

# Chapter 7

## Equity & Inclusion



When beginning our research on equity, diversity, and inclusion under the umbrella of FemTech.dk research, we engaged with new literature, theory, and analytical approaches from research on equity and inclusion – research we did not know prior to FemTech.dk but which has been fundamental to our activities. In this chapter, we introduce the theoretical vocabulary we have learned as we entered this research space. Our purpose is to provide a short introduction to the most important concepts we found essential and relevant for our purpose of exploring diversity in computer science and to give readers a quick introduction to the most important concepts, which they then can use to initiate equity work in their institutions.

However, we encourage readers who want to expand their knowledge to dive into some of the foundational literature on equity in order to gain much more detailed insights on the complexity of historic structures that challenge equity today (Crenshaw 1988, 1989; Haraway 1990, 1991; Butler 1999; Ensmenger 2010; Ahmed 2012, 2016, Benjamin 2016, 2019a, b; Hicks 2017).

But before we discuss all the important concepts, we begin with a fundamental discussion of technology design which we brought to FemTech.dk initially. Namely, the discussion on politics in technologies – since this argument demonstrates why equity and inclusion in computer science are critically important for democratic societies, and why we urgently need to take action.

### **Technologies Have Embedded Politics, and It's a Technical Problem**

“Artefacts have politics” is a much quoted phrase from Landon Winner’s famous paper (Winner 1986) where he uses the height of a bridge to demonstrate how access to a beach can be controlled by preventing public buses (thus, people who do not have their own car) from traveling under it en route to the beach. Bringing the

argument to classification schemes, Lucy Suchman (Suchman 1994) began an important debate about whether such schemes and categories bring in politics when designed into digital systems. Engaging in this debate, Wanda Orlikowski (Orlikowski 1995) demonstrated how classification schemes and categories developed under apartheid in South Africa were very much used to restrict and constrain particular people while enabling other groups in their efforts. Clearly, technologies are not apolitical artefacts; they bring certain values embedded in the kinds of classification schemes that serve as the infrastructure in these systems (Bjørn and Balka 2007). This political nature of technology requires us to pay attention to several questions: Who benefits from the use of technologies, and how? Who are the designers and creators of technology? Where are the designers and creators of technology located in the world, and how do their perspectives, frames of reference, and/or privileges shape the technologies that are built?

So why is this a technical problem? Let's unpack that through the example of designing a dental appointment scheduling IT system for a pediatric dental clinic in Denmark. In Denmark, pediatric dental clinics are part of primary schools, and patients are scheduled for regular visits by the clinic secretary, which includes informing a child's parents of the appointment date and time. So, unlike adult dental systems, appointments are initiated by the clinic – not the patient – and patients and parents are informed of appointments.

Before 2001, all invitation letters from the school dental clinic were sent by physical mail to the child's address. However, in 2001 e-Boks (governmental digital mail) was implemented in Denmark, and in 2010 NemID (unique citizen login to all official IT systems such as tax IT-systems, school IT-systems, banking IT-systems, pension IT-systems, healthcare IT-systems etc.) was implemented. These two IT systems together allow all Danish citizens to receive and communicate digitally with public and private entities. While these systems indeed decreased the resources otherwise spent on envelopes and stamps, they also introduced new challenges and considerations about societal classification systems and their impact on the user interface, on the algorithms, and on the database models and tables embedded within such technical systems. Let's dive into the problem.

Prior to the implementation of e-Boks and NemID, all dental invitation letters were sent to the mother of a child in a physical envelope. The letter was easy to share within the household, e.g., by hanging it on the fridge or a shared pinboard – or simply by handing it to household members. However, when the invitations to dental appointments became digitalized using the same classification scheme and 'algorithm' as in the paper-based system, invitations were limited to mothers' e-Boks and were no longer easily shared among household members.

Figure 7.1 presents a simplified database diagram of the pediatric dental appointments IT system, demonstrating how appointments are performed at specific clinics, scheduled by the secretary, and invitations sent to patients. Further, the patient entity includes a number of fields such as mother's name, father's name, and address. The database structure is based on the assumption that patients' households comprise these variables. Now algorithms for sending the invitation use the data in the database but also include assumptions about the household, namely that it is the

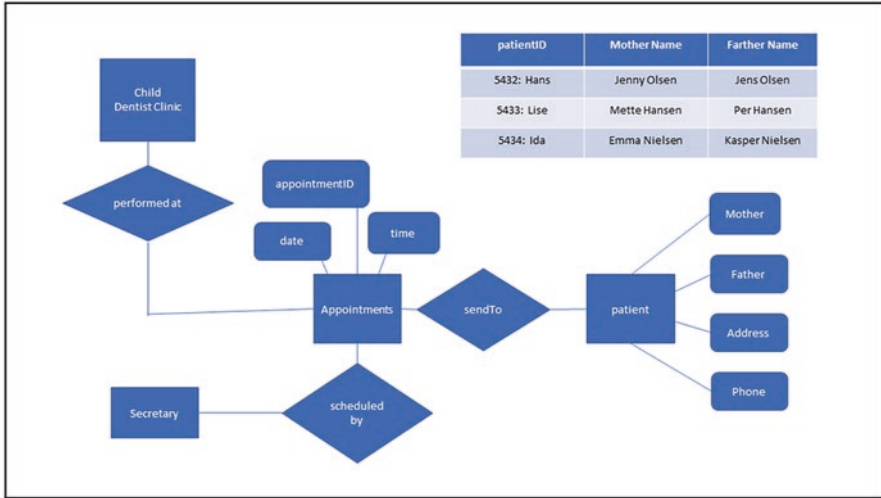


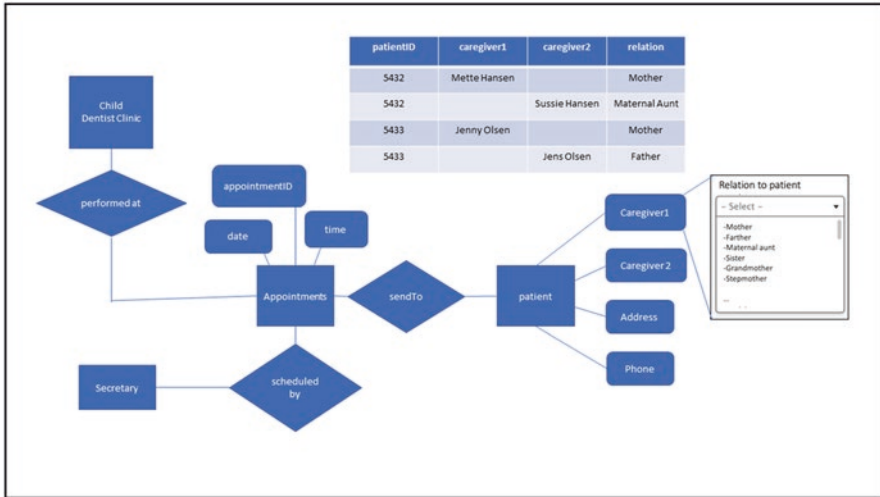
Fig. 7.1 Database model for pediatric dental appointment illustration no. 1 on how the politics of classifications and categories is a technical problem

