Research Article

Kheng Lee Koay*, Matt Webster, Clare Dixon, Paul Gainer, Dag Syrdal, Michael Fisher, and Kerstin Dautenhahn

Use and usability of software verification methods to detect behaviour interference when teaching an assistive home companion robot: A proof-of-concept study

Abstract: When studying the use of assistive robots in home environments, and especially how such robots can be personalised to meet the needs of the resident, key concerns are issues related to behaviour verification, behaviour interference and safety. Here, personalisation refers to the teaching of new robot behaviours by both technical and non-technical end users. In this article, we consider the issue of behaviour interference caused by situations where newly taught robot behaviours may affect or be affected by existing behaviours and thus, those behaviours will not or might not ever be executed. We focus in particular on how such situations can be detected and presented to the user. We describe the human–robot behaviour teaching system that we developed as well as the formal behaviour checking methods used. The online use of behaviour checking is demonstrated, based on static analysis of behaviours during the operation of the robot, and evaluated in a user study. We conducted a proof-of-concept human–robot interaction study with an autonomous, multi-purpose robot operating within a smart home environment. Twenty participants individually taught the robot behaviours according to instructions they were given, some of which caused interference with other behaviours. A mechanism for detecting behaviour interference provided feedback to participants and suggestions on how to resolve those conflicts. We assessed the participants' views on detected interference as reported by the behaviour teaching system. Results indicate that interference warnings given to participants during teaching provoked an understanding of the issue. We did not find a significant influence of participants' technical background. These results highlight a promising path towards verification and validation of assistive home companion robots that allow end-user personalisation.

Keywords: human–robot interaction, companion robots, behaviour interference, formal verification

1 Introduction

A long-term goal of robotics research is the use of assistive robots in the home. Such robots have started to appear in various guises ranging from stationary helpers [1] to cleaning robots [2] to robotic companions [3,4]. In previous research, companion robots have been designed to serve useful functions for their users, while carrying out those tasks in a socially acceptable manner [5]. The combination of an autonomous mobile robot and

* **Corresponding author: Kheng Lee Koay,** Department of Computer Science, School of Physics, Engineering & Computer Science, University of Hertfordshire, College Lane, Hatfield, Hertfordshire, AL10 9AB, United Kingdom, e-mail: K.L.Koay@herts.ac.uk

Matt Webster: School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool, L2 2QP, United Kingdom, e-mail: m.p.webster@ljmu.ac.uk

Clare Dixon: Department of Computer Science, University of Manchester, Manchester, M13 9PL, United Kingdom, e-mail: clare.dixon@manchester.ac.uk

Paul Gainer: Department of Computer Science, University of Liverpool, Liverpool, United Kingdom, e-mail: p.gainer@liverpool.ac.uk

Dag Syrdal: Department of Computer Science, School of Physics, Engineering & Computer Science, University of Hertfordshire, College Lane, Hatfield, Hertfordshire, AL10 9AB, United Kingdom, e-mail: dagsyrdal@gmail.com

Michael Fisher: Department of Computer Science, University of Manchester, Manchester, M13 9PL, United Kingdom, e-mail: michael.fisher@manchester.ac.uk

Kerstin Dautenhahn: Departments of Electrical and Computer Engineering and Systems Design Engineering, University of Waterloo, Ontario, Canada, e-mail: kerstin.dautenhahn@uwaterloo.ca

a "smart-home" environment, where the robot is able to extend its capabilities via access to the home sensor network, has been investigated in a number of large-scale projects, e.g. refs. [6,7], motivated by the use of robotic solutions to address the concerns of cost and care issues resulting from an ageing population [8,9].

The use of home assistance robots to help older adults stay independent in their homes faces many challenges. One of these challenges is to allow the robot to be personalised, e.g. that the robot can be taught to change its functional behaviours in response to the changing needs of the older adult. Our previous research [7] investigated these issues, proposing and evaluating a teaching system in a human–robot interaction (HRI) experiment. Results were encouraging, showing the potential of the system to be used easily by a number of stakeholders, including health professionals, formal and informal carers, relatives, friends and the older persons themselves to adapt the robot to meet changing needs.

End-user personalisation is an active area of research since it has been recognised that robots to be used "in the field" need to allow users to adapt, modify and teach a robot. Despite the best intentions to provide user-friendly interfaces, often only experienced programmers can achieve complex behaviour prototyping [10], but research has demonstrated the feasibility to create systems that support end-user robot programming and personalisation. To give a few examples, Lourens and Barakova [11] suggested a user-friendly framework to allow users with minimal programming experience to construct robot behaviours. Building on this work, pilot tests demonstrated how such an approach could be used, e.g. in the area of robot-assisted therapy for children with autism [12]. Furthermore, trigger-action rules have been suggested and tested with end-user developers to personalise home automation environments [13] and achieve robot behaviour personalisation [14]. Recent trends of End User Development and associated challenges have been identified [15], with a focus on approaches specific to Internet of Things and rule-based systems, e.g. the trigger-action paradigm. For a recent systematic review on end-user development of intelligent and social robots with a focus on visual programming environments, see ref. [16].

A key issue that arises from end-user personalisation of robot behaviours, and which is the focus of this article, is that of *behaviour verification* and *behaviour interference*. Behaviour verification is concerned with the effect of adding behaviours via the teaching system and checking whether the new behaviours violate operational goals. Our previous research on methods of behaviour verification is described in refs. [17–20]. Behaviour interference for a home-assistive robot is a part of a verification approach but deals with the *consequences* of teaching new behaviours

which may inadvertently affect the execution of existing behaviours or not be executed themselves due to already existing behaviours.

This article describes a behaviour interference detection mechanism embedded into a teaching system for a home companion robot, designed to be used by carers, relatives and older adults themselves (rather than robotics experts or programmers). We conduct an evaluation of the system with 20 participants (age range 26–69) in order to gain insights on their views of the functionality of the system, but importantly, to also investigate the *usability* of such a system, which would be a crucial factor in any future envisaged deployments of such systems in people's own homes or care homes. Few studies have investigated an actual deployment of companion robots in real-world settings. Examples include studies focusing on therapeutic and educational outcomes for children with autism, e.g. a 1-month study involving daily HRIs in children's homes [21], or the year-long deployment of a therapeutic robot used by staff in a special needs nursery school [22]. Results from such "field studies" highlight the importance of usability and a number of challenges have been identified that will influence whether or not people are willing to keep using such a system. In a pioneering study involving home companion robots, De Graaf et al. described a 6-month, long-term study which placed 70 autonomous robots in people's homes [23]. Investigating cases of non-use, i.e. refusal or abandonment of the robot, the authors conclude "the challenge for robot designers is to create robots that are enjoyable and easy to use or (socially) predictable to capture users in the short-term, and functionally-relevant and possess enhanced social behaviours to keep those users in the longer-term" [24] (p. 229).

The contributions of the article are to identify and classify when newly added behaviours might affect (or be affected by) the execution of existing behaviours and report on a HRI study in relation to this. The HRI study evaluates the usability of the interference detection system and provides an opportunity to assess participant actions and reactions towards the system.

The remainder of this article is organised as follows. Section 2 describes the overall setting of this study and relevant background, including descriptions of the HRI scenario, the robot behaviours and the previously developed behaviour teaching system (TEACHME). Section 3 discusses our approach to formal verification and behaviour interference checking, analysing and categorising behaviour interactions. Section 4 outlines the TEACHME system's novel enhancement that allows users to add new behaviours to the robot and being notified of possible behaviour interference. Section 5 describes the user evalu-

ation carried out with the results being reported and discussed in Section 6. Concluding remarks and future work are presented in Section 7.

2 Setting and background

The research is being conducted in a typical suburban British 3-bedroom house in a residential area off campus but near the University of Hertfordshire. It has normal house furnishings but has been upgraded to a smart home. It is equipped with sensors and cameras which provide information on the state of the house and its occupants. Over 60 sensors report on electrical activity, water flow, doors and cupboard opening/closing etc. User locations are obtained via ceiling mounted cameras [25], and robot locations via ROS navigation [26]. Since this location has been used for many different HRI studies, there are no permanent residents occupying the house, but its ecological validity is far greater than laboratory experiments performed on campus. However, in order to allow for researcher-led controlled experiments, the setting is more constrained than the deployment of robots in real homes occupied by their owners/tenants. We call this location, which bridges the gap between a "real" and entirely "simulated" home environment (laboratory), the *UH Robot House* which is available to University of Hertfordshire researchers but also other researchers as an environment to test and evaluate smart home and robotics technology [27]. The Robot House, as a natural and realistic setting of a home environment, has been used in many HRI studies, e.g. [6,28–31].

The physical sensors relevant to the present study range from sensors monitoring activity of electrical devices in the house (e.g. "fridge door is open," "microwave is on," "TV is on" etc.), to sensors attached to furniture (e.g. detecting operation of cupboard door, drawers etc.), to sensors monitoring water flow and temperature (able to detect e.g. "toilet is flushing," "taps are running" etc.) and, finally, pressure sensors (e.g. located on sofas, beds etc. to indicate occupation).

The study reported here used a commercially available robot, the Care-O-bot3® robot manufactured by Fraunhofer IPA [32]. It is a multi-purpose, mobile manipulator that has been specifically developed as a mobile robotic assistant and companion to support people in domestic environments and is based on the concept of a robot butler (Figure 1) [33].

The robot's high-level decision-making uses a production rule approach where each behaviour comprises
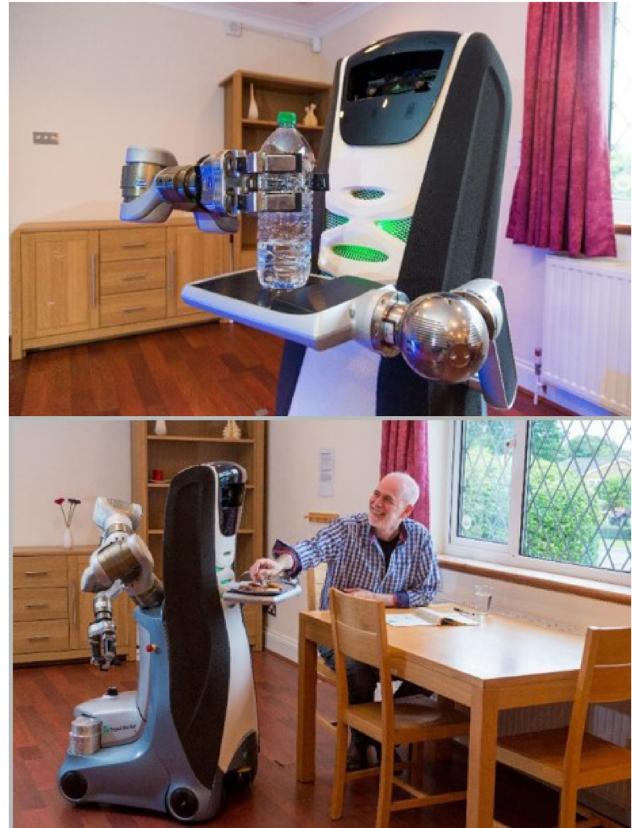

Figure 1: Illustration of Care-O-bot3® operating in the UH Robot House. Photography by Pete Stevens, www.creativeempathy.com.

sets of rules (preconditions or guards) which, if satisfied, execute actions. The rules can check the house and robot sensor values both instantaneously and within a temporal horizon (e.g. "has the doorbell rung in the last 10 seconds?"). Actions are generally related to the robot but can also set other values which can be subsequently checked by the rules. Thus, actions are either robotic (e.g. "move to location X, raise tray"), or sensory/memory based (e.g. "User has been informed to take her medicine"). A more detailed description of the ontology of the house and the robot control system approaches are described in the studies by Saunders et al. [34,35].

