Meet your Maker: A Social Identity Analysis of Robotics Software Engineering

Carlos Gavidia-Calderon  
The Alan Turing Institute  
United Kingdom  
The Open University  
United Kingdom  
cgavidia-calderon@turing.ac.uk

Amel Bennaceur  
The Open University  
United Kingdom  
amel.bennaceur@open.ac.uk

Tamara Lopez  
The Open University  
United Kingdom  
tamara.lopez@open.ac.uk

Anastasia Kordoni  
Lancaster University  
United Kingdom  
a.kordoni@lancaster.ac.uk

Mark Levine  
Lancaster University  
United Kingdom  
mark.levine@lancaster.ac.uk

Bashar Nuseibeh  
The Open University  
United Kingdom  
bashar.nuseibeh@open.ac.uk

ABSTRACT

Software systems often reflect the values of the people that engineered them: it is vital to understand and engineer those values systematically. This is crucial for autonomous systems, where human interventions are not always possible. The software engineering community shows some positive values—like altruism—and lacks others—like diversity. In this project, we propose to elicit the values of the engineers of autonomous systems by analysing the artefacts they produce. We propose to build on the social identity theory to identify encouraged and discouraged behaviours within this collective. Our goal is to understand, diagnose, and improve the engineering culture behind autonomous system development.

CCS CONCEPTS

• Software and its engineering ➔ Programming teams; • Social and professional topics ➔ Computing profession.

KEYWORDS

robotics software engineering, social identity, autonomous systems, game theory, empirical software engineering

1 INTRODUCTION

We have not completely overcome the longing for the tribe, when a man was still an inseparable part of the collective.  

Mario Vargas Llosa, The Call of the Tribe, 2018

In the early days of computing, it was common for a single developer, or coder, to build production-ready software. Nowadays, software development is an increasingly social endeavour, in which many coders across the globe work together. Software developers have a large impact on modern life through the tools they build. Given their influence, researchers have used methods from the social sciences to address the question of “who software engineers are”. Ethnographic studies have found evidence of a culture among software engineers, with shared beliefs and values [30]. Research examining software development practice has identified aspects of work that motivate [2] or satisfy developers within the workplace [9], and ways in which developers perceive that they are successful [36] or belong to communities [37].

Software development is unique, exhibiting significant differences to other professions. To mention a few, from a sociological perspective developers are unusually altruistic [15]. The open-source movement develops and maintains software artefacts for free, relying mostly on volunteers. For example, the Log4j utility—powering iCloud and Twitter—is free to use. Also, and unlike other engineering disciplines, the profession is mostly unregulated. There is actually a resistance to professional licensing among software professionals [21] and often times, developers do not even need a university degree to be involved in large software projects. On the negative side, software development has a severe gender-diversity problem, even when compared with other STEM disciplines [20]. Social psychology studies show the presence of implicit gender biases that influence hiring decisions in software teams [40].

Robotics software engineers, or just roboticists, have some characteristics that distinguish them from the wider software community. For one, they have diverse educational backgrounds: in addition to training in computer science, many roboticists are formally trained in electrical and mechanical engineering [5, 10, 43]. Roboticists have
also been found to anthropomorphise the systems they build [6]. In common with other STEM disciplines, robotics engineering has a gender diversity problem [22, 38], and sometimes a disregard for ethical concerns [43]. Robotics engineering, in common with other software engineering professions, is a team effort, requiring the cooperation of software engineers, testers, hardware engineers, and product designers to develop autonomous systems [6]. Team members share values and develop new ones during system development. Often unintentionally, they embed these values in the autonomous systems they build. It is crucial to understand the values and relationships within the robotics software engineering community, given that the systems they build are becoming part of our daily lives. Many robotic systems—like drones, self-driving cars, or industrial robots—operate with some degree of autonomy, sharing space and interacting regularly with human beings. A system’s relationship with humans is defined by the values of the engineering team that built it. For society to trust these autonomous systems, they need to trust that the values of its engineering team are aligned with societal values. The values of the makers are reflected in the products they fabricate [5], raising the need to promote the development of an inclusive, altruistic culture within robotics engineering teams.

Decades of research have identified multiple behaviours emerging within the software engineering collective [4]. However, the group dedicated to building software for autonomous systems—like drones, self-driving cars, or industrial robots—is still understudied [43]. In this extended abstract, we propose research to identify the behaviours that originate within the robotics engineering community.