mother who is responsible for children’s dental appointments. This means that the invitation is sent to the mother. However, when the name and address is not just physically printed on paper but is instead used to direct which digital e-Boks the invitation should be addressed to, it limits who has access to the information about dental appointments in the household, which again limits the household’s agency in deciding how to organize the tasks.

As concretely experienced by the first author of this book, the change to e-Boks meant that her children missed their dental appointments because in her household she would receive the digital appointments, but in her family, it is her husband who takes on the task of ensuring that their children do not miss dental appointments. For this example, a simple change could be to ‘re-design’ the algorithm for sending notifications of dental appointments to include both mother and father. However, it is not enough to simply change the algorithm since the underlying database continues to create problems.

If we have a case where a mother chooses to raise her child with her own sister, and thus the three are living together and both the child’s mother and maternal aunt need to be informed of dental appointments, then the secretary can choose to circumvent the system, using it in a different way than intended by the designers, and simply enter the name of the maternal aunt in the field reserved for the father. The algorithm would remain the same and the secretary would achieve the task of informing both mother and aunt of dental appointments (Fig. 7.2).

However, this would be an incorrect use. Instead, a re-design (see Fig. 7.2) of the underlying database and categories would be more appropriate. Here the issue could be resolved by replacing the categories of ‘mother’ and ‘father’ with caregiver1 and



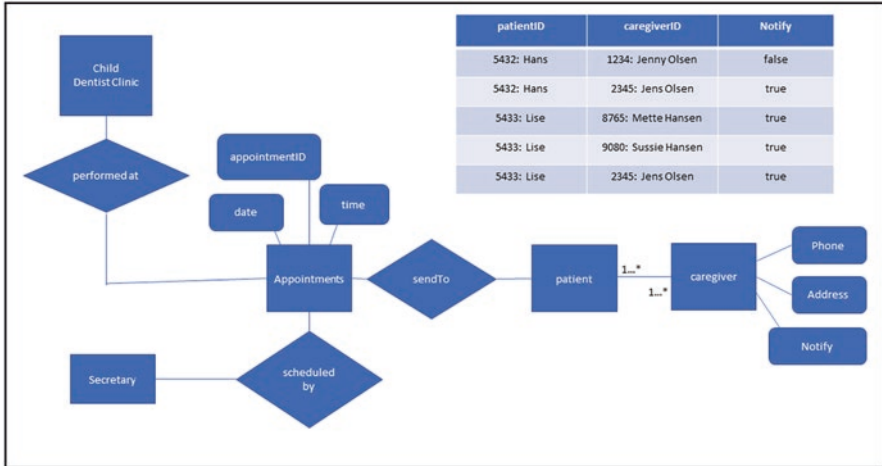
**Fig. 7.2** Dental appointment database system, where ‘mother’ and ‘father’ are replaced by the category of ‘caregiver’; illustration no. 2 of how the politics of classifications and categories is a technical problem

caregiver 2 in the database and potentially adding a dropdown menu to the user interface allowing the secretary to indicate the relation between child and caregiver.

Adding the ‘relation’ dropdown menu means implementing a new classification scheme for this technical design. It is evident that the classification scheme embedded in the dropdown menu for potential types of relations is based on certain assumptions about which relationships a pediatric dental patient can have. Such categories and classifications are based on assumptions embedded in society. In designing such a dropdown menu, the IT developer needs to consider the completeness of categories capturing a pediatric dental patient’s relations. Such technical decisions thus require careful examination of the completeness of the classification, including that those dental patients might have two mothers or two fathers, or that patients in rainbow families have multiple fathers and mothers who together are responsible for the child’s dental appointments.

Complicating the matter further in exploring the family structures in Denmark, it is not uncommon to have divorced families, where previous partners find new partners – and where families comprise ‘mine’, ‘your’, and ‘our’ children. Considering re-designing the pediatric dental appointment system based on the 2022 family structures in Denmark, we in Fig. 7.3 show how the database design with multiple relations between patients and caregivers could look like.

The database design in Fig. 7.3 is based on the fundamental assumption that patients can have one or more caregivers (1…\* ) and that each caregiver can have one or more relations to multiple patients (1…\* ). All schemes for classifying these relations have been removed from the database design. Instead, the algorithm for whom to inform about dental appointments is re-designed and is now based on a new



**Fig. 7.3** Dental appointment database, where each child can have multiple caregivers and each caregiver can have multiple children, and where the algorithm is re-designed to use a Boolean notification feature (true/false) rather than the classification of caregiver; illustration no. 3 of how the politics of classifications and categories is a technical problem

variable, 'Notify', which is also included in the user interface, allowing the secretary to indicate whom to notify. This design imposes no limitations on different family structures, as the database 'table' in Fig. 7.3 demonstrates that Jenny Olsen and Jens Olsen have Hans as their child; when Hans has a dental appointment, Jens Olson need to be notified. Further, Lise is living with her mother, Mette Hansen, and her maternal aunt, Sussie Hansen, who both need to be notified of dental appointments. What we also see is that Jens Olson also has a relationship to Lise (his daughter from a previous marriage), and he also needs to be notified of dental appointments for Lise.

Our point here is not to provide a step-by-step introduction to database design but to demonstrate that each time an IT developer makes a technical decision and implements categories and classification systems, it matters for how the user interface, the algorithms, and the database structures are created and implemented. Decisions about database design structures matter for how to design appropriate algorithms which can search, relate, and manipulate the data. Decisions about algorithmic design matter for which kinds of data manipulations and visualizations can be produced to connect back to the database design. Finally, decisions on user interfaces also impact database design, since new features or variables implemented in the user interface need to be accommodated in the database structure. It is not enough to change, e.g., the gender classification scheme in a banking system from binary 'woman' and 'man' to include 'non-binary', 'other', and 'prefer not to say' if the fundamental statistics and visualizations continue to only report on binary data.