Robot behaviours defined as production rules are held as tables in a mySQL database. The rules themselves are encoded as SQL statements and are generated by the TEACHME teaching system, described in more detail in Section 4.

Memory-based values are also held as "sensors" and are used to define or infer knowledge about the house or activities within the house at a higher semantic level. For example, it may be inferred from electrical sensory activity in the kitchen that the "sensor" called "Preparing meal" is currently true. Other memory sensors are used to cope

with on-going events in the house which are not reflected by the physical environmental sensors (similar to Henderson and Shilcrat [36]). For example, a sensor with the label, "User has been reminded to take their medicine" might be set if the robot has given such a reminder to the user and would typically be used to ensure that the reminder was not repeated. Temporal additions to the rules allow the system to formulate rules such as "has the user been reminded to take their medicine in the last 4 hours?"

Behavioural selection is via priority. Thus, where more than one behaviour has all of the preconditions evaluate as "true," the behaviour with the highest priority will execute first. If all the priorities are equal and all the preconditions to the rules are true, then a non-deterministic choice of the behaviours will be made (in practice, the first rule from the rule set is chosen. However, as the rule set is returned by an SQL query the order of results is not guaranteed, which makes the choice non-deterministic). The use of priorities provides a mechanism for resolving conflicts between actions when the conditions for more than one rule hold in a given instant (we call this *behaviour interference*).

The Care-O-bot3® robot [32] is equipped with facilities for manipulating the arm, torso, "eyes," robot LEDs, tray and has a voice synthesiser to express given text. Typical robot actions would be for example, "raise tray," "nod," "look forward," "move to location x," "grab object on tray," "put object x at location y," "say hello" etc.

Production rules can be created in three ways. First, by low-level coding (e.g. using C++ or Python). Second, by a medium-level teaching mechanism which allows easy creation of behaviours and setting of behavioural priorities, but relies on the user to cope with higher-level memory-based issues. We envisage that the second facility would be used by technical "experts" generating sets of behaviours for the first time. However, creating behaviours in this way is very similar to low-level programming in that a very logical and structured approach to behaviour creation is necessary. Third, a high-level teaching facility is provided which allows the user to easily create behaviours. This generates appropriate additional scaffolding code but does *not* allow priority setting, all newly added behaviours are automatically generated with equal priority. The cost of this simplification is a loss of generality; however, it is compensated for by ease of use.

As we concentrate in providing a mechanism for (non-technical) end users to create behaviours, behavioural interference resolution is a challenge. Other approaches to this include providing preferences e.g. in Soar [37], planning, e.g. Hierarchical Task Networks [38] or learning rule utilities e.g. in ACT-R [39]. However, all of these approaches require detailed knowledge of the underlying system, as well as an understanding of the concept of interference and how to use such systems to resolve it. These requirements make it unsuitable for end users such as older adults or their carers who want to use a home companion robot. One of the key aims of this study was to allow *non-expert* users without detailed technical knowledge to recognise behavioural interference and reflect on how they might approach resolving it.

Next, we explain formal verification and how it is being used in our approach for detecting possible behaviour interference as a consequence of users teaching the robot new behaviours.

3 Formal verification and behaviour interference checking

Formal methods are a family of mathematical approaches which allow for the specification, design and analysis of computer systems [40]. Formal methods can also be used to verify software and hardware systems, in a process known as formal verification. There are a wide variety of software tools available for formal verification, including model checkers and automated theorem provers. The aim of formal verification is to show the correctness (or incorrectness) of algorithms and protocols using mathematical analysis or formal proof. Formal verification has been used extensively in the design and development of safety- and mission-critical computer systems [41,42].

Formal verification is often used in the design and implementation stages of software development, prior to deployment. After this point, if the software is to be modified then the formal verification process must be repeated. While many formal verification tools are automatic in their operation (e.g. model checkers like SPIN [43], NuSMV [44] or PRISM [45]), the process of creating and validating models is often not automatic, and must be done "by hand." Formal verification has been applied to autonomous robots in various settings [46] including HRI [47,48], home service robots [49] and collaborative robot applications [50].

In our previous work, we explored the use of model checkers for the formal verification of the robot behaviours within the Robot House [17–20]. In the study by Webster et al., our approach was based on a model of the activity of a typical person within a house [17]. In the study by Dixon et al. [18] an input model for the model checker NuSMV [44] was constructed by hand, and later this process was automated by a tool that can directly read in sets of behaviours and automatically generate

an input model [20]. This could potentially be used to re-generate models of robot behaviours and be used to formally verify properties of the system with newly added behaviours. However, due to the complexity of explaining counter models to users we chose to take a different approach.

In the present study, we use static checking of behaviours to identify potential interactions between them. This could be used by a technical expert setting up the behaviours initially for the robot or by a user after deployment to personalise the behaviours. In particular, here we consider the case where the user adds new behaviours for the robot to follow. When a new behaviour is created by the user it is possible for the new behaviour to interact with other existing behaviours in what we term "behaviour interference."

Example 1. The system may contain a behaviour (B24) that says *If the fridge door is open go to the kitchen* (with priority 30). A newly added behaviour (B37) might be *If the fridge door is open and someone is sitting on the sofa go to the sofa and say "the fridge door is open"* (with priority 10). As the priority of the second behaviour (B37) is less than the first (B24) and whenever the preconditions of the second behaviour are satisfied then the preconditions of the first behaviour are also satisfied then the newly-added behaviour will never run.

Checking for and reporting such conditions allow the users to identify when new or existing behaviours will not or might not be executed even when their preconditions are satisfied.

3.1 Behaviour interference

Note that in the scenario we discuss here, with users adding behaviours to the behaviour repertoire of a robot, the new behaviours are given the same priority. However, when defining behaviour conflicts we consider a more general case where newly added behaviours could have *any* priority, so the same system could also be used by technically competent users who would also be permitted to input behaviour priorities.

Analysis of the robot behaviours revealed that some potential problems with a new behaviour can be quickly identified without the use of a model checker. A behaviour *b* is defined as follows:

$$b : \text{ IF } p_1 \wedge p_2 \wedge \ldots \wedge p_n \text{ THEN } A,$$

where p_i are preconditions that must evaluate to true in order for some sequence of actions A to take place. The use of the logical *and* connective "$\wedge$" specifies that all n preconditions must be true. The set of preconditions for a behaviour b is denoted $P(b)$. The behaviour has a priority $\pi(b) \in \mathbb{N}$ which determines its relative priority. Recall, if there are a number of behaviours whose preconditions are all true, the robot scheduler will execute the behaviour with the highest priority. If all the priorities are equal, then the scheduler will choose non-deterministically one of the behaviours for execution (in practice, the first rule from the rule set is chosen. However, as the rule set is returned by an SQL query the order of results is not guaranteed, which makes the choice non-deterministic).

Given the behavioural selection algorithm of the robot scheduler, it is possible for a behaviour to always be chosen over another. For example, if the preconditions of behaviour b_1 are the same as the preconditions of behaviour b_2, and b_1 has a higher priority than b_2, then b_1 will always execute instead of b_2. In fact, this is also true if b_1's preconditions are a subset of b_2's preconditions, as whenever b_2's preconditions are true, b_1's preconditions must also be true. In this case, we say that b_1 *overrides* b_2, or conversely, that b_2 is *overridden* by b_1:

Definition 1. A behaviour b_1 *overrides* another behaviour b_2 if $P(b_1) \subseteq P(b_2)$ and $\pi(b_1) > \pi(b_2)$.

In Example 1, behaviour B37 (the newly added behaviour) is overridden by behaviour B24 so behaviour B37 will never be executed.

It is also possible for a behaviour b_1 to be scheduled instead of b_2 in some circumstances, but not others. For example, if the two behaviours b_1 and b_2 from the previous definition have equal priorities, then either behaviour may be chosen to execute. This is called *interference*:

Definition 2. A behaviour b_1 *interferes with* another behaviour b_2 if $P(b_1) \subseteq P(b_2)$ and $\pi(b_1) = \pi(b_2)$.

Example 2. Assume now that we have behaviours as described in Example 1 where both behaviours have priority 10. We will refer to these as B24a and B37a. Now behaviour B24a interferes with behaviour B37a. In situations where the fridge door is open, and it is not the case that someone is sitting on the sofa then B24a will be executed. However, when both the fridge door is open, and someone is sitting on the sofa then either behaviour might be executed. In Example 1, behaviour B37 *will* never be executed. Here there are situations where behaviour B37a *might* never be executed.

Overriding and interference demonstrate two ways in which behaviours can prevent the execution of other behaviours. It is also possible to identify the potential for overriding and interference in a wider range of cases. For example, if a behaviour b_1's preconditions are a superset of the preconditions of b_2, i.e. $b_2 \subseteq b_1$, and the extra preconditions of b_1, i.e. $P(b_1) \backslash P(b_2)$, may also be true at some point during the execution of the robot, then we can say that behaviour b_1 *potentially* overrides b_2:

Definition 3. A behaviour b_1 *potentially overrides* another behaviour b_2 if $P(b_2) \subseteq P(b_1)$ and $\pi(b_1) > \pi(b_2)$.

Furthermore, we can extend this idea to interference, allowing for a behaviour to *potentially* interfere with another behaviour of the same priority:

Definition 4. A behaviour b_1 *potentially interferes with* another behaviour b_2 if $P(b_2) \subseteq P(b_1)$ and $\pi(b_1) = \pi(b_2)$.

Definitions 1–4 provide a set of guidelines for identifying conflicts and potential conflicts between the robot's behaviours. Additionally, they can be used to identify when robot's behaviour set contains behaviours that are likely to overlap and result in unpredictable or undesired activity. The guidelines are summarised in Table 1.

Example 3. Let us assume that a behaviour set consists of two behaviours:

$$b_1 : \text{IF } p_1 \text{ THEN } a_1,$$
$$b_2 : \text{IF } p_1 \wedge p_2 \wedge p_3 \text{ THEN } a_2,$$

where $\pi(b_1) = 70$ and $\pi(b_2) = 50$. Note that behaviour b_1 overrides behaviour b_2 (by Definition 1) as $\{p_1\} \subseteq \{p_1, p_2, p_3\}$ and $\pi(b_1) > \pi(b_2)$. Assume we add a new behaviour

$$b_3 : \text{IF } p_2 \text{ THEN } a_3,$$

where $\pi(b_3) = 0$. Then behaviour b_2 potentially overrides behaviour b_3 (by Definition 3) as $\{p_2\} \subseteq \{p_1, p_2, p_3\}$ and $\pi(b_2) > \pi(b_3)$.