![Figure 1: A game-theoretic model of an identity-driven behaviour within open-source. It models two maintainers (A and B), that can either review code contributions or ignore them. Maintainer B would prefer to ignore them, but this might trigger a response from maintainer A, given that reviewing code is an identity expectation.](image)

Our approach, shown in Figure 2, builds game-theoretic models of identity-driven behaviours, informed by previous work examining cultural aspects of software engineering communities (eg. [2, 9, 36]). Based on these models, we can propose interventions that discourage negative behaviours and promote positive ones. To explain these behaviours, we propose using the social identity approach: a social psychology perspective that explains group behaviour [34]. It establishes that human behaviour can be explained by group membership. These groups can be a function of, for example, gender, nationality, or profession. Social identity has explanatory power. For instance, gender discrimination in the workplace, and division of labour in a household can be explained with social identity models, but resist classic economic analysis [1]. Our focus is on the identities associated to the robotics engineering practice. We will study identity at different levels of granularity: from wide categories like “roboticist”, to more specific ones like “testing engineer”.

In section 2, we present our approach for modelling identity-driven behaviours to investigate interactions within the roboticist community. In section 3, we elaborate on how those models can support the development of an altruistic and inclusive robotics engineering culture. In section 4, we briefly discuss related work. Finally, section 5 concludes the paper.

2 MODELLING THE ROBOTICIST IDENTITY

In software engineering, game-theoretic models have been used to diagnose and remove problems in software processes [13, 14]. In this extended abstract, we propose using game-theoretic models to understand identity dynamics within robotics teams. Game theory is an adequate framework for reasoning about relationships within a social identity [1]. Myerson defines game theory as “the study of mathematical models of conflict and cooperation between intelligent rational decision-makers” [23]. These decision-makers, or players, interact in a game using strategies, that determine the utility they perceive. Using game theory, we can predict player behaviour by obtaining the Nash equilibrium of the game-theoretic model. At equilibrium, each player adopts a strategy that is the best response to the strategies of the rest of the players.

In identity-based game-theoretic models, identity expectations favour some strategies and discourage others [1]. For example, open-source projects receive code contributions from developers outside the core team. These candidate contributions constitute pull requests. In order to be incorporated to the project’s codebase, or merged, they need to be approved by senior developers, called maintainers [16]. Solidarity and cooperation are core values of open-source software [15]. Hence, we expect the “open-source maintainer” identity to favour the strategy of volunteering to review code contributions from the community.

Group members that do not align to identity expectations experience anxiety and discomfort, that we model as utility loss [1]. This negative effect is not limited to the offending member, but extended to the rest of the social group. Following the open-source maintainer example, let us imagine that personal reasons made a maintainer neglect or delay code reviews. This would produce a utility loss on all the project’s maintainers. To restore identity, group members can penalise the offending person to make them...
We can use statistical techniques to find if an identity model's tions as a response. This identity-driven behaviour code reviews, and other maintainers ignoring their code contribute 1, the showcase our approach. In the development process shown in Figure 2. Also use them to test interventions for the removal of undesired not only a diagnosis tool for robotics engineering teams. We can respond to the strategies resulting from the equilibrium analysis that can be explained by identity dynamics. These behaviours could indicate a low level of kinship or altruism within the group, two attributes that have been positively associated with a sense of virtual community in open source development [37].

Our goal is to build game-theoretic models, like the one in Figure 1, of the identities within robotics engineering teams. These models should reflect the expectations and behaviours that arise by being part of the robotics engineering collective. In them, players are team members sharing an identity, like maintainers A and B in Figure 1, and their strategies are behaviours of interest, like “Review Code” or “Ignore Maintainer’s contributions”. Player utility $u_A$ and $u_B$ in Figure 1—depends on a metric team members try to maximise, like number of accepted code contributions for open-source developers.

We can obtain players, strategies, and utility values by mining process data. Model validation will also be driven by process data. For example, the “open-source maintainer” identity model in Figure 1 establishes a dependency between a maintainer’s pull request merges and their own accepted code contributions. Using time series data from code repositories, we can analyse the relationship between these two quantities. For example, a Granger causality test can determine if past values of merged pull-requests contribute to the prediction of future values of accepted code contributions [12]. We can use statistical techniques to find if an identity model’s assumptions and predictions hold. Software engineering is in a privileged position for statistical data analysis. Every activity, from requirements gathering to defect management, is supported by software that generates and exposes data.

From the many dimensions of social identities, our identity models—like the one in Figure 1—only represent a behavioural expectation and the penalty for members that do not comply with it. This modelling scope is defined by the available process data, the behaviours reflected in them, and the statistical methods to determine causal relations between these behaviours. Formulating plausible hypothesis about the behaviours that define an engineering team’s identity is challenging. We plan to gather our candidate identity-driven behaviours from the team under analysis, using observation, semi-structured interviews and surveys. We believe robotics engineers are in a better position to identify the group-level behaviours of their workplace.

3 A BETTER ENGINEERING CULTURE

We can use these game-theoretic models to determine behaviours that can be explained by identity dynamics. These behaviours correspond to the strategies resulting from the equilibrium analysis of an identity’s game-theoretic model. These identity models are not only a diagnosis tool for robotics engineering teams. We can also use them to test interventions for the removal of undesired behaviour, using the process shown in Figure 2.