Clearly the above example is simplistic, and many IT systems are much more complex, embedding multiple and related databases which are not so easily

examined and changed. One example is IT job portals. IT job portals often implement different types of predictive natural language processing technologies such as Microsoft’s Word2vec to train the recommender functionality in the portal. Such technologies are trained by using different neural network models to learn word association in large datasets and uses the learned relations to predict and match people with jobs. When training algorithms based on historic data on job relations, the algorithms will learn the historic bias in jobs, such as that women historically do not hold top management positions whereas men historically do. This means that the historic bias will persist in newly implemented prediction systems unless IT developers and designers find ways to circumvent and balance bias considering not the past as a predictor of the future but how we want the future to be a predictor of the future.

In Fig. 7.4 we demonstrate the risk of bias embedded within algorithms based on a historically biased dataset. The example shows a LinkedIn message received by the first author on November 5, 2018, when she had been a full professor in the Department of Computer Science for 3 years. The message suggests that a top job pick was ‘Easy Online Part-Time Job’ as a ‘Web Search Evaluator’, which, if anyone should be in doubt, she was overqualified for. LinkedIn is not alone; several other recruitment tools at large IT companies such as Amazon have been trained to vet applications by learning patterns in historic resumes (Dastin 2018), clearly re-introducing bias from the past to the future.

Other large IT systems where the classification schemes and categories enable or constrain certain populations include IT systems for insurance, in immigration, in job centers, or in hospitals (Bjørn and Balka 2007; Boulus-Rødje 2018; Møller et al. 2019, 2021a, b; Asbjørn Ammitzbøll Flügge and Naja Holten Møller 2021; Petersen et al. 2021). For example, the classification schemes embedded in private insurance policies in Denmark have systematically let to mistreatment of pregnant women (Hall 2020). All insurance companies are highly digitalized in Denmark, which means that these policies have been enforced through IT systems, and thus that

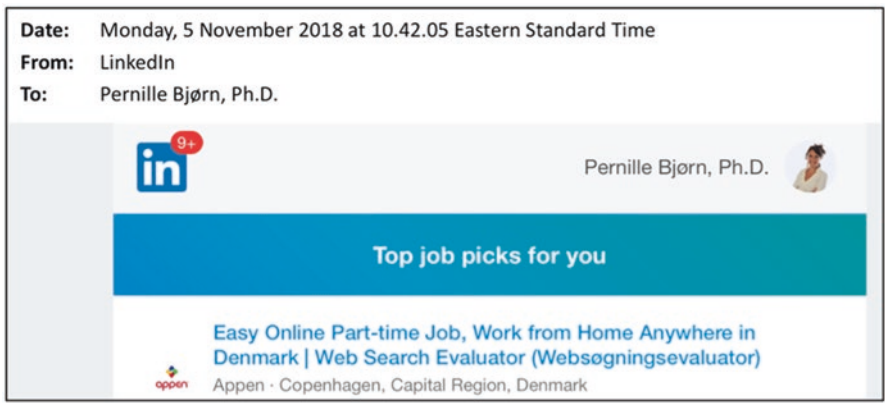


Fig. 7.4 Demonstrating bias in LinkedIn prediction of job

changing the behavior – following the law – requires re-designing the IT systems – databases, algorithms, and user interfaces.

As we showed above, the categories embedded in IT systems risk introducing problematic classification schemes which constrain certain populations. While this might not have been the IT developer's intention, it does not change the fact that when software designers, IT developers, and computer scientists – maybe unintentionally – design systems with problematic categories, it has real impacts on the lives of real people.

When computer scientists build a social media application for people to rent each other's homes (e.g., Airbnb) or develop a personal driving service where people drive others around in their own cars (e.g., Uber) (Sachs 2015; Kircher 2017), they use their own experiences of living and working in San Francisco's Silicon Valley as more generally applicable to other parts of the world. However, they tend to forget that the world is not the same everywhere, and that the conditions for travel or renting out houses vary.

To illustrate this point, let's look at an example from research conducted over several years by Nina Boulus-Rødje and the first author that explores challenges faced by tech entrepreneurs in Palestine. When we create technologies, these are socially situated within certain translocal infrastructures (Bjørn et al. 2017). You cannot import the technological concepts from Silicon Valley to Ramallah in the West Bank and expect success (Bjørn and Boulus-Rødje 2018). Tech entrepreneurs cannot simply adapt Western concepts to a land of occupation (Boulus-Rødje and Bjørn 2021). If the problem of getting parcels is not about local transportation and drop-off boxes but fundamentally about border control and harassment, there is no technological fix. Further, technology developed locally within Palestine cannot simply transcend the separation wall and reach the outside world (Boulus-Rødje et al. 2015; Boulus-Rødje and Bjørn 2019) if global technological infrastructures such as Apple's App Store or global payment gateways are inaccessible (Bjørn and Boulus-Rødje 2018).

Technology intersects with societal constructs such as workers' rights (Bødker et al. 1988; Kensing and Blomberg 1998) and through such encounters is transformed while transforming society. Tech entrepreneurs do not merely provide technological platforms allowing others to participate in the sharing economy. Instead, they risk unintentionally building an infrastructure that eliminates workers' rights (since they are not employees) and reinforces hidden structural racism in who gets to rent what kinds of houses (since landlords can choose tenants without justification) (Martin et al. 2014). For example, research found that prospective Airbnb guests with African American-sounding names are 16% less likely to be accepted than guests with White-sounding names (Edelman et al. 2017) and that facial recognition software does not work correctly on darker skin tones, introducing discrimination by design into Uber applications (Sachs 2015; Barry 2021).

Technologies have politics (Suchman 2003) – intentional as well as unintentional – and it is urgently important that we train new computer scientists to take their share of the responsibility for identifying and taking actions to reduce the risks of constraints for certain users embedded in the design. If problematic



classifications are embedded in technology producing biased interfaces, biased database systems, or biased algorithms, it is vital that technology developers be trained to analyze and discover such problems – allowing them to correct the problems or perhaps prevent them in the first place.