These guidelines can be computed for the robot's behaviour database in less than a second, meaning that they can be used by the robot's TEACHME system to quickly

1. *Make the robot say 'the doorbell is ringing' when the doorbell rings.*
2. *Make the robot move to the sofa area when the TV is on and raise its tray.*
3. *If it is Tuesday and midday make the robot say 'It's time for your medicine'.*

Figure 2: Initial set of behaviours taught to the robot during training of the participant.

determine in real-time whether a new behaviour suggested by the user is likely to conflict with existing behaviours. While useful and efficient, these guidelines do not provide the same level of verification made possible by exhaustive state space analysis using model checkers. However, they do allow a partial analysis of the robot's software that can be used to give timely and meaningful feedback to the robot's users.

The guidelines above are implemented in a software tool called the *Behaviour Checker* (*BC*). The BC works by parsing two databases, one containing the existing behaviours used by the robot, and the other containing the new behaviours which have been defined by the user. After parsing, the new behaviours are compared to the already existing behaviours, and behaviour conflicts are identified. Table 1 shows the different types of feedback generated by the BC. The feedback given to the user, for a new behaviour b_n, and an existing behaviour b_e, can be seen in Figure 6. Note that this is simplified for the user in two ways. First, in cases where more than one definition applies, the BC will output only the most severe conflict, with overriding being the most severe, followed by interference, potential overriding and potential interference. For example, if $P(b_n) = P(b_e)$ and $\pi(b_n) > \pi(b_e)$, then both $P(b_n) \subseteq P(b_e)$ and $P(b_e) \subseteq P(b_n)$, so following Definitions 1 and 3, b_n both overrides and potentially overrides b_e and the BC will output only the former. Second, in the case where a Definition is satisfied by more than one existing behaviour, only one of the existing behaviours will be shown to the user at a time to avoid overloading the user with an extensive list of behaviours.

Table 1: Behaviour Checker feedback for different conflicts

	$\pi(b_n) < \pi(b_e)$	$\pi(b_n) = \pi(b_e)$	$\pi(b_n) > \pi(b_e)$
$P(b_n) \subseteq P(b_e)$	b_n is potentially overridden by b_e	b_n interferes with b_e	b_n overrides b_e
$P(b_e) \subseteq P(b_n)$	b_n is overridden by b_e	b_n potentially interferes with b_e	b_n potentially overrides b_e

4 The TEACHME system and behaviour interference notification

In this section, we describe the TEACHME system that allows the users, carers or relatives to input new behaviours into the robot. Full details of this part of our system, as well as evaluations of its usability and usefulness in a user study can be found in ref. [7]. Note, a key difference of TEACHME, compared to other popular approaches on human–robot teaching that extensively use machine learning approaches (see overview in ref. [51]), is the explicit representation of robot behaviours as rules implemented as tables in a database. A similar rule-based approach is taken by Porfirio et al. [47]; however our approach supports utilisation of external sensors and behaviours that involve continuous time (e.g. remind the user to take medicine every 4 h). This allows us to conduct experiments that are situated within a realistic Robot House smart home environment ontology which includes knowledge about sensors, locations, objects, people, the robot and (robot) behaviours.

The motivation of this approach was to provide a format that can easily be read, understood and manipulated by users who are not programmers, in order to facilitate easy and intuitive personalisation of the robot's behaviour by end-users.

We also explain in this section how any problems detected between two behaviours are presented to the user.

4.1 Teaching system – TEACHME

In order to create behaviours the user must specify *what needs to happen* (the actions of the robot) and *when those actions should take place*. An example of the user teaching interface (i.e. GUI) is shown in Figures 3–5 and displays the actions a non-technical user would use to create a simple behaviour to remind the user that the kettle is on.

The steps consists of "what" the robot should do followed by "when" the robot should do it. Steps are as follows: the user chooses to send the robot to the kitchen entrance and then presses a "learn it" button. This puts the command into the robot memory (top of Figure 3). Then the user makes the robot say, "The kettle is on, are you making tea?" This is not in the robot's current set of skills and so is entered as text input by the user (bottom of Figure 3). This is followed by a press of the "learn it" button. Now the two actions are in the robot's memory and the user can define when these actions take place.

The user is offered a number of choices based on events in the house (such as user and robot location, the settings of sensors showing the television is on, the fridge door is open, etc.) and a diary function (reminders to carry out a task, e.g. to take medicine or phone a friend at a particular day and time) shown on the top of Figure 4. The user chooses a house event occurring in the kitchen (bottom of Figure 4). Again this is followed by pressing the "learn it" button. Having completed both "what" and "when" phases the user is shown the complete behaviour for review and can modify it if necessary (Figure 5). Once satisfied, the user presses a "make me do this from now on" button and the complete behaviour becomes part of the robot's behaviour database.

4.2 Behaviour interference detection and reporting

The behaviour interference function was embedded within the TEACHME system. This is called when the completed behaviour is signalled by the user to be ready for scheduling on the robot.

A challenge for this type of notification is to make it both understandable to a naïve user and to provide mechanisms for rectifying possible interference issues.

The screen that appears when interference is detected is shown in Figure 6. It informs the user that a problem has been found in two behaviours – an existing behaviour and the behaviour the user just created. It continues to list the "when" factors that are causing the interference – which are effectively the preconditions or sets of preconditions that are equivalent between the behaviours. It also offers some choices to help the user ignore or rectify the interference.

In order to evaluate our system we conducted a proof-of-concept user study in the UH Robot House.

5 User evaluation

The aim of this study was to investigate the usability of the system and if there was any difference in understanding the concept of behaviour interference between already technically trained users (e.g. participants with a computer science background) and those without any systems/programming or robotics background. For future applications of assistive home companion robots, it is essential to know whether users of any systems designed
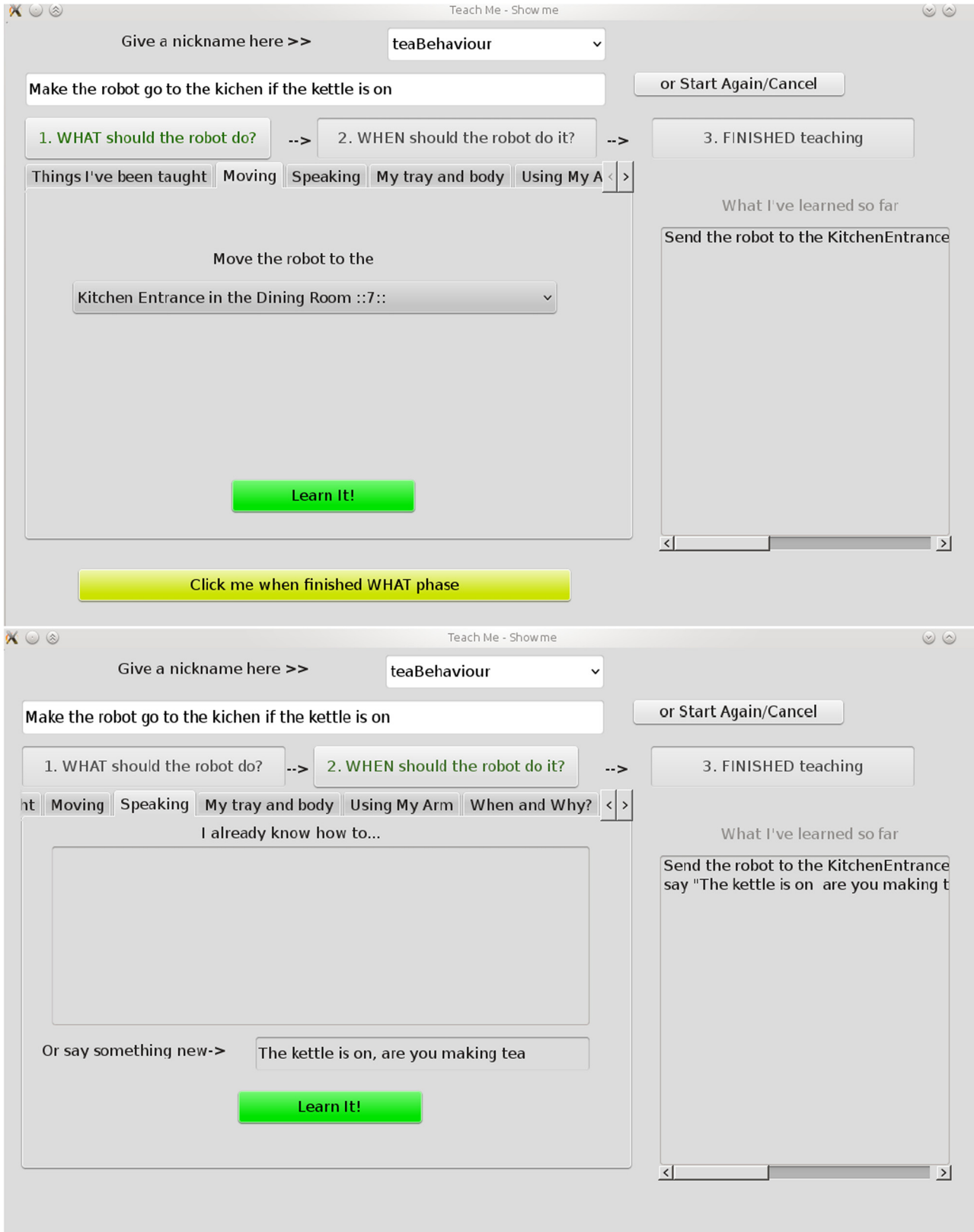
Teach Me - Show me

Give a nickname here >>   teaBehaviour

Make the robot go to the kichen if the kettle is on

or Start Again/Cancel

1. WHAT should the robot do? --> 2. WHEN should the robot do it? --> 3. FINISHED teaching

Things I've been taught | Moving | Speaking | My tray and body | Using My A < >

Move the robot to the

Kitchen Entrance in the Dining Room ::7::

Learn It!

What I've learned so far

Send the robot to the KitchenEntrance

Click me when finished WHAT phase

Teach Me - Show me

Give a nickname here >>   teaBehaviour

Make the robot go to the kichen if the kettle is on

or Start Again/Cancel

1. WHAT should the robot do? --> 2. WHEN should the robot do it? --> 3. FINISHED teaching

ht | Moving | Speaking | My tray and body | Using My Arm | When and Why? < >

I already know how to...

Or say something new->   The kettle is on, are you making tea

Learn It!

