulness of the model. To ensure the reliability of these conclusions, the questionnaire had a relatively high number of questions, with both quantitative and qualitative clear questions about the participants' profiles, opinions and viewpoints. Thus, we carefully collected enough and clear information from the participants to mitigate the possibility of wrongfully concluding that there is or not a relationship. However, the limited number of participants could pose a threat to the inferences made on the collected data and the conclusions of the evaluation.

5. Conclusions and future work

The work presented in this paper has its roots on a systematic mapping study we performed on gender issues in software engineering, whose findings indicate the existence of gender issues in software systems but limited approaches for addressing them in Requirements Engineering. The data collected was our basis for the development of a conceptual model for gender-inclusive requirements. The model provides a common taxonomy of gender concepts that impact software to support the discussion and analysis of such concepts in the elicitation process with the aim of formulating gender-inclusive requirements.

In this paper, we illustrate the use of the model, summarize the process created to help using the model with the definition of a set of questions and guidelines, discuss a prototype tool as a proof of concept for using the proposed conceptual model through the created process and which has been developed using the model itself, and finally, present an evaluation of the model. The GIRE prototype tool implements the GIRE process, and consists of three tasks—selection of concepts, selection of questions for each concept, and description of document details—to create a gender-inclusive requirements document. This document is intended to support software and requirements engineers in using the model's knowledge in the elicitation process.

The evaluation of the model was performed by 31 participants, including bibliographic authors and experts in conceptual modeling, regarding its usefulness, difficulty of understanding, its components completeness and adequacy, strong and weak points, and lastly, use and recommendation. The results were positive for the four components and for the complete model. Furthermore, the feedback received was very encouraging and provided valuable insights for future improvement, as well as constructive critiques and suggestions that open the way for prospective research directions.

Indeed, there is still a long road to follow to successfully develop gender-inclusive systems that benefit everyone. Thus, for future work, the conceptual model should be further improved. Despite the positive results of the evaluation, the sociocultural context component could be further refined to allow a more grounded representation of the context of development and use of software. This improvement was suggested by experts in conceptual modeling. Moreover, further validation is required to expand the performed preliminary evaluations. These should assess the validity of the results obtained as well as extend the findings with additional analysis and viewpoints.

Finally, the GIRE prototype tool should be fully implemented and an evaluation of its gender-inclusiveness would be required. Also, an evaluation of the GIRE tool and the results it produces would be necessary to assess the extent to which the knowledge of the conceptual model and the process the tool implements are useful for formulating gender-inclusive requirements that contribute to producing *de facto* gender neutral software systems.

CRediT authorship contribution statement

Inês Nunes: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Ana Moreira:** Conceptualization, Methodology, Validation, Visualization, Writing – review & editing, Supervision, Project administration, Funding acquisition. **João Araujo:** Conceptualization, Methodology, Validation, Visualization, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the reviewers for their comments, and also NOVA LINCS, Portugal (UIDB/04516/2020) for their financial support.

