Explaining Algorithmic Decisions with respect to Fairness
Qusai Ramadan¹ Amir Shayan Ahmadian¹ Jan Jürjens¹,² Steffen Staab¹,³ Daniel Strüber⁴

Keywords: Software Fairness; Explainable Software; Model-Based Analysis; UML

1 Abstract

Decision-Making Software (D-MS) may exhibit biases against people on grounds of protected characteristics such as gender and ethnicity. Such undesirable behavior should not only be detected but also explained. To avoid complicated explanations and expensive fixes, fairness awareness has to be proactively embedded in the design phase of the system development. With fairness by design, system developers have to be supported with tools that detect and explain discriminations during the system architecture design [Ra18a].

Only avoiding protected characteristics in a D-MS does not prevent discrimination. Due to data correlations, other data may act as proxies for protected characteristics, thereby causing the so-called discrimination by proxy [GBM17]. There are two possible explanations for the correlations: (1) Societal fact. For instance, if females are more likely to have long hair more than males, then long hair can act as a proxy for the gender. (2) Information flow. The actual input of a D-MS may contain data that resulted from processing protected characteristics. For example, in an insurance company, it might be authorized to use the gender for identifying an insurance tariff but it might be not allowed to use gender for deciding about the reimbursement factor. However, if the insurance tariff is used as input to the reimbursement D-MS, a discrimination against gender have to be reported because the insurance tariff indirectly leaks a signal about the gender to the reimbursement D-MS.

Existing works only consider two approaches: white- (e.g., [Da17]) and black-box approaches (e.g., [GBM17]). While white box approaches can uncover discrimination, they cannot uncover discrimination in the above mentioned sense, as they did not consider possible information flow between system components. While black box approaches may solve this, they do not produce a witness to describe where and how a data flow can happen. Moreover, both approaches cannot be used in the early phase of the system design [Ra18a].

We aim to support developers with tools to reason about hidden flows for protected characteristics to a D-MS during the modeling of the system architecture. Detecting hidden information is a key challenge in security engineering [De76]. However, a model-based information flow analysis approach that supports fairness analysis is lacking [Ra18a]. We propose to develop a model-based discrimination analysis framework (see Fig. 1):

Input: (i) a requirements document containing fairness requirements. (ii) a model (UML) describing the structural and behavioral aspects of a system. (ii) a database of historical data.

Process: (i) Annotating a system model with fairness requirements. For this, we plan to

¹ Universität Koblenz-Landau, Germany. Email: qramadan, ahmadian, juerjens, staab@uni-koblenz.de
² Fraunhofer Institute for Software and System Engineering, Germany
³ University of Southampton, UK
⁴ University of Gothenburg, Gothenburg, Sweden. Email: danstru@chalmers.se

doi:10.18420/se2019-50
extend the privacy UML profile in [Ah17b].

(iii) **Generating fairness constraints** provided as specifications, which have to be satisfied by a system model. In this step, proxies are also identified based on historical data from the system database. (iii) **Verifying the fairness constraints** of step (ii) against the system model (using a model checker). The last two tasks can be realized by extending CARiSMA, a tool that provides different security checks to support model-based security analysis [Ah17a]. **Output:** A witness reporting a violation. For instance, a sequence of actions that if executed will violate the fairness constraints.

**Other open challenges.** Information flow is not the only source for discriminations. Other sources are: First, a nonalignment between the organizational needs and the system models due to misunderstandings between expert stakeholders about fairness terminologies. This challenge can benefit from the model transformation technology, as proposed in our work in [Ra17]. Second, trade-offs between fairness and privacy requirements. A privacy requirement may disallow a D-MS from accessing protected characteristics. Although this requirement sounds as a fairness support, it prevents the D-MS from being able to uncover up-to-date proxies, as a proxy identification requires accessing to protected characteristics. For this, we plan to extend our work on conflicts detection [Ra18b].

**Literatur**