What I've learned so far

Send the robot to the KitchenEntrance
say "The kettle is on are you making t

Figure 3: Screenshots of the teaching interface. The "What" phase where the user has directed the robot to move to the kitchen and say "The kettle is on, are you making tea?."
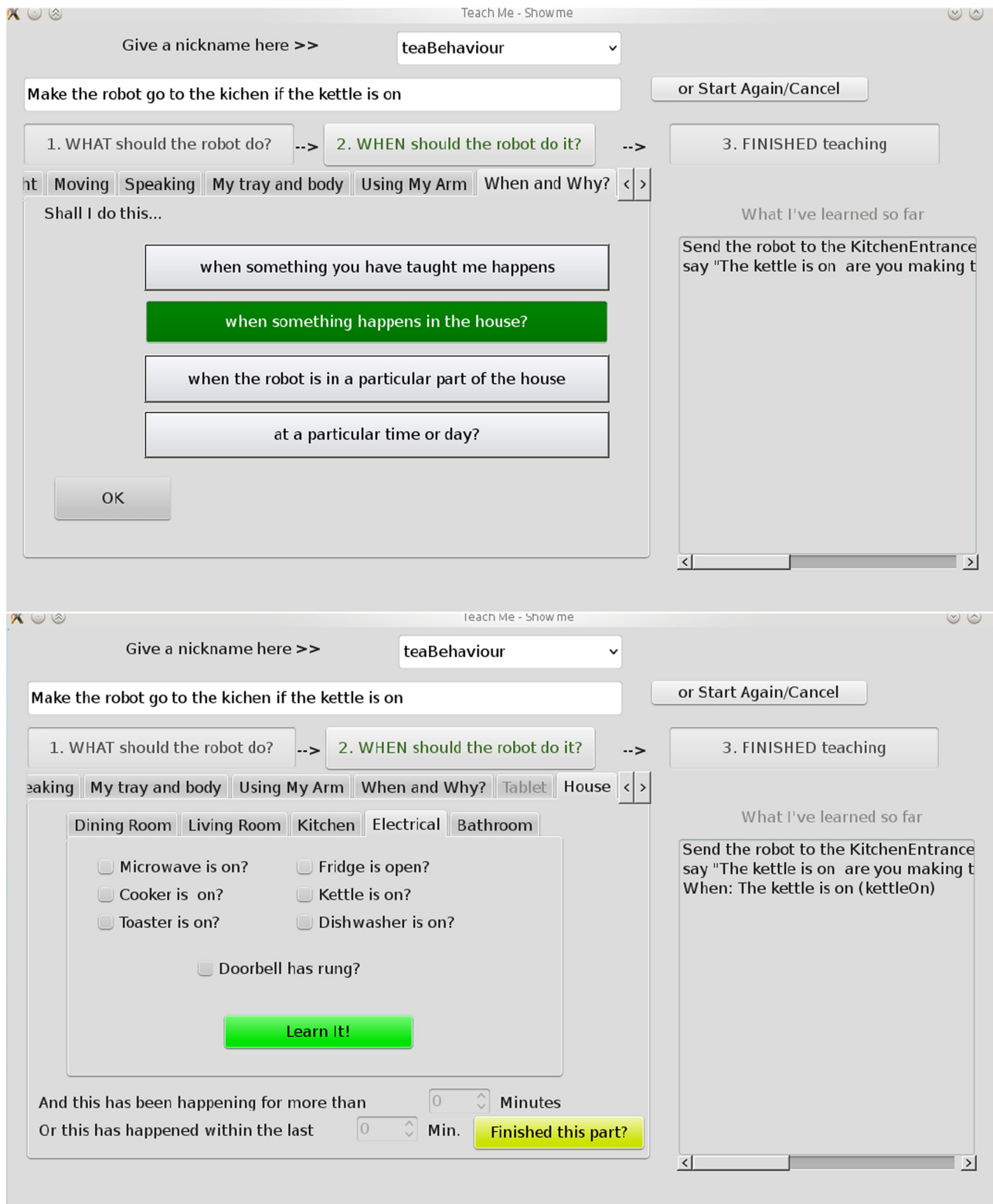
Figure 4: Screenshots of the ᴛᴇᴀᴄʜMᴇ teaching interface. In the "When" phase the user chooses to consider events in the house and clicks on the kettle option.

in this domain need to have technical training to under-stand behaviour interference. The HRI study was carried out with the approval of the University of Hertfordshire Ethics Committee, Protocol number COM/SF/UH/02083.
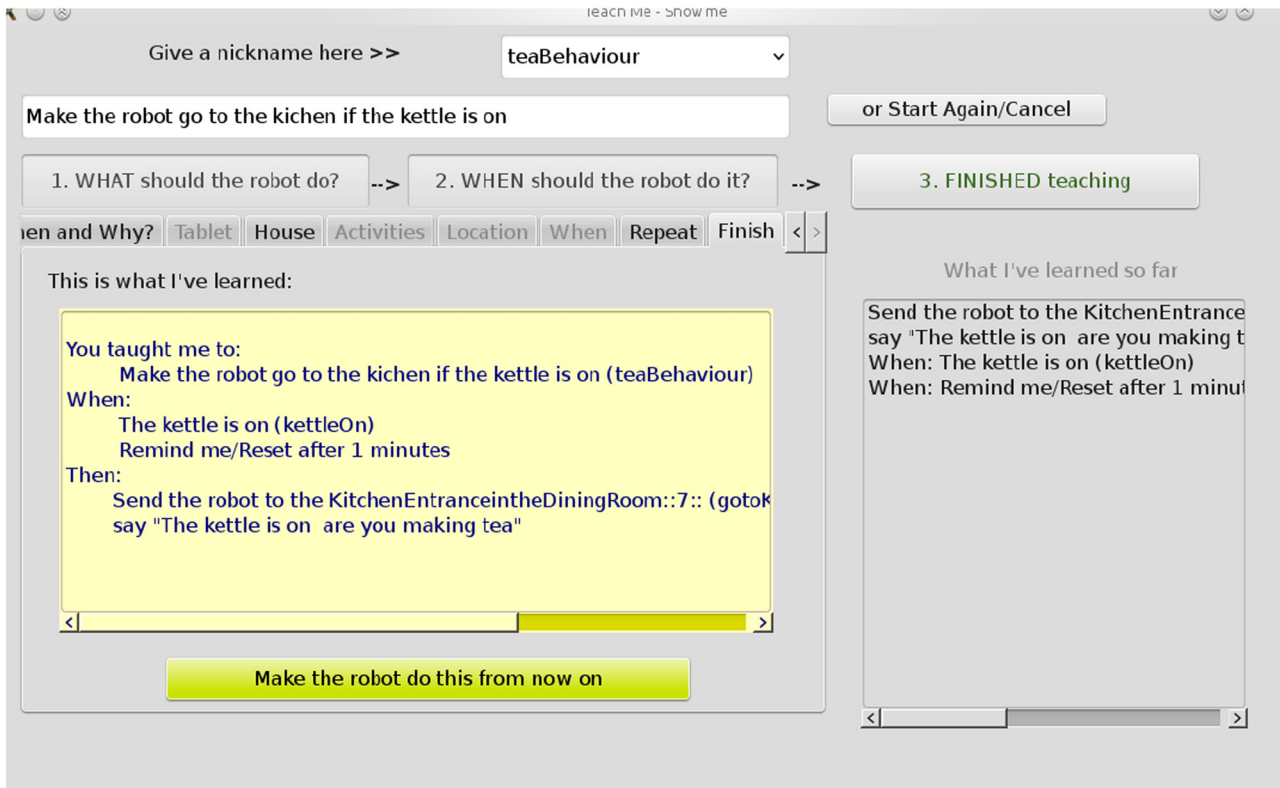
Figure 5: Screenshots of the TEACHME teaching interface. The user reviews the complete taught behaviour.

5.1 Research questions

Our research questions were as follows:

1. Is the robot teaching system still considered usable by users when it includes automatic behaviour interference checking?
2. Do users find the mechanism for informing them about behaviour interference effective in helping them understand the nature of behaviour interference?
3. Does the participant's background or gender have any effect on detecting and solving behaviour interference issues?

Note, regarding the third research question we expected that participants' familiarity with robots might have an impact on the results. Regarding gender we did not have any particular hypothesis, so this was an exploratory question.

5.2 Participants

Twenty participants were recruited, 11 female and 9 male, who took part individually in the experiment. The participants had a mean age of 48.15 and a median age of 47.

The youngest participant in the sample was 26 and the oldest participant was 69 with an interquartile age range of 35–61. The participants were either postgraduate students at the University of Hertfordshire (typically with a computer science background) or people who had previously expressed a wish to interact with robots in the Robot House. The latter group, which included some University staff, had minimal technical experience, although some had taken part in robotics experiments in the past. Of the 20 participants, 17 had interacted with robots in general before (although not necessarily this robot, and none had used the TEACHME system previously) and eight had experience in programming robots. Figure 7 shows the distribution of the participants based on their prior robot programming experience, with their age and gender information.

5.3 Proof-of-concept experiment: methodology

The study was conducted in the UH Robot House with the Care-O-bot3® robot, see Section 2 for detailed descriptions.

On arriving at the Robot House, each participant was invited to review and sign the appropriate ethics forms
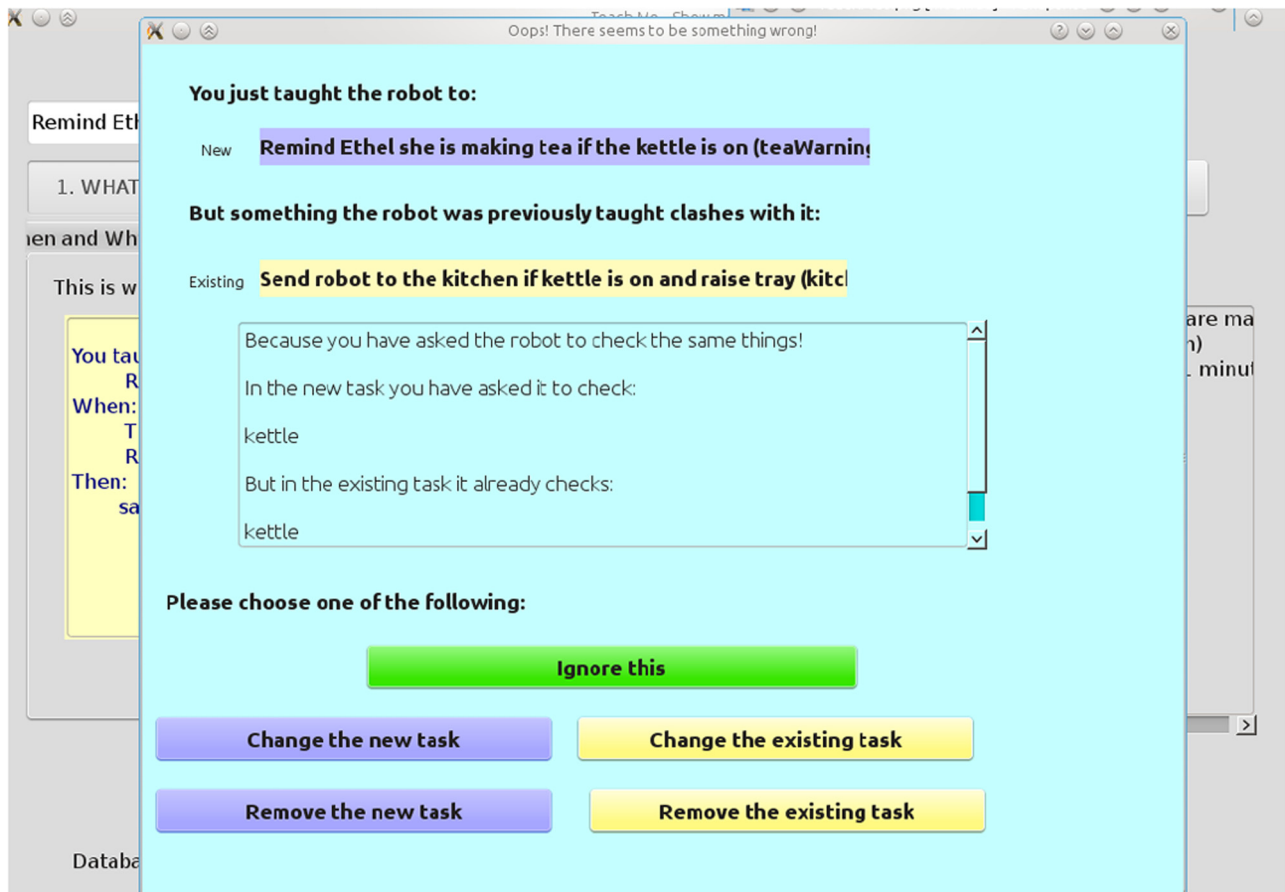
Figure 6: Screenshots of the TEACHME teaching interface. The system detects a possible interference between two behaviours and asks the user to take action to resolve it.