References

[1] E.U. EIGE, Effectiveness of institutional mechanisms for the advancement of gender equality, 2014.
[2] HCI guidelines for gender equity and inclusivity. URL https://www.morgan-klaus.com/gender-guidelines.html, visited in January 2021.
[3] U.N. Secretary-General, Special edition: Progress towards the sustainable development goals, 2019.
[4] I. Latz, et al., Equal rights for women and girls in the world's constitutions, 2014.
[5] UNESCO, Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM), 2017.
[6] R.K. Mark West, H.E. Chew, I'D Blush if I Could: Closing Gender Divides in Digital Skills Through Education, EQUALS and UNESCO, 2019.
[7] Z. Wang, et al., Competence-confidence gap: A threat to female developers' contribution on github, in: 40th IEEE/ACM ICSE-SEIS, IEEE, 2018, pp. 81–90.
[8] V. Kuechler, et al., Gender differences in early free and open source software joining process, in: IFIP Conf. on Open Source Systems, Springer, 2012, pp. 78–93.
[9] M. Burnett, et al., GenderMag: A method for evaluating software's gender inclusiveness, Interact. Comput. 28 (6) (2016) 760–787.
[10] C.C. Hsu, Comparison of gender differences in young people's blog interface preferences and designs, Displays 33 (3) (2012) 119–128.
[11] M. Rowan, J. Dehlinger, Observed gender differences in privacy concerns and behaviors of mobile device end users, Procedia CS 37 (2014) 340–347.
[12] G. Williams, Are you sure your software is gender-neutral? Interactions 21 (1) (2014) 36–39.
[13] S. Breslin, B. Wadhwa, Exploring nuanced gender perspectives within the HCI community, in: India Conf. on HCI, 2014, pp. 45–54.
[14] N. Oudshoorn, et al., Configuring the user as everybody: Gender and design cultures in information and communication technologies, Sci. Technol. Hum. Values 29 (1) (2004) 30–63.
[15] D. Ford, et al., Paradise unplugged: Identifying barriers for female participation on stack overflow, in: ACM SIGSOFT Intl. Symposium on FSE, 2016, pp. 846–857.
[16] N. Imtiaz, et al., Investigating the effects of gender bias on GitHub, in: IEEE/ACM 41st ICSE, IEEE, 2019, pp. 700–711.
[17] J.A. Rode, E.S. Poole, Putting the gender back in digital housekeeping, in: 4th Conf. on Gender & IT, 2018, pp. 79–90.
[18] R. Ibrahim, et al., Degendering games: Towards a gender-inclusive framework for games, in: IADIS Int. Conf. GET, MCCSIS, 2010, pp. 127–130.
[19] D. Metaxa-Kakavouli, et al., Gender-inclusive design: Sense of belonging and bias in web interfaces, in: HCI, 2018, pp. 1–6.
[20] I. Nunes, A Conceptual Model for Gender-Inclusive Requirements (Master's thesis), School of Sciences and Technology, NOVA University of Lisbon, 2021, URL https://tinyurl.com/37cx5b78.
[21] I. Nunes, A. Moreira, J. Araujo, Conceptual modeling of gender-inclusive requirements, in: International Conference on Conceptual Modeling, Springer, 2021, pp. 395–409.
[22] K. Petersen, et al., Systematic mapping studies in software engineering, in: EASE 2008, 2008, pp. 68–77.
[23] H. Zhang, et al., Identifying relevant studies in software engineering, Inf. Softw. Technol. 53 (6) (2011) 625–637.
[24] L. Beckwith, et al., Effectiveness of end-user debugging software features: Are there gender issues? in: SIGCHI Conf. HCI, 2005, pp. 869–878.
[25] L. Beckwith, et al., Tinkering and gender in end-user programmers' debugging, in: SIGCHI Conf. on HCI, 2006, pp. 231–240.
[26] L. Beckwith, M. Burnett, Gender: An important factor in end-user programming environments? in: VL/HCC, IEEE, 2004, pp. 107–114.
[27] M. Burnett, et al., Gender differences and programming environments: Across programming populations, in: ACM-IEEE ESEM, 2010, pp. 1–10.
[28] V. Grigoreanu, et al., Can feature design reduce the gender gap in end-user software development environments? in: VL/HCC, IEEE, 2008, pp. 149–156.
[29] T. Katerina, P. Nicolaos, Examining gender issues in perception and acceptance in web-based end-user development activities, Edu&IT 23 (3) (2018) 1175–1202.
[30] D. Liu, X. Guo, Exploring gender differences in acceptance of mobile computing devices among college students, ISeB 15 (1) (2017) 197–223.
[31] N. Muscanell, et al., Make new friends or keep the old: Gender and personality differences in social networking use, Comput. Hum. Behav. 28 (1) (2012) 107–112.
[32] K. Oyibo, et al., Gender difference in the credibility perception of mobile websites: A mixed method approach, in: UMAP, 2016, pp. 75–84.
[33] K. Tzafilkou, et al., Gender-based behavioral analysis for end-user development and the 'rules'attributes, E&IT 22 (4) (2017) 1853–1894.
[34] M. Warkentin, N. Malimage, Overcoming mixed-gender requirements misspecification with the modified coherence method, 2012.
[35] J. Winn, C. Heeter, Gaming, gender, and time: Who makes time to play? Sex Roles 61 (1–2) (2009) 1–13.
[36] G. Gao, et al., Gendered design bias: Gender differences of in-game character choice and playing style in league of legends, in: CHI, 2017, pp. 307–317.
[37] C. Hilderbrand, et al., Engineering gender-inclusivity into software: Tales from the trenches, 2019, arXiv preprint arXiv:1905.10361.