A good way to begin is to ensure a diverse workforce: a diverse group of tech developers and designers. This agenda is increasingly gaining traction in the industry and in education. But we also need to educate and empower tech developers, including computer scientists, to prevent the creation of bias and barriers that can act as exclusion mechanisms. Positive change needs to be embraced at multiple levels in the computing ecosystem, besides the mere introduction of “diverse” teams. We argue for the need to introduce structural changes in both tech education and the tech industry. Such changes take time and effort, and we suggest beginning with including critical approaches to computing and accessibility into the core computer science curriculum, as researchers in computing education have been increasingly advocating for (Ko et al. 2020). Further we must ensure that organizations do not just engage in what former Google researcher Timnit Gebru referred to in an interview as “diversity theatre” (Preston 2021), when diversity commitments fail to truly empower and support the work of (often under-represented social groups) in the areas of bias prevention and equity.

Note that the implied user is often built in the image of the designer or developer, since people act and develop based on what they know and experience. However, users of technology are multiple and diverse simultaneously; if you develop a technology for a pharmacy in Danish society, you might have homogeneity across pharmacies. However, while each pharmacy generally follows the same procedures, contextual contingencies will always exist and must be considered in technology design (Bjørn et al. 2009). Thus, if we move the pharmacy technology to the Philippines, the main purpose of the pharmacy remains the same, but the procedures might differ vastly, resulting in different use of language and vocabulary within the IT system (Jensen and Bjørn 2012). Developing technology for the global market clearly requires us to work in teams with diverse perspectives and backgrounds to ensure that we consider the potential barriers that we risk building into our systems, allowing us to take steps to make them more inclusive – or at least not exclusive by design (Møller et al. 2017; Møller 2018; Matthiesen et al. 2020, 2022).

The FemTech agenda sees the responsibility for addressing the risk of embedded bias in system design as collective: empowering change begins in education, within computer science programs, enabling graduating computer scientists to think critically and intentionally about the power of classification schemes and the impact of bias built into technologies, and thus to make better choices when designing and maintaining digital interfaces and infrastructures.



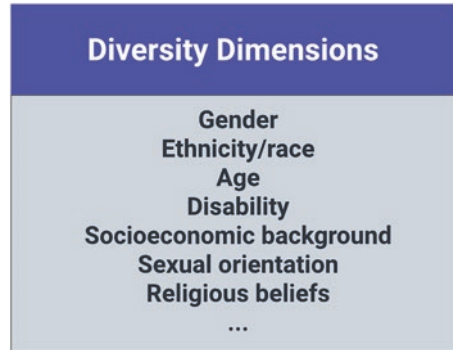
## Diversity Dimensions and Equity Classification

Studying equity and inclusion in computer science from an interventionist perspective provides different challenges and opportunities for our endeavors. Our role as insiders (being computer scientists) meant that we had prior relevant knowledge about practices (e.g., programming), vocabulary (e.g., nerd culture), and artefacts (e.g., micro-processors). However, from ethnographic approaches, we must acknowledge that being an insider does not make you an expert in studying your own field (Forsythe 1993, 1999, 2001). Basic assumptions about the field risk hiding important aspects that need scrutinizing. Important aspects of the practices risk being invisible to insiders, who simply take such aspects for granted rather than question their very existence through examination. Further, as researchers, we knew that important knowledge about gender, equity, and inclusion already existed in research literature. Thus, as with any other new domain, we needed to become familiar with the core concepts and research vocabulary – not just within the field of computer science (empirical focus) but also within existing research on gender and equity (theoretical focus). We needed theoretical concepts to help us question fundamental assumptions within the empirical field to make them noticeable and visibly available for scrutinizing. Thus, an important part of FemTech research is to establish a theoretical reasoning that can help us explore our own blind spots as insiders in the field.

In this section we want the reader to reflect on how social inequalities in tech manifest in relation to specific social markers – we call this a *diversity classification scheme*. The scheme is not exhaustive, and it is meant to be open; each context would call for additional dimensions. We also want the reader to reflect on the different areas in which social inequalities in technology can manifest. A diversity classification scheme is not only about gender but fundamentally about all different kinds of ways that people can be unique while still being part of a larger community – and the different ways in which social inequality can manifest in relation to these differences such as sexism, racism, and ableism, just to name a few. A diversity classification scheme is not a checklist but an incomplete list of aspects that technology designers and developers need to consider and critically reflect on to understand how the dynamics of social inequalities manifest in relation to forms of human diversity shaped by technology.

There are four main areas where social inequalities can manifest in technology design and development: (1) user interfaces, (2) databases & data structure, (3) algorithms, and (4) team composition & power dynamics. The first three areas concern the technical design and link back to the prior section: “Technologies have embedded politics and it’s a technical problem”, where we unpack how diversity and equity is a technical problem. The risk of social inequality embedded in user interface design, database design, and algorithms often considers ‘an omnipotent user’ stripped of all social markers for the design. However, the world is full of diverse people, and when engaging with biased technology, people ‘who do not fit the characteristics of the omnipotent user’ will be constrained in their interactions

**Fig. 7.5** Diversity dimensions & social identifiers – an incomplete list



with the technology. The fourth area is then related to the actual design and development process of technology, considering ‘who’ belongs to the group of people developing technologies as well as the hierarchy, decision power, and power dynamics. All four areas are important if we want to create unique, innovative, and relevant technologies and be mindful of bias and how biases can manifest in technologies, along different social markers – in the design, testing, maintenance, and use of technology. The diversity classification scheme is relevant for all four areas.

Fundamentally, there are infinite diversity dimensions that are relevant for technology design – including gender, race/ethnicity, disability, age, sexuality, religion, and socioeconomic background – depending on which kinds of individual projects and technologies are being designed. Technologies are used by everyone; thus, technologies should be able to express and consider all kinds of diversity dimensions (Fig. 7.5).

*Gender as a diversity dimension for technology* has broadly received much attention within computer science research in the last couple of years (Breslin and Wadhwa 2014; Hicks 2017; Buolamwini and Gebru 2018; Frieze and Quesenberry 2019; D’Ignazio and Klein 2020; Albusays et al. 2021). When we created FemTech in 2016, a gender lens was our focus and has guided much of our work throughout all the design artefacts presented in this book. We originally conformed to a cis-gender binary framework, influenced by our institution’s focus on “attracting more women” to computing, rather than being intentionally inclusive of trans\*, intersex, and gender-non-conforming people. In current editions, we ensure that expansive gender language is used, intentionally targeting people from all under-represented gender identities. However, from the beginning we strived to create not gendered artefacts but artefacts that were as gender-neutral as possible. We were not successful in this with Cyberbear, but we did succeed in this with GRACE and Cryptosphere.