Let us use the “open-source maintainer” identity example to showcase our approach. In the social identity model shown in Figure 1, the equilibrium analysis predicts some maintainers ignoring code reviews, and other maintainers ignoring their code contributions as a response. This identity-driven behaviour has a negative impact on team morale and productivity, so we will attempt its removal. During process improvement, we posit that by publishing review reminders in the project’s messaging platform would make ignoring reviews unappealing. Under the assumption that these reminders would reduce the utility of the offending maintainer from $u_A = 4$ to $u_A = 2$ (bottom-right leaf in Figure 1), the updated model has an equilibrium where maintainers do not reciprocate by ignoring contributions. These results can serve as an input to later deploy this practice within the team.

Other identity-driven behaviours can be addressed with our approach. For instance, let us imagine we use process data from project management systems to build the social identity model of the “startup roboticist” identity. Then, equilibrium analysis predicts engineers working long hours, discouraging leaving early by scheduling late meetings. This identity-driven behaviour can lead to burnout and a corresponding low level of job satisfaction [9]. Pressure for working extra hours is specially challenging for women, that sadly have the lion’s share of family chores in many households [35]. This outcome signal areas for improvement in the work culture: during process improvement, we can test if some form of “taxation” can remove the long-hours behaviour at equilibrium. In practice, taxation can be implemented by requiring a justification and senior management approval to schedule a late-hours meeting.

![Figure 2](image.png)

**Figure 2**: Improving robotics teams by social identity modelling. Using process data and observing the teams, we build game theoretic models of their social identity (like the one in Figure 1). Equilibrium analysis of these models produce a list of identity-driven behaviours. If any of these behaviours are harmful, we can use the model to test interventions for its removal.

4 RELATED WORK

Existing work has promoted the need to consider ethics [27] and values [41] during the development of software systems. Some approaches have been proposed to assess and study values in software engineering [42], to incorporate social values in software design patterns [19], and to measure the impact of values in requirements engineering activities [26]. Values are well studied in human-computer interaction and information systems [7], and especially in healthcare [18]. Existing approaches focus on early stages of the development process [29], with little attention given to the satisfaction of values in deployed software systems or on the design of sociotechnical interventions to operationalise these values. We will build on existing work in Values@Runtime [3] to
consider values during operation in order to monitor and reflect on these values. We will also explore the use of emotional goals of roboticists to design inclusive software and processes [17].

Social identities may also help guide the planning of adaptive interventions [24, 28] that require engagement and cooperation of humans. Existing work on enabling software systems to reason about identities to enable cooperation [11] provides a direction for planning identity-aware interventions.

5 CONCLUSION AND FUTURE WORK

In this extended abstract, we used two identity-driven behaviours to showcase our approach: code reviewing in open-source projects, and work-life balance in startup companies. For demonstration purposes, this paper relied heavily on assumptions. In future work, we will use the procedure described in Figure 2 to explain each of these behaviours using process data. While data for the work-life balance issue would require an industrial collaborator, the code reviewing study can take advantage of the features and data publicly available at GitHub.

We will also embed Responsible Research and Innovation (RRI) principles in our work, by adopting the AREA (anticipate, reflect, engage, and act) framework [25]. Our proposed approach models identity-driven behaviours within a robotics engineering team, using software process data from individual engineers’ contributions. We anticipate these models —along with the data used to build them— can be used to identify specific engineers and how they conform to a team’s identity. Our approach is designed to diagnose groups, and its application to assess individuals is a clear misuse. Using the model’s outputs to single out and penalise team members is a risk we need to mitigate. Our plan is to apply differential privacy techniques [8] to the process data we extract, to guarantee the privacy of the members of a robotics team under study.

Acting in line with a group identity enables a sense of responsibility. Research has shown that a strong common identity involves a sense of duty, for acting in certain ways consistent with in-group values. By understanding a robotics team value system, responsibility within the team is turned from an individual’s expectation into a team’s expectation, meaning that taking the praise or blaming becomes a group attribute. Thinking about responsibility in this way may reduce psychological distress, especially when creating autonomous systems for time-constrained situations, such as emergencies.

In taking a mixed methods approach to examine cultural aspects of software engineering for autonomous systems [33], our vision is to translate an improved understanding of robotics engineering into societal gains. We will accomplish this by focusing on understanding the identity of developers and designing appropriate sociotechnical interventions for improving the working culture of robotics engineering teams. We can orient the altruism of the open-source movement towards urgent and neglected engineering challenges, like misinformation detection in non-English languages [31]. We may also alleviate the robotics engineers shortage, increasing the hiring pool by addressing its gender-diversity problem [32].

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/V026747/1, EP/R013144/1]; and Science Foundation Ireland [grant number 13/RC/2049_P2].

REFERENCES


Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009