and complete a short demographics questionnaire on gender, age and technical background.

For the actual HRI scenario presented to participants, we used narrative framing, allowing participants to feel part of a consistent "story." This technique has been used successfully in HRI, cf. refs. [52–55]. It has also been used in long-term HRI studies on home companion robots where, inspired by ref. [56], it facilitated prototyping episodic interactions in which narrative is used to frame each individual interaction [57], or to provide an overall narrative arc encompassing multiple interactions and scenarios [28]. In the present study, we used narrative framing extensively, including multiple props, personas and allocated roles for participants.

5.3.1 Scenario: new technician

The participants were asked to imagine that they had just been accepted for a job with a fictitious company called

"Acme Care Robots" (ACR) as a robot technician. It was explained that ACR builds robots to assist older adults with the aim of helping them stay in their own home (rather than being moved to a care home), and that it was their job to create behaviours on the robot. They were told that this was their first day of training and following training they would receive their first assignment.

In order to reinforce the illusion of technician training, we used props – all persons involved in the experiment were given white laboratory coats to wear. This included the participant, the experimenter (who also acted as the "trainer") and a third person required to be present by the University for safety purposes (Figure 8).

Training commenced with the experimenter introducing the TEACHME system and explaining in general terms how it worked. The participants were then invited to use the system to create the behaviours shown in Figure 2. After each behaviour was taught the participant was invited to test the behaviour, e.g. after teaching the robot to say "the doorbell is ringing" when the doorbell rings,
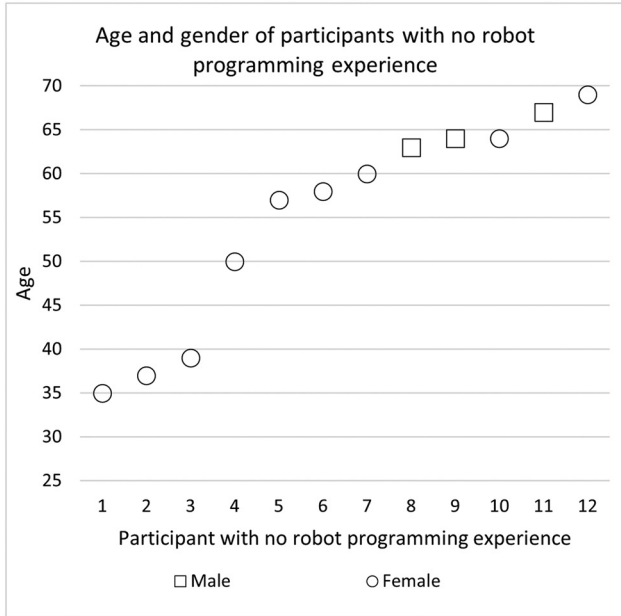
(a)
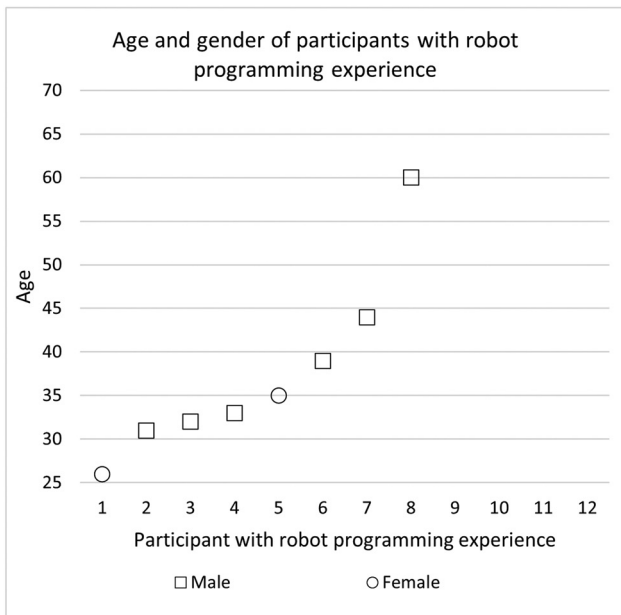
(b)

Figure 7: Demographics of participants' gender, age and robot programming experience. (a) Demographics of participants with no robot programming experience. (b) Demographics of participants with robot programming experience.

the participant or experimenter would ring the physical doorbell in the Robot House and check that the resulting robot actions were correct.

Three individual behaviours were taught in the training phase. These were chosen to be relatively simple but also to exercise the speech, movements and diary functions of the robot.

Figure 8: Training in progress. The experimenter is facing the window, the participant is facing the laptop computer with the TEACHME system running. The Care-O-bot3® robot is present in the background.

Having completed training the participant was given their first assignment. This involved setting up behaviours for a fictitious older lady, a "persona" called "Ethel" who supposedly lived in the Robot House. Details of the assignment sheet given to the participant are shown in Figures 9 and 10.

The choice of a naturalistic setting for the study, the Robot House, the narrative framing approach (as discussed above), the introduction of a user persona, and the use of props was meant to enhance the believability, plausibility and ecological validity of the scenarios as well as enhance users' engagement and immersion. The aim was to encourage participants to immerse themselves into the role of a robot training technician. Props have been a common tool in human–computer interaction research and development for decades, e.g. ref. [58]. With regards to the development of scenarios for home companion robots, narrative framing of HRI scenarios has been used successfully, e.g. ref. [28,57]. The use of personas has been pioneered by Alan Cooper in human–computer interaction. According to Cooper, "Personas are not real people, but they represent them throughout the design process. They are hypothetical archetypes of actual users." [59], see also studies on user and robot personas in HRI, e.g. refs. [60,61].

5.3.2 Interfering behaviours

The behaviours shown in the assignment sheets consisted of four tasks. The first three we call the "A" section, and the fourth task the "B" section. In the following

---

First Assignment

Acme Care Robots have been contacted by the relatives of an older lady called Ethel.
Ethel has a Care-O-bot3®robot assisting her in her home.
Ethel's relatives want you to create and test some new behaviours that they think will help Ethel in her daily living activities.

This is what we know about Ethel:

- *Ethel lives at home alone.*
- *Although she has some movement difficulties (she uses a Zimmer frame) she is able to cope relatively well with daily living activities, however she finds carrying objects around the house difficult.*
- *She often forgets to turn off appliances when cooking (for example leaving a pot of potatoes boiling).*
- *Because of her hearing difficulties her TV volume is very loud – and thus hearing the doorbell or phone becomes difficult.*
- *Ethel's best friend is Mabel – and sometimes they organise a day out together. Ethel sometimes forgets to call Mabel.*

Figure 9: Background information given to the participant after the training phase has completed. This is then followed by the actual assignment shown in Figure 10.

description, the priorities of all the behaviours are equal, i.e. $\pi(b_{1a}) = \pi(b_{1b}) = \pi(b_{1c})$, etc.

The first task of the "A" section was designed to make the robot attend the kitchen with its tray raised whenever activity was detected. Thus, if "Ethel" were carrying out some kitchen activity the robot would be present. Activity was inferred from the kettle being on, the fridge door being open, or the microwave being on. This can be formalised using the definitions from Section 3.1 as the following three behaviours:

First Assignment – continued:
Ethel's relations have asked you to set up the following behaviours on the Care-O-bot3®.

1. *To help with her movement difficulties please make sure that the robot is at the kitchen entrance with its tray raised whenever the kettle is on, or the fridge door is open or the microwave is on.*
2. *Whenever Ethel is sitting on the sofa please make the robot join her there.*
3. *Ethel sometimes forgets that she is making a cup of tea, please inform her and send the robot to the sofa area when the kettle is on.*

Final behaviour.

4. *Ethel sometimes forgets to close the fridge door. If the fridge door is open please tell Ethel.*

Figure 10: Behaviour assignments for the participant.

$$b_{1a} : \text{ IF } k \text{ THEN } a_1; a_2,$$
$$b_{1b} : \text{ IF } f \text{ THEN } a_1; a_2,$$
$$b_{1c} : \text{ IF } m \text{ THEN } a_1; a_2,$$

where k means that the kettle is on, m means that the microwave is on and f means that the fridge door is open. The action a_1 means that the robot moves to the kitchen entrance, and a_2 means that the robot raises its tray.

The second task was designed so that the robot would proceed to the sofa if Ethel was sitting there:

$$b_2 : \text{ IF } s \text{ THEN } a_3,$$

where s means that Ethel is sitting on the sofa and a_3 means that the robot moves to the sofa. It was implied that Ethel could place something on the robot's tray while in the kitchen, which the robot would then bring to her once she sat on the sofa.

The third task contained an interference issue. It required that if the kettle was on the robot should inform Ethel that she might be making a cup of tea, and it should proceed to the sofa:

$$b_3 : \text{ IF } k \text{ THEN } a_4; a_3,$$

where a_4 means that the robot says, "Are you making tea?" As the preconditions of b_{1a} and b_3 are equal ($P(b_{1a}) = P(b_3)$) and their priorities are equal ($\pi(b_{1a}) = \pi(b_3)$) the behaviour b_{1a} interferes with behaviour b_3 and vice versa by Definition 2, and both behaviours potentially interfere with each other by Definition 4.