Moving forward with initiatives, events, and strategies, we consider gender as a non-binary dimension that includes trans\* and other gender-non-conforming identities. Gender is a social construct, shaped by social norms, and different societies have different gender norms that again affect people differently. Since our work focuses on gender diversity in computing in a Scandinavian country, these are the gender norms we engaged in our design artefacts. However, we are well aware that

different gender norms exist, and that other initiatives in other countries should identify, address, and challenge the gender norms shaping computer science in those countries.

*Ethnicity/race as a diversity dimension for technology* was put on the agenda in 2020 with the increased mobilization of the Black Lives Matter movement in the USA and in human computer interaction research (Ogbonnaya-Ogburu et al. 2020). Understanding ethnicity and race from a global perspective is difficult. The concept of race is foundational to systemic repression in, for example, the USA (Noble 2018), and racial classifications are social constructions first and foremost developed and performed through historic situations of slavery and colonization (Benjamin 2019a, b). Racial and ethnic classifications are constructed differently in different contexts, and social inequalities that manifest in relation to them are unfortunately pervasive: technology is increasingly scrutinized as one of the areas in which racism and discrimination are embedded, with harmful social impacts. Racism also exists in Scandinavia, including Denmark. Thus, it is important for the design of technologies to consider the ways in which ethnicity produces social inequality to ensure that problematic societal markers are not being reproduced and potentially re-enforced through IT system designs. Ethnicity is an important diversity dimension to include in equity interventions and needs to be situated within the specific societal context considered for a technology design.

Being honest about our own work, we initially in 2016 did not address ethnicity as a diversity dimension explicitly in our design artefacts; however, we did have a strong focus on including participants from a wide range of socioeconomic backgrounds, and we focused on reaching out to non-Danes, ensuring that all our events were in English. When we recruited participants for the FemTech workshops, we initially explicitly reached out to high schools in the lower socioeconomic areas of Copenhagen, which also provided us a diverse group of participants in terms of ethnicity.

*Age as a diversity dimension for technology* includes considerations for how to address various aspects of the growing elderly population for technology design (Tellioğlu et al. 2014; Hornung et al. 2017) as well as for children and youth (Boyd 2007; Thyssen 2015; Pinkard et al. 2017). We need to consider the digital divide between digital natives growing up with the internet and having different advantages for technology use and older adults for whom internet access is not necessarily a main part of their lives. Age in technology development is also related to privacy and security, such as considerations of who has access to which types of data under which conditions (e.g., parents' access to children's data) and of when people are considered adults, which varies between societies. Age can also be considered in terms of experience, as a number, or in terms of bodily decay. How we understand age depends on the living conditions of a specific geographical location. We have not directly addressed age in our design of artefacts and events. However, while most participants in our FemTech workshops were teenagers, a few were in their twenties. These were people who for different reasons had moved to Denmark from abroad (as refugees or immigrants) and thus began high school later. At the GRACE events, we had mixed age groups; at the Danish event, participants ranged from

primary school children to retirees. At the two conference events, we estimated the participants to be 22–60 years old.

*Disability as a diversity dimension for technology* includes considerations of both mental and physical health. Disability studies in computer science is an ongoing attempt to include voices related to personal and social experiences of disability in the academic field (Spiel et al. 2020). Mental and physical health can shape people's experiences and access to technologies (e.g., blind software developers use screen readers for programming (Potluri et al. 2018)). However, instead of viewing disability as primarily the loss of a function in an individual (the so-called medical model of disability), contemporary research stresses how disability arises in the interaction between functional limitations and impairments and social and physical barriers (social model of disability). Disability is used also as an analytical lens to identify problems that can be a vehicle for developing new areas for research. As stated by Jennifer Mankoff, Gillian Hayes, and Devva Kasnitz: “A *better understanding of what constitutes a problem from a disability studies perspective can help to enrich existing research and illuminate new areas of inquiry*” (Mankoff et al. 2010, p. 3). Thus, critically addressing and understanding how people with a disability (temporary, permanent, or situational) face barriers in sociotechnical spaces also offers an opportunity to drive the field of technology design forward for all. In FemTech, we have not addressed disability in our published work, but current FemTech work by the third author of this book is pushing the agenda further, considering how we can bring in disability and accessibility as part of FemTech.

*Socioeconomic background as a diversity dimension for technology* places the focus on the socioeconomic backgrounds of people and places, and how such aspects shape people's access to or inaccessibility to engage with technology. Often technology is articulated as the driver of making the world equally accessible to everybody – and how, for example, digital platforms remove physical borders in a globalized world. However, barriers remain. Classist algorithms using healthcare spending as a proxy for healthcare needs or using collected health data on wearable devices to determine health insurance costs perpetuate inequalities (Christophersen et al. 2015; Vartan 2019). Bias manifesting along nationalities and geographical contexts, rooted in colonialism, are still prevalent. There are distinct differences between working as a software developer in the Global South versus the Global North (Bjørn 2019). Where you are located matters for your translocal contingencies (Bjørn et al. 2017), infrastructural accessibility (Bjørn and Boulus-Rødje 2018), or implicit bias (Matthiesen et al. 2022) and shapes the global encounters mediated by technology in different ways. By paying attention to socioeconomic and geopolitical conditions (e.g., for refugees (Stickel et al. 2015)), when we explore and design technology, we will notice the nature of the taken-for-granted assumptions about sociotechnical infrastructures that serve as the foundation for contemporary technology development. This will allow us to challenge the status quo and begin creating inclusive and diverse technology development practices, which are accessible for a larger global group.

*Sexual orientation and religious beliefs are diversity dimensions relevant for technology* in considering both the classification schemes we embed in the applications (Abid et al. 2021) and how people's personal beliefs or sexual orientation are

important areas for technology innovation (Mustafa et al. 2020). As with other diversity dimensions, sexual orientation and religious beliefs open the design space for technology development. Muslim prayer practices were the driving force for adding a digital compass to smartphones, now standard in most phones; and dating apps are examples of the importance of diversity in sexual orientation and religious beliefs for the analytical design perspectives of technology (Hariri et al. 2021). However, additional considerations are important. In the rise of social media, we have also witnessed a new type of situation where sexual orientation and religious beliefs have driven online harassment in anonymous fora (Rubin et al. 2020) and in the workplace (Tenorio and Bjørn 2019) (Fig. 7.5).