[38] M. Kay, et al., Unequal representation and gender stereotypes in image search results for occupations, in: ACM Conf. on HCI, 2015, pp. 3819–3828.
[39] S. Leavy, Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning, in: Wksh on Gender Equality in SE, 2018, pp. 14–16.
[40] B. Lenz, et al., Gender issues in the digitalized 'smart'mobility world–conceptualization and empirical findings applying a mixed methods approach, in: HCI, Springer, 2019, pp. 378–392.
[41] A. May, et al., Gender differences in participation and reward on stack overflow, Empir. Softw. Eng. 24 (4) (2019) 1997–2019.
[42] A.M.G. Ramos, T. Rojas-Rajs, Inclusion of gender perspective in design and IT environments, in: HCI, 2016, pp. 1–4.
[43] S.J. Simon, The impact of culture and gender on web sites: An empirical study, Database Adv. Inf. Syst. 32 (1) (2000) 18–37.
[44] S. Bardzell, Feminist HCI: Taking stock and outlining an agenda for design, in: SIGCHI Conf. on HCI, 2010, pp. 1301–1310.
[45] S. Brooke, "Condescending, rude, assholes": Framing gender and hostility on stack overflow, 2019.
[46] N. Marsden, et al., Developing personas, considering gender: A case study, in: 29th Australian Conf. on Computer-Human Interaction, 2017, pp. 392–396.
[47] J.A. Rode, A theoretical agenda for feminist HCI, Interact. Comput. 23 (5) (2011) 393–400.
[48] K. Spiel, et al., How to do better with gender on surveys: A guide for HCI researchers, Interactions 26 (4) (2019) 62–65.
[49] A. Schlesinger, et al., Intersectional HCI: Engaging identity through gender, race, and class, in: Conf. on HCI, 2017, pp. 5412–5427.
[50] C. Wohlin, et al., Experimentation in Software Engineering, Springer Science & Business Media, 2012.
[51] S. Stumpf, et al., Gender-inclusive HCI research and design: A conceptual review, Found. Trends® HCI 13 (1) (2019).
[52] E. Díaz, et al., Are requirements elicitation sessions influenced by participants' gender? An empirical experiment, Sci. Comput. Program. 204 (2021).
[53] A.Y. Aksekili, C.J. Stettina, Women in agile: The impact of organizational support for women's advancement on teamwork quality and performance in agile software development teams, in: LASD, Springer, 2021, pp. 3–23.
[54] M. Fernández-López, et al., Methontology: From ontological art towards ontological engineering, 1997, URL https://www.aaai.org/.
[55] N.F. Noy, et al., Ontology development 101: A guide to creating your first ontology, 2001, URL http://protege.stanford.edu/publications.
[56] T.R.D. Saputri, S.-W. Lee, Addressing sustainability in the requirements engineering process: From elicitation to functional decomposition, JSEP 32 (8) (2020).
[57] S. Buchmüller, et al., Bridging the gender and generation gap by ICT applying a participatory design process, Pers. Ubiquitous Comput. 15 (7) (2011) 743.

Inês Nunes completed her M.Sc. in Computer Science in 2021 from NOVA University of Lisbon, School of Science and Technology.

Ana Moreira is an Associate Professor at Universidade Nova de Lisboa where she leads the Software Engineering group. Currently, her main research topics are requirements engineering, architecture design, model-driven development, software quality, variability and trade-off analysis. She is a member of several editorial boards (TSE, SoSyM, TAOSD). She has been or still is a member of the Steering Committee for UML/MODELS, RE, AOSD and several regional conferences. She has co-organized various international workshops and is co-founder of the international movements pUML and Early Aspects. She was Conference Chair and Program Committee Chair of several international events, the Foundations Track Chair for MODELS 2013, and co-General Chair for RE 2017.

João Araújo is an Associate Professor of the Department of Informatics at the Universidade Nova de Lisboa. His principal research interests are Requirements Engineering, Model-driven Engineering, and Quality of Requirements Models. He has been a co-founder of the series of the Early Aspects workshops held at AOSD, OOPSLA, SPLC, and ICSE. Additionally, he served on the organization committees of MoDELS, RE, ECOOP, SAC, AOSD, and ICSE in the past few years. He was the co-General Chair of RE'17.