Similarly, the fourth task (the single task in the "B" section) contained an interference issue. This task re-

quired that the robot should say, "The fridge door is open," when the fridge door was open:

$$b_4 : \text{ IF } f \text{ THEN } a_5,$$

where a_5 means that the robot says that the fridge door is open. As the preconditions for b_{1b} and b_4 are equal $(P(b_{1b}) = P(b_4))$ and their priorities are the same $(\pi(b_{1b}) = \pi(b_4))$ behaviours b_{1b} and b_4 interfere and potentially interfere in a similar way to b_{1a} and b_3.

5.3.3 Order of interference detection

The behaviour interferences described above were presented to users during the proof-of-concept experiment. Two versions of the TEACHME system were created: one with behaviour interference checking, and one without in order to evaluate how participants responded to these two versions.

To rule out familiarity effects (where all participants experienced the checking procedure in the same order) the two versions of the software were pseudo-randomised between participants. Note, in both conditions, i.e. with and without behaviour checking, after using the interface to teach the robot a new behaviour, participants would test the behaviour they created on the physical robot multiple times. In the checker "off" condition, participants would be puzzled when the robot did not carry out the desired task. Participants could then go back to the interface and try to resolve the problem. In the checker "on" condition, they would be alerted to why this interference has happened and could subsequently attempt to resolve the problem.

The 20 participants were randomly allocated into two groups of 10 persons (10 in group X and 10 in group Y). Before the "A" section those participants in group X had checking turned on – the Y group had checking turned off. Once the "A" section was complete the X group would have checking turned off and the Y group have checking turned on. This meant that for example a participant might receive an interference warning after the "A" section issue, but not after the "B" section. Another participant might receive an interference warning after the "B" section issue but not after the "A" section. Figure 11 shows participants' distribution across the two experimental conditions based on robot programming experience. Note that the distribution of participants with robot programming experience was not equal, i.e. three participants in group X and five participants in group Y.

5.4 Measures

After both the A section and the B section, the participant was asked to complete a questionnaire (Table 2). The ques-
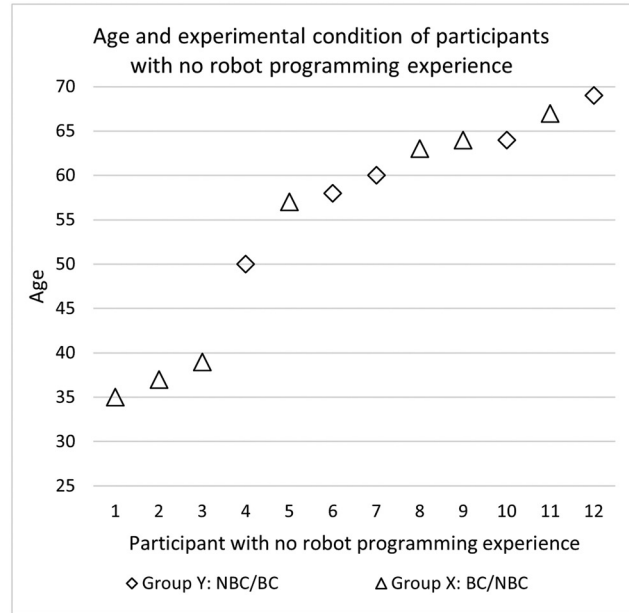
(a)
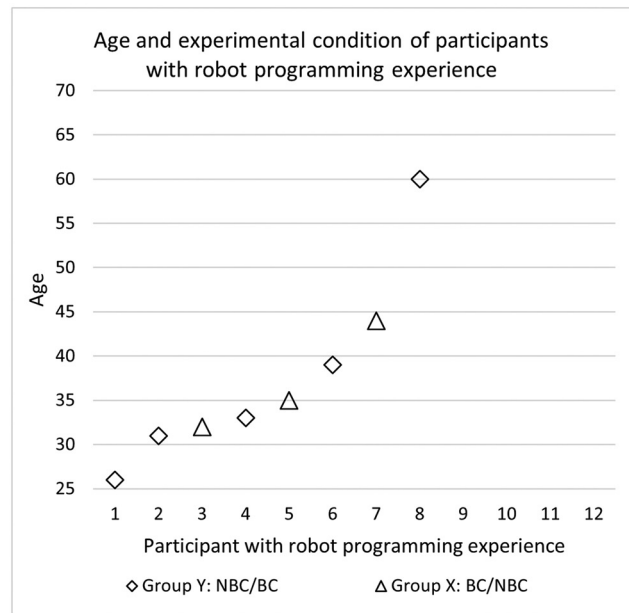
(b)

Figure 11: Participants' age distribution across different experimental conditions. (a) Distribution of participants with no robot programming experience. (b) Distribution of participants with robot programming experience.

Table 2: Usability Questionnaire used in the present study

Modified Brooke's Usability Scale (5 point Likert scale), items 1–10, complemented by 2 additional items 1 – "Not at all," to 2 – "Not really," 3 – "Maybe," 4 – "Yes probably," 5 – "Yes definitely"

1.	I think that I would like to use the robot teaching system like this often
2.	I found using the robot teaching system too complex
3.	I thought the robot teaching system was easy to use
4.	I think that I would need the support of a technical person who is always nearby to be able to use this robot teaching system
5.	I found the various functions in the robot teaching system were well integrated
6.	I thought there was too much inconsistency in the robot teaching system
7.	I would imagine that most people would very quickly learn to use the robot teaching system
8.	I found the robot teaching system very cumbersome to use
8.	I felt very confident using the robot teaching system
10.	I needed to learn a lot of things before I could get going with the robot teaching system
11.	The robot teaching system helped me resolve inconsistencies in the relative's instructions
12.	The robot teaching system helped me understand how behaviours can interfere with each other

Note, the last two items are referred to as "Question 11" and "Question 12" in this article.

tionnaire was based on a modified version of Brooke's System Usability Scale (SUS) which rates the general usability of an interactive system [62]. Answers to questions are based on a 5-point Likert scale with values ranging from 1 – "Not at all," to 2 – "Not really," 3 – "Maybe," 4 – "Yes probably" and 5 – "Yes definitely." Note, half of the questions are positively phrased (odd numbered questions), half are negatively phrased (even numbered questions). We had used this scale in a previous validation of the TEACHME system [7]; however, here we extended the questionnaire with two additional Likert scale items referred to as *Question 11* ("The robot teaching system helped me resolve inconsistencies in the relative's instructions") and *Question 12* ("The robot teaching system helped me understand how behaviours can interfere with each other"). Note, Q11 addresses the ability of the system to solve the problems at hand (e.g. resolving inconsistencies), while Q12 is probing participants' understanding of the principle that different robot behaviours may interfere with each other.

The participants were also given an opportunity to write an expanded answer to these two questions if they wished. Following the "B" section, participants could provide further written comments.

6 Results and discussion

In this section, we provide the results for the user study. In the following, the abbreviation "BC" is for "Behaviour Checking" and "NBC" is "No Behaviour Checking."

6.1 Usability outcome variables

There were three outcome variables: one from the responses to the SUS (based on items 1–10 shown in Table 2), and the two additional items (Questions 11 and 12) mentioned above.

6.1.1 SUS responses

SUS responses for each of the two repeated measures conditions are presented in Table 3. Note, "difference" reported in this and other tables refers to the differences in scores for each participant in this repeated measures study. To calculate the SUS score, 1 is subtracted from each of the values of the odd numbered questions. The values for the even numbered questions are subtracted from 5. The sum of these scores is then multiplied by 2.5, which results in a score between 0 and 100. A SUS score above 68 is considered above average, while scores less than 68 are considered below average [62].

The mean scores are consistent with our previous experiment on usability of the TEACHME system [7] (that did not involve any behaviour interference detection)

Table 3: SUS responses for the two repeated measures conditions

Interference	Mean	SD	Med.	Min	25th–75th	Max
With	80.12	11.65	81.25	65	67.5–90.625	97.5
Without	78.38	11.25	76.25	62.5	70–87.5	97.5
Difference	1.75	6.29	0	−5	−2.5–2.5	20

Table 4: SUS response in terms of presentation order

Order	Mean	SD	Med.	Min	Max
BC (NBC/BC)	78.25	11.31	77.5	65	97.5
NBC (NBC/BC)	76	9.66	75	62.5	90
Diff. NBC/BC	2.25	7.95	−1.25	−5	20
BC (BC/NBC)	82	12.29	82.5	65	95
NBC (BC/NBC)	80.75	12.7	78.75	62.5	97.5
Diff. BC/NBC	1.25	4.45	0	−2.5	12.5

NBC = no behaviour checking, BC = behaviour checking.

with high usability. This indicates that there were no significant or salient differences between the two repeated measures conditions and suggests a positive response to research question 1 ("Is the robot teaching system still considered usable by users when it includes automatic behaviour interference checking?").

Table 4 considers the effects of the presentation order, i.e. whether behaviour checking is turned on or off for Sections A and B. NBC/BC denotes behaviour checking was turned off for Section A and then on for Section B (group Y, above the line in the table) and BC/NBC denotes that behaviour checking was turned on for Section A and then off for Section B (group X, below the line in the table). Tables 7 and 10 have a similar structure. Presentation order effects in terms of SUS responses were insignificant (Table 4).

6.1.2 Question 11 of usability questionnaire

As Tables 5 and 6 suggest, there were differences between the two repeated measures conditions. These differences were significant with a moderate effect size [63] calculated in the manner suggested by Rosenthal [64] (Wilcoxon sign-rank test $p < 0.01$, effect size $r = 0.60$), and participants considered the system with behaviour checking more favourably, partly providing a positive response to research question 2 ("Do users find the mechanism for informing them about behaviour interference effective in help-

Table 5: Question 11 – The robot teaching system helped me resolve inconsistencies in the relative's instructions (no. of persons)

	Not at all	Not really	Maybe	Yes, probably	Yes, definitely
NBC	1	3	7	7	2
BC	0	1	4	9	6

NBC = no behaviour checking, BC = behaviour checking.

Table 6: Question 11 – Descriptives

	Mean	SD	Med.	Min	25th–75th	Max
NBC	3.3	1.03	3	1	3–4	5
BC	4	0.86	4	2	3.5–5	5
Diff.	0.7	1.03	0	0	0–1	4

NBC = no behaviour checking, BC = behaviour checking.

ing them understand the nature of behaviour interference?").

Table 7 suggests that there were no effects from presentation order in terms of responses to the two different conditions for question 11.

6.1.3 Question 12 of usability questionnaire

Tables 8 and 9 suggest that there were significant differences between the two repeated measures conditions. These differences were significant with a moderate effect size (Wilcoxon sign-rank test $p < 0.05$, effect size $r = 0.57$). Thus, participants considered the interference detection system helped them understand the interference issue better, providing a positive response to research question 2.

Table 10 suggests that there were no effects from presentation order in terms of responses to the repeated measures variable.

6.2 Demographics outcome variables

6.2.1 Gender

There were no relationships between the repeated measures conditions and gender for any of the three outcome variables (see the difference between mean in Table 11). This means that the data do not show a significant impact

Table 7: Question 11 – Presentation order

Order	Mean	SD	Med.	Min	Max
BC (NBC/BC)	4.1	0.88	4	3	5
NBC (NBC/BC)	3.4	0.97	3.5	2	5
Diff. NBC/BC	0.7	0.82	0.5	0	2
BC (BC/NBC)	3.9	0.88	4	2	5
NBC (BC/NBC)	3.2	1.14	3	1	5
Diff. BC/NBC	0.7	1.25	0	0	4

NBC = no behaviour checking, BC = behaviour checking.