As we work to create inclusive environments, we need to consider the different diversity dimensions and acknowledge that diversity is not always something you can ‘see’. You cannot immediately see who people are, where they come from, or which ‘characteristics’ contribute to making them who they are. In creating an inclusive environment, whether for computer science education, software development work, or any other aspect of society where digital technologies are used, we must consider that people are different and assume that the people we design for are different from ourselves. We cannot rely on our own experiences and bodies as a template for others. The unconscious process by which designers configure users as fundamentally resembling themselves is defined as “I-methodology” (Akrich 1995), and this implicit representation process presents clear constraints even for user-centered design practices (Oudshoorn et al. 2004). Software developers and designers must learn as part of their education to be mindful and aware of the biases that can occur in design processes and in the application of technology to different sociocultural contexts. Being aware of the multiple intersecting diversity dimensions, and of how they can affect the design of interfaces, databases, and algorithms, is necessary to actively get an edge in our digital innovations. By designing while keeping in mind the rich variety of social identities, we improve technology for all people instead of just a few (who typically resemble the individuals who make up technology design teams). In having a diverse and inclusive workplace that considers the rich variety of human difference and that is mindful of the social dynamics that manifest in relation to diversity, we have direct access to noticing and identifying the otherwise invisible exclusive mechanisms in our technologies – which can give tech companies a competitive advantage over other software company competitors. Software developers and computer scientists in their education will benefit greatly from learning about and experiencing working actively with diversity dimensions, connected social identities, and related mechanisms of bias and discrimination, enabling them to use these insights when developing digital technologies. It has always been fundamental to computer science education and software development practice to work together in teams and with people from different professions. When designing IT system for pharmacies, software developers need to be able to talk with pharmacists, and when designing IT systems for healthcare practitioners, they need to be able to talk to doctors and nurses. Thus, the skills required to engage with other professions with the aim of designing technologies are part of the core curriculum of computer science. Collaborating and communicating are

fundamental skills and expertise that are critically important to designing technology for people and society.

We argue in this book for extending the existing perspective on user-centered design and including teaching and learning about diversity dimensions in technology development as core and fundamental skills and expertise for two reasons. First, because paying attention to diversity dimensions connected to social identities opens the field of computer science in terms of who belongs and can succeed in the field; second, because diversity dimensions can be used strategically in technology design to reveal spaces for new innovations, technologies, and practices shaping a just and fair society of tomorrow.

## Equity and Intersectionality

The diversity dimensions introduced above are important as individual dimensions, and together they benefit technology research and innovation by extending the analytical and design agendas in novel directions. However, rather than use these dimensions as a mere checklist for innovation, we must pay attention to the historic conditions that created unbalanced participation in the first place. The dynamics of social inequality have historically manifested in relation to social identities (gender, race/ethnicity, age, etc.), having a concrete impact on the starting point for individuals' actions. We must consider the history that produced certain unequal situations in society in general to understand the unbalanced diversity in computer science. "[E]qual process (...) make[s] no sense at all in a society in which identifiable groups had actually been treated differently historically and in which the effects of this difference in treatment continued into the present" (Crenshaw 1988, p. 1345). Different societies have different historical backgrounds; thus, comprehending how the different diversity dimensions are shaped historically requires insights into the historically situated conditions. The practice of ensuring diversity and inclusion is not a process of equal access for all, since the conditions for people to participate at the outset are not equal.

Moving into the situated historical conditions for computer science in Denmark, introduced in the beginning of this book, we need to pay attention to social inequality as it manifests in the field, indicated by the numbers of women and other gender-minority faculty in the computer science department; the so-called Matthaeus effect for distributing grants, indicating a self-reinforcing mechanism whereby already successful researchers keep getting funded; and the statistics for the privilege of supervising PhD students (see Chap. 1). While some women have succeeded as computer scientists and received national and international recognition, only very recently (since 2016) can we detect an improvement in numbers in Denmark. To understand the current situation, we need to revisit the history of computer science.

Historically, computer science as a field and domain emerged during WWII, as men historically were recruited to the military as soldiers while women worked on measuring missile trajectories or breaking communication codes (Ensmenger 2010;



Hicks 2017). The term “software engineering” was coined by a woman, Margaret Hamilton, and the first computer bug was found by another woman, Grace Hopper. Katherine Johnson, Dorothy Vaughn, and Mary Jackson worked at NASA as ‘computers’, where they made the calculations allowing for space travel. Software was woven by threaded copper wires into the core rope memory for the Apollo moon landing by women working as Raytheon’s expert seamstresses, nicknamed ‘Little Old Ladies’ (Rosner et al. 2018a, b). The ENIAC women were the first to program a general purpose computer (Ensmenger 2010), and Jean Valentine, Joan Clarke, Margaret Rock, Mavis Lever, and Ruth Briggs all worked to break Nazi Germany’s Enigma code at Bletchley Park. Computer science and programming began as a women’s occupation in the USA and UK.

During WWII, Denmark was occupied by Germany and thus was not part of developing the field of computing via military endeavors. This meant that computing did not arrive in Denmark until after the war, and here computing began in industry (Sveinsdottir and Frøkjær 1988). We know little about the work of the early women in the Danish computing industry, since it is not well documented; however, in a few places women are mentioned as ‘hulkort damer’ (punch-card ladies). When computer science became an academic field in 1970, it was during the student rebellion in which universities in Denmark changed from being controlled by professors (the vast majority being men) to allow equal representation for student and staff on different committees. There were women when computer science was first created; however, only one woman, Edda Sveinsdottir, is mentioned by name in the written history (DIKU 2021). There are no gender statistics available from the University of Copenhagen until 1997; however, that year there were 18 women out of 241 students (7.47%). The years with the lowest numbers of women students were 2004 (3.66%) and 2011 (3.9%), when their share was below 4% (Forskningsministeriet 2021). These low percentages are surprising given that Denmark is known for its high ranking for equality; however, in recent years Denmark has not been among the top 10 countries on the equality index, and even our Nordic neighbors Iceland, Norway, Finland, and Sweden occupy the top 4 positions (Forum 2020).