Table 8: Question 12 – The robot teaching system helped me understand how behaviours can interfere with each other (no. of persons)

	Not at all	Not really	Maybe	Yes, probably	Yes, definitely
NBC	2	1	3	7	7
BC	0	1	0	8	11

NBC = no behaviour checking, BC = behaviour checking.

Table 9: Question 12 – Descriptives

	Mean	SD	Med.	Min	25–75th	Max
NBC	3.8	1.28	4	1	3–5	5
BC	4.45	0.76	5	2	4–5	5
Difference	0.65	1.09	0	0	0–1	4

NBC = no behaviour checking, BC = behaviour checking.

of gender on the usability of the system and how it helped them to resolve inconsistencies and understand behaviour interference.

6.2.2 Prior interaction with robots and programming experience

There were no relationships between the repeated measures conditions and prior interactions with robots for any of the three outcome variables (Table 12).

There were also no relationships between experience of programming robots and the repeated measures conditions for any of the three outcome variables (Table 13).

Thus, our data do not reflect a significant effect of participants' background on detecting and solving behaviour interference issues, in response to research question 3 ("Does the participant's background have any effect on detecting and solving behaviour interference issues?").

Table 10: Question 12 – Presentation order

	Mean	SD	Med.	Min	Max
BC (NBC/BC)	4.6	0.52	5	4	5
NBC (NBC/BC)	4	1.25	4	1	5
Diff. NBC/BC	0.6	0.97	0	0	3
BC (BC/NBC)	4.3	0.95	4.5	2	5
NBC (BC/NBC)	3.6	1.35	4	1	5
Diff. BC/NBC	0.7	1.25	0	0	4

NBC = no behaviour checking, BC = behaviour checking.

Table 11: Outcome differences between the two repeated measures conditions according to gender

	Mean	SD	Med	Min	25–75th	Max	Wilcoxon *p*. val
SUS-M	0.56	5.12	0	−5	−2.5–2.5	12.5	0.5
SUS-F	2.73	7.2	0	−2.5	−2.5–2.5	20	
Q.11-M	0.44	0.53	0	0	0–1	1	0.61
Q.11-F	0.91	1.3	0	0	0–1.5	4	
Q.12-M	0.33	0.5	0	0	0–1	1	0.46
Q.12-F	0.91	1.38	0	0	0–1	4	

M = male, F = female, Q = question.

7 Conclusions, limitations and future work

We defined and implemented a static behaviour checking system that considers the preconditions and priorities of behaviours to identify cases where behaviours will never be executed or may not be executed. We incorporated this into the ᴛᴇᴀᴄʜMᴇ System on the Care-O-bot3® robot in the Robot House that fed back problems to users by a graphical user interface. We carried out a user evaluation study to elicit their views on this system.

Regarding the static behaviour checking system we elected to carry out checks on behaviour interference as it was straightforward to explain results to an end-user. An alternative approach would be to add the new behaviour, re-construct the underlying model of the system and carry out full model checking. The main issue with this approach we perceive is how to explain any output to the end-user.

While the participants in this study did not find the two conditions (with behaviour checking and without behaviour checking) different in terms of *general usability*, they did find that the behaviour checking approach was

Table 12: Outcome differences between the two repeated measures conditions according to participants with and without prior robot interaction experience

	Mean	SD	Med.	Min	25–75th –	Max	Wilcoxon *p*. val.
SUS-P	0.29	3.94	0	−5	−2.5–2.5	12.5	0.21
SUS-NP	10	11.46	12.5	−2.5	5–16.25	20	
Q11-P	0.65	1.06	0	0	0–1	4	0.41
Q11-NP	1	1	1	0	0.5–1.5	2	
Q12-P	0.65	1.17	0	0	0–1	4	0.5
Q12-NP	0.67	0.58	1	0	0.5–1	1	

P = prior interaction, NP = no prior interaction, Q = question.

Table 13: Outcome differences between the two repeated measures conditions according to participants with and without prior robot programming experience

	Mean	SD	Med	Min	25–75th	Max	Wilcoxon p val.
SUS-P	−0.94	2.65	−1.25	−5	−2.5–0.625	2.5	0.19
SUS-NP	3.54	7.42	1.25	−2.5	−2.5–5	20	
Q11-P	0.75	1.39	0	0	0–1	4	0.7
Q11-NP	0.67	0.78	0.5	0	0–1	2	
Q12-P	0.75	1.39	0	0	0–1	4	1
Q12-NP	0.58	0.9	0	0	0–1	3	

P = programmed before, NP = not programmed before, Q = question.

significantly more useful for resolving and understanding inconsistencies in the robot's behaviour.

Furthermore, we found that technical background did not have a significant effect in understanding the nature of behaviour interference.

If those results can be confirmed in future larger scales studies, then this is an encouraging direction for the development of robot personalisation systems that allow robot behaviour creation and modification that can be carried out by both technical and non-technical users. Specifically, the issue of behaviour interference and the resulting conflicts could be understood by non-expert users when detected and reported effectively. However, although such mechanisms can report on interference, a separate issue is how an end user could potentially deal with and resolve the problem. In the study reported in this article, the user had only limited options (deleting or amending behaviours or simply ignoring the issue). In more complex cases, solutions may be to provide additional behaviour priority modification. Further investigation of these more complex cases and their possible solutions would be a valid next step in this area of research.

The integration of the behaviour checking system into the Robot House also represents an additional tool to compliment formal verification. Formal verification is often used "offline," i.e. prior to system deployment. In addition, it is usually performed by highly-skilled verification engineers. However, the use of behaviour checking based on static analysis of behaviours described in this article has shown that such tools can be used online, during the operation of a multi-purpose companion robot. Furthermore, it can be used to give timely and informative feedback directly to end-users during robot behaviour teaching.

There are several limitations to our work. First, the relatively small number of participants is a major limitation, and the sample of participants is not ideally balanced in terms of gender and programming background. Second,

it would have been helpful to have each participant carrying out several sessions in a longer-term study. Third, video recording and analysis of participants' actions and reactions to those two conditions, and their interactions with the experimenter during the experiment, could have added additional detailed information on how participants experienced the two conditions. Finally, we only tested each participant in two conditions, with and without behaviour checking. A larger scale study, with a between-participant design, could study different variations of the behaviour checking approach, in order to gain more detailed feedback on the usability and usefulness of the system, and how to improve the system, rather than, as we did in this study, only considering the presence or absence of behaviour checking.

With respect to future work there are a number of directions we could improve the initial static behaviour checking system. Currently, the behaviour checking system is limited to behaviours in which the triggering condition is a sequence of conjunctions, i.e. p_1, p_2 and p_3, etc. A more general approach would use Boolean formula as the triggering condition, so that disjunctions, negations and nested formula could also be included in the condition, e.g. $(p_1 \text{ or } p_2)$ and $((\text{not } (p_3)) \text{ and } p_4)$. Previously, we assumed that the preconditions of behaviours were conjunctions of atomic statements and represented these as sets of preconditions. The definitions for overriding and interfering were presented as subset checking between sets of preconditions for two behaviours. Let $F(b_i)$ denote the Boolean formula representing the preconditions for behaviour b_i (which in Section 3 was a conjunction). An alternative way to show $P(b_1) \subseteq P(b_2)$ where $P(b_1)$ and $P(b_2)$ were representing sets of conjunctions is to check whether $F(b_2) \rightarrow F(b_1)$ (where $\rightarrow$ is logical implication) is a valid formula. If the preconditions of behaviours can now be more complex Boolean formula, to check conditions that previously were subsets, i.e. $P(b_1) \subseteq P(b_2)$ we would now need to check that $F(b_2) \rightarrow F(b_1)$ is a valid

formula. We could program this directly or call to a theorem prover for propositional logic or SAT solver. This would allow a greater range of flexibility of programming the robot, both by developers at the code level, and end-users using the TEACHME system.

Second, a more detailed study of the allowable preconditions could be made so that interactions between temporal constraints (such as it is between 10 and 11 and it is morning) or spatial constraints (such as near the sofa and in the living room) could be dealt with properly. For these, we would need better representations related to time and space with definitions for terms such as morning, afternoon and evening. Constraint solvers or spatial reasoners might be useful for reasoning here but we would have to check what types of statements are allowed concerning time or space and how best to reason about them together with Boolean formulae.

Third, when adding a new behaviour we have just presented one behaviour to the user that satisfies the guidelines. A more detailed study might show all behaviours that match the guidelines and these could perhaps be ordered in some way with the stronger conditions first (overriding, then interfering, then potential overriding and finally potential interfering).

Another avenue of future work would be to integrate the use of more powerful verification tools like model checkers and theorem provers and to convert the low-level technical output of these types of systems into easy-to-understand feedback for an end-user. In particular, a parser for the UH Robot House rules that translated these into input to the NuSMV model checker was described in ref. [20]. If we add a new behaviour we could check properties such as the pre-conditions for the new property will never be satisfied (so it will never be run) or if the pre-conditions are satisfied on some execution sometime the behaviour will run. Similar checks for existing behaviours could also be carried out. However, how to report those results to non-technical users needs more investigation.

Furthermore, the TEACHME interface would be more robust if it supports priority settings and provides functionality for users to create novel temporal and non-temporal memory variables as opposed to relying on predetermined memory variables, created by technical users, in the present version. The future development of the user interface we designed might benefit from insights gained in recent research on interfaces to allow novice users to comprehend and debug software and systems, e.g. refs. [65,66].

Finally, larger scale user studies specifically targeting older adults as well as adults with dementia or other health-related conditions, ideally performed in participants' own homes, could further illuminate the usefulness and usability of our developed system and its impact on applications to support healthy and independent living. More generally, the techniques and systems presented in this article could be further developed and applied as well to other application domains, including therapy and education, where robots need to be taught new behaviours by non-expert, novice users. In addition to using the system in order to provide assistance functionalities, our approach could also be extended, e.g. to teaching a robot social behaviours.

Author contributions: The datasets generated during the current study are available from the corresponding author on reasonable request.

Conflict of interest: Authors state no conflict of interest.

References

[1] Emotech, "Olly – the first home robot with personality," 2018, https://www.indiegogo.com/projects/olly-the-first-home-robot-with-personality#/ [Accessed: July 7, 2021].

[2] iRobot, "Roomba® e5 Robot Vacuum," 2021, https://shop.irobot.co.uk [Accessed: July 7, 2021].

[3] SoftBank Robotics, "Pepper," 2021, https://www.softbankrobotics.com/emea/en/pepper [Accessed: July 7, 2021].

[4] Mayfield Robotics, "Kuri," 2018, https://www.heykuri.com/explore-kuri/ [Accessed: July 7, 2021].