Birgitte Possing, a professor of history and women in Denmark, tries in her book to unpack some of the conditions explaining historical gender inequality in Denmark (Possing 2018). Referring to professor of law Hanne Petersen, she suggests that in the ’80s and ’90s there was a marriage between two different political movements in Denmark. On one hand was the historical embedded cooperative movement (“andelstanken”) stipulating that all are equal, and which has been strong in Denmark since the 1700s. On the other hand, a new liberal thinking was introduced in the late ’90s, often referred to by the slogan “du er din egen lykkes smed”, which can be understood as a Danish version of the American “dream”, meaning that you are responsible for your own success and that if you fail, it’s your own fault. Thus, responsibility for equal conditions in Denmark was left to the individual, and formal organizations responsible for ensuring equality were shut down in 2000 (Possing 2018). Possing proposes that one explanation for Denmark’s lack of gender equity is that when society combines the cooperative idea that everybody is equal with that of individual responsibility for ensuring equal access, any analysis of or pointing to



problematical existing structures with unequal conditions becomes an individual concern rather than a collective responsibility.

The very idea and understanding that there are fundamental conditions embedded in society causing some people to have privilege and better conditions for success than others – and that these conditions are based on people’s gender, ethnicity, disability, or socioeconomic conditions – must be acknowledged as the starting point before new initiatives to make change can have long-term impact.

Following Possing’s argument, as part of a process towards making computer science diverse and inclusive, we must consider the historically unequal conditions in academia based on gender, ethnicity, disability, or socioeconomic background. We need to pay attention to the people who are under- or unrepresented within the field and find ways to mobilize and encourage their efforts in joining and using the opportunities that digital skills and expertise bring for social mobility in the society. We must find ways to allow under-represented groups in computer science to enter and shape the field in their own ways, creating new agendas for technology design and use. It is not about getting people who are currently not included to fit into existing schemes stipulating the nature of computer science and computer scientists. Instead, the approach we argue for in this book is to open the field and allow newcomers from diverse backgrounds to shape and transform the field to their interests and perspectives, and to recognize that we all have responsibility for collective, structural change in order to empower new perspectives and new efforts that push against normative frameworks. Encouraging diversity in computer science is *not about equality, it is about equity*.

*Equity* is a concern directed at balancing the support, encouragement, cost, and so on, in relation to the benefit, reward, outcome, et cetera, of an activity, taking into account individual conditions. Thus, equity is fundamentally about the fair distribution of resources based on actual need, which requires us to be better equipped to critically assess whose needs have been overlooked and which groups are more likely to incur negative social outcomes due to bias and discrimination. This means that making change is not about providing equal opportunity for all but about identifying who is excluded and focusing our interventions there. Further, making interventions towards equity is not an individual responsibility but a collective responsibility directed at providing and improving the conditions for equity.

So, what does collective group responsibility really mean? Ogbonnaya-Ogburu, Smith, To, and Toyama provide an excellent example of this in their 2020 paper on critical race theory (Ogbonnaya-Ogburu et al. 2020). They list an immediate estimation of the 133 CHI Academy – a prestigious award and recognition in the research field of human–computer interaction – showing that more than 90% recipients were White and that none were of Black/African descent (ibid.). The CHI Academy is supposed to be global and thus has a collective responsibility to ensure that people recognized within the field represent the community. Celebrating people’s achievements is a collective responsibility of the field, and we as researchers should carefully consider whether we are considering all relevant people or whether we are unintentionally neglecting and overlooking people who do not fit the norm. Being chosen for such an honor is not an objective decision but always a negotiation

among groups with people of power (who have already been chosen earlier); thus, groups in power need to consider their own privilege and provide space (and power) to others if we are to see a change. Equity is about providing space, privilege, and power to people entering and transforming the field in new ways – people who are not the norm but who will take the field in new and innovative directions. In such efforts it is critical that we consider that the *diversity dimensions intersect*. Where diversity dimensions intersect, active attention is required to reduce the risk of neglecting important achievements (since they do not fit the norm of evaluating achievement) and recognize how individual conditions serve as barriers.

*Intersectionality* refers to the complex overlapping of diversity dimensions, created to consider the problematic consequences of treating race and gender as mutually exclusive categories of experience and analysis (Crenshaw 1989, p. 139). The problem is that we tend to consider one category exclusively rather than how the categories interlink. “Women” tends to mean White women, and “Black” tends to mean Black men. In her famous paper, Kimberlé Crenshaw shows how a Black woman failed in her legal efforts to demonstrate that General Motors did not hire Black women before 1964 and fired all Black women hired after 1970. General Motors successfully argued that they hired women (White) as well as Black (men) and thus she could not show discrimination since some parts of the case focused on race and others on gender – and that these dimensions were seen as mutually exclusive categories (Crenshaw 1989). Exploring the experiences of Black women in computing, Rankin and Thomas find that “because women of color share the same gender as white women but differ in race, they are subjugated to a different reality and set of social injustices that are often ignored by gender-focused efforts” (Rankin and Thomas 2020, p. 199).

It is critically important to consider how the diversity dimensions intersect instead of addressing categories as mutually exclusive; focusing on single categories means that certain populations risk falling between them and thus are neglected in interventions. They end up as residual categories (Matthiesen and Bjørn 2016, 2017; Matthiesen et al. 2020, 2022) in our diversity dimension classification. Residual categories are the “in-between” categories that do not fit the formal classifications because they are neither-nor. When aspects, things, people, concepts, identities, and so forth are residual, they risk being overlooked and becoming invisible. They do not exist as part of the society receiving attention and thus are forgotten and potentially unintentionally omitted from technology design considerations. Stina Matthiesen in her research on global software development shows how the classification schemes of corporate email addresses disadvantaged software developers working from Poland compared with software developers working from Denmark (Matthiesen et al. 2020). As it turned out, an international company assigned email addresses to developers in Denmark using abbreviations of people’s names; however, software developers in Poland were assigned email addresses beginning with ‘xxx’, indicating that they were not physically located in Denmark. Their colleagues would not respond promptly to emails from addresses beginning with ‘xxx’ because this classification was also used for external consultants, who were not seen as part of the company, and thus not important to answer rapidly.

Because of this labeling and classification scheme, software developers working outside Denmark were disproportionately ignored. This was due not to gender, socio-economic, ability, or other individual diversity dimensions but to the intersection between perspectives on external consultants and perspectives on global work.