[5] K. Dautenhahn, "Socially intelligent robots: dimensions of human–robot interaction," *Philos. Trans. R Soc. B Biol. Sci.*, vol. 362, no. 1480, pp. 679–704, 2007.

[6] F. Amirabdollahian, R. op den Akker, S. Bedaf, R. Bormann, H. Draper, V. Evers, et al., "Assistive technology design and development for acceptable robotics companions for ageing years," *Paladyn, J. Behav. Robot.*, vol. 4, no. 2, pp. 94–112, 2013.

[7] J. Saunders, D. Syrdal, N. Burke, K. Koay, and K. Dautenhahn, "'Teach Me – Show Me'–end-user personalisation of a smart

home and companion robot," *IEEE Trans. Hum.-Machine Sys.*, vol. 46, no. 1, pp. 27–40, 2016.

[8] B. Przywara, "Projecting future health care expenditure at European level: drivers, methodology and main results," in *European Economy, European Commision, Economic and Financial Affairs*, 2010.

[9] Eurostats, "Population projections - online database," 2019, https://ec.europa.eu/eurostat/web/population-demography-migration-projections/population-projections-data [Accessed: July 7, 2021].

[10] J. Huang, T. Lau, and M. Cakmak, "Design and evaluation of a rapid programming system for service robots," in *2016 11th ACM/IEEE International Conference on Human–Robot Interaction* (*HRI*), 2016, pp. 295–302.

[11] T. Lourens and E. Barakova, "User-friendly robot environment for creation of social scenarios," in *Foundations on Natural and Artificial Computation*, J. M. Ferrández, J. R. Álvarez Sánchez, F. de la Paz, and F. J. Toledo, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 212–221.

[12] E. Barakova, J. Gillesen, B. Huskens, and T. Lourens, "End-user programming architecture facilitates the uptake of robots in social therapies," *Robot. Autonom. Sys.*, vol. 61, no. 7, pp. 704–713, 2013.

[13] G. Ghiani, M. Manca, F. Paternò, and C. Santoro, "Personalization of context-dependent applications through trigger-action rules," *ACM Trans. Comput.-Hum. Interact.*, vol. 24, no. 2, art. 14, pp. 1–33, 2017.

[14] N. Leonardi, M. Manca, F. Paternò, and C. Santoro, "Trigger-action programming for personalising humanoid robot behaviour," in *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Association for Computing Machinery, New York, NY, USA, 2019, art. 445, pp. 1–13.

[15] F. Paternó and C. Santoro, "End-user development for personalizing applications, things, and robots," *Int. J. Hum.-Comp. Stud.*, vol. 131, pp. 120–130, 2019.

[16] E. Coronado, F. Mastrogiovanni, B. Indurkhya, and G. Venture, "Visual programming environments for end-user development of intelligent and social robots, a systematic review," *J. Comp. Lang.*, vol. 58, art. 100970, 2020.

[17] M. Webster, C. Dixon, M. Fisher, M. Salem, J. Saunders, K. L. Koay, et al., "Formal verification of an autonomous personal robotic assistant," in *Formal Verification and Modeling in Human-Machine Systems: Papers from the AAAI Spring Symposium* (*FVHMS 2014*), 2014.

[18] C. Dixon, M. Webster, J. Saunders, M. Fisher, and K. Dautenhahn, "'The fridge door is open' – temporal verification of a robotic assistant's behaviours," in *Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1–3, 2014. Proceedings*, A. Mistry, M. Leonardis, M. Witkowski, and C. Melhuish, Eds., Cham: Springer International Publishing, 2014, pp. 97–108.

[19] M. Webster, C. Dixon, M. Fisher, M. Salem, J. Saunders, K. L. Koay, et al., "Toward reliable autonomous robotic assistants through formal verification: A case study," *IEEE Trans. Hum.-Machine Sys.*, vol. 46, no. 2, pp. 186–196, 2016.

[20] P. Gainer, C. Dixon, K. Dautenhahn, M. Fisher, U. Hustadt, J. Saunders, et al., "Cruton: Automatic verification of a robotic assistant's behaviours," in *Proceedings of FMICS-AVOCS 2017, vol. 10471 of LNCS*, Springer, 2017, pp. 119–133.

[21] B. Scassellati, L. Boccanfuso, C.-M. Huang, M. Mademtzi, M. Qin, N. Salomons, et al., "Improving social skills in children with asd using a long-term, in-home social robot," *Sci. Robot.*, vol. 3, no. 21, art. eeat7544, 2018.

[22] D. S. Syrdal, K. Dautenhahn, B. Robins, E. Karakosta, and N. C. Jones, "Kaspar in the wild: Experiences from deploying a small humanoid robot in a nursery school for children with autism," *Paladyn, J. Behav. Robot.*, vol. 11, no. 1, pp. 301–326, 2020.

[23] M. M. A. de Graaf, S. BenAllouch, and J. A. G. M. van Dijk, "Long-term evaluation of a social robot in real homes," *Interact. Stud.*, vol. 17, no. 3, pp. 462–491, 2016.

[24] M. De Graaf, S. B. Allouch, and J. Van Diik, "Why do they refuse to use my robot?: Reasons for non-use derived from a long-term home study," in *2017 12th ACM/IEEE International Conference on Human–Robot Interaction* (*HRI*), 2017, pp. 224–233.

[25] N. Hu, G. Englebienne, and B. J. A. Kröse, "Bayesian fusion of ceiling mounted camera and laser range finder on a mobile robot for people detection and localization," in *Proceedings of IROS Workshop: Human Behavior Understanding, Vol. 7559 of Lecture Notes in Computer Science*, 2012, pp. 41–51.

[26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, et al., "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[27] University of Hertfordshire, "Robot House," 2021, https://robothouse.herts.ac.uk/ [Accessed: July 7, 2021].

[28] K. L. Koay, D. S. Syrdal, K. Dautenhahn, and M. L. Walters, "A narrative approach to human–robot interaction prototyping for companion robots," *Paladyn, J. Behav. Robot.*, vol. 11, no. 1, pp. 66–85, 2020.

[29] A. Chanseau, K. Dautenhahn, K. L. Koay, M. L. Walters, G. Lakatos, and M. Salem, "How does peoples' perception of control depend on the criticality of a task performed by a robot," *Paladyn, J. Behav. Robot.*, vol. 10, no. 1, pp. 380–400, 2019.

[30] M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn, "Would you trust a (faulty) robot? Effects of error, task type and personality on human–robot cooperation and trust," in *2015 10th ACM/IEEE International Conference on Human–Robot Interaction* (*HRI*), 2015, pp. 1–8.

[31] D. S. Syrdal, K. Dautenhahn, K. L. Koay, and W. C. Ho, "Integrating constrained experiments in long-term human–robot interaction using task- and scenario-based prototyping," *Inform. Soc.*, vol. 31, no. 3, pp. 265–283, 2015.

[32] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, et al., "Care-O-bot®3 - creating a product vision for service robot applications by integrating design and technology," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1992–1998.

[33] U. Reiser, T. Jacobs, G. Arbeiter, C. Parlitz, and K. Dautenhahn, "Care-O-bot®3-vision of a robot butler," in *Your Virtual Butler: The Making-of*, R. Trappl, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 97–116.

[34] J. Saunders, N. Burke, K. L. Koay, and K. Dautenhahn, "A user friendly robot architecture for re-ablement and co-learning in a sensorised homes," in *Proceedings of the 12th European Conference Advancement Assistive Technology in Europe, (AAATE13)*, 2013.

[35] J. Saunders, M. Salem, and K. Dautenhahn, "Temporal issues in teaching robot behaviours in a knowledge-based sensorised home," in *Proceedings 2nd International Workshop on Adaptive Robotic Ecologies*, Dublin, Ireland, (ARE13), 2013.

[36] T. Henderson and E. Shilcrat, "Logical sensor systems," *J. Robotic Syst.*, vol. 1, pp. 169–193, 1984.

[37] J. E. Laird, *The Soar Cognitive Architecture*, Cambridge, Massachusetts/London, England: The MIT Press, 2012.

[38] I. Georgievski and M. Aiello, "HTN planning," *Artif. Intell.*, vol. 222, no. C, pp. 124–156, 2015.

[39] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*, Mahwah, NJ: Erlbaum, 1998.

[40] M. Fisher, *An Introduction to Practical Formal Methods Using Temporal Logic*, Chichester: Wiley, 2011.

[41] G. Holzmann, "Inspiring applications of Spin," 2018, http://spinroot.com/spin/success.html [Accessed: July 7, 2021].

[42] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, "Formal methods: Practice and experience," *ACM Comput. Surv.*, vol. 41, pp. 19:1–19:36, Oct. 2009.

[43] G. J. Holzmann, "The model checker SPIN," *IEEE Trans. Softw. Eng.*, vol. 23, pp. 279–295, May 1997.

[44] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, et al., "NuSMV version 2: An opensource tool for symbolic model checking," in *Proceedings of the International Conference on Computer-Aided Verification* (CAV), vol. 2404 of LNCS, Springer, 2002.

[45] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proceedings of the 23rd International Conference on Computer Aided Verification* (CAV'11), vol. 6806 of LNCS, Springer, 2011.

[46] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal specification and verification of autonomous robotic systems: A survey," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–41, 2019.

[47] D. Porfirio, A. Sauppé, A. Albarghouthi, and B. Mutlu, "Authoring and verifying human–robot interactions," in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 75–86.

[48] H. Kress-Gazit, K. Eder, G. Hoffman, H. Admoni, B. Argall, R. Ehlers, et al., "Formalizing and guaranteeing* human-robot interaction," June 2006, arXiv:2006.16732 [cs].

[49] M. Kim, K. C. Kang, and H. Lee, "Formal verification of robot movements – a case study on home service robot SHR100," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4739–4744, Apr. 2005.

[50] L. Lestingi and S. Longoni, *HRC-Team: A model-driven approach to formal verification and deployment of collaborative robotic applications*, PhD thesis, Politecnico di Milano, 2017.

[51] A. G. Billard, S. Calinon, and R. Dillman, "Learning from humans," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Cham: Springer, 2016, pp. 1995–2014.

[52] M. Mara, M. Appel, H. Ogawa, C. Lindinger, E. Ogawa, H. Ishiguro, et al., "Tell me your story, robot: introducing an android as fiction character leads to higher perceived usefulness and adoption intention," in *Proceedings of the 8th ACM/*

[53] A. Rosenthal-von der Pütten, C. Stramann, and M. Mara, "A long time ago in a galaxy far, far away... the effects of narration and appearance on the perception of robots," in *26th IEEE International Symposium on Robot and Human Interactive Communication* (RO-MAN), Piscataway, NJ: IEEE Press, 2017, pp. 1169–1174.

[54] P. Bucci, L. Zhang, X. L. Cang, and K. E. MacLean, "Is it happy? behavioural and narrative frame complexity impact perceptions of a simple furry robot's emotions," in *CHI '18: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, P. Bucci, L. Zhang, X. L. Cang, and K. E. MacLean, Eds., Association for Computing