## Cultural Taxation and the Imposter Phenomenon

Reaching for equity for all – considering all the diversity dimensions – is a direction and future goal, not where computer science and software development are in 2022. To make the change, we need multiple people from around the globe and in different professions and research areas of computer science education and practice to pave the road to equity and inclusion. Responsibility for gender diversity should not uniquely fall to women and gender minorities to advocate, and it is not the responsibility of immigrants to advocate for ethnic diversity. Instead, it is the work of the majority and of people with power in the field to notice and create space to invite and distribute power for otherwise invisible voices. Equity is also a process of decision power, and of how new groups need to get voice and access to the distribution of value. What counts as value depends on the context – in academia, value includes things like citations, awards, grants, and mentoring of PhD students – and all these criteria are mutually dependent (see Chap. 1). What is often not valued is the effort involved in equity work.

Equity work takes effort and resources and often adds extra work of advocacy for under-represented groups. Often institutions seeking to attract more people from diverse backgrounds will ask the few people within under-represented groups to act as mentors, as role models, and to be visible – atop existing advocacy work and their normal work. Concretely, we, the authors of this book, have multiple times joined events internal or external to the university with the purpose of recruiting more women to computer science. We have been asked to recruit current computer science students from our own program, to help others by acting as mentors or as instructors for programming workshops for women and non-binary individuals. While good intentions underlie these invitations, such work is often unpaid, takes time away from work on subject matter (students’ studies or our research), and is not valued as real work. Fundamentally, such efforts – while important – do not add to people’s CVs more than a little ‘nice to have’, so while peers from majority groups do not have to join such activities, they simply have more time to focus on their individual carriers or studies. The extra burden of diversity work thus risks reducing under-represented groups’ opportunities for individual success. This extra work of minority groups has been identified as *cultural taxation* by Amada Padilla (Padilla 1994; Joseph and Hirshfield 2011).

*Cultural taxation* is the “obligation to show good citizenship towards the institution by serving its needs for ethnic representation on committees, or to demonstrate knowledge and commitment to a cultural group, which may even bring accolades to the institution but which is not usually rewarded by the institution on whose behalf

the service is performed” (Padilla 1994, p. 26). The problem here is not whether diversity work is important for the institution: it is. The problem is that diversity work is seen as important but that, in terms of reward systems for promotion, graduating with excellence, or receiving awards, it is not viewed as relevant for estimating intellectual excellence. Thus, each time under-represented groups spend time and effort on diversity work, they risk reducing the quality of their own resumes. Further, organizations often fail to understand and acknowledge that diversity work cannot simply be turned on and off but is instead embedded in the lived experiences and interactions of under-represented individuals, which at times can be extremely stressful and pose a high risk of burnout (Padilla 1994). Visibility of diverse representation is important – under-represented groups benefit from ‘seeing’ themselves represented in faculty and in auditoriums, and their opinions are important for decision-making. However, it is an ongoing challenge for organizations to ensure that under-represented groups spend their limited time and representation on important and impactful agendas while supporting their careers. Further, organizations should consider how to appreciate the value of diversity work as part of excellence with direct link to awards, promotions, prestige, and privilege. Diversity work of under-presented groups is needed to push the balance towards equity, and seeing under-represented groups succeed is critical for the experience of belonging to a field.

The term *imposter phenomenon* has been used to describe the feeling of not belonging to a field, profession, or community despite results, qualifications, and competences (Clance and Imes 1978). This phenomenon (also referred to as the imposter syndrome) has particularly been identified in high-achievement environments of high competition such as academia (Langford and Clance 1993). Studies have shown that the imposter phenomenon is more prevalent in women and members of under-represented racial, ethnic, and religious groups; thus, researchers have argued that organizations must pay attention to these challenges, which risk countering diversity efforts (Chrousos and Mentis 2020). Actions to mitigate the imposter experience have been proposed as therapeutic approaches; the former chief operating officer at Facebook Sheryl Sandberg wrote the controversial book *Lean In*, wherein she proposes that women overcome the imposter syndrome and take leadership by *leaning in* and sitting at the table (Sandberg 2013). While we do not doubt the presence of the imposter phenomenon in high-achievement environments, we would argue that by introducing the imposter syndrome to the discussion on equity, we risk moving responsibility for the alien experience of under-represented groups from the external surroundings to a personal internalization. Sandberg, in her guidebook for women in tech leadership, places the responsibility for women’s success on women’s own abilities and performance – hiding the role of the institutional conditions that produce unbalanced access to success. The fundamental message in *Lean In* is that women must themselves take power – it is not given to them. However, the missing message is that people in power must relinquish some of their power if organizations are to provide space for alternative voices. Navigating the imposter syndrome – which fundamentally is about discomfort and anxiety in high-achievement workplaces – is not about teaching under-represented groups more

technical skills and expertise while allowing them to navigate in existing biased organizational situations. Instead, it is about changing structurally biased circumstances, allowing them to succeed on their own terms and develop themselves as well as the field of digital technology. In the words of Ruchikan Tulshyan and Jodi-Ann Burey, “Stop telling women they have imposter syndrome”; we should be “fixing bias, not women” (Tulshyan and Burey 2021). It is critically important that we not place the responsibility on the individual to join in but instead consider this challenge of equity as a collective responsibility we all must take – especially people in power.

Equity initiatives are not about creating diversity committees – populated by under-represented groups paying cultural taxes – who can then advise and council decision-makers. Instead, equity initiatives are about inviting the under-represented groups to be full members of the decision power committees and ensuring that the interaction and communication – language and vocabulary – are appropriate for diverse groups and having a respectful and genuine interest in making a change. Women like Sheryl Sandberg who have reached top positions are not automatically the best advocates for equity, since they have managed to navigate the current circumstances and, in that process, risk internalizing the systemic bias on which the system is built. In the process of becoming successful, the few under-represented individuals do much work to fit in and internalize the same metrics and behaviors for what success entails. Therefore, when inviting under-represented individuals to join important decisions as full members, it is important to consider (1) how we can recruit and invite people with different perspectives from ours, and (2) how we can train all decision-makers in equity as a collective responsibility. We cannot expect that simply because an individual is from an under-represented group they are interested in building or know how to build an organization characterized by equity. The challenge for decision-makers (no matter their background) is to figure out how to mainstream equity within the organization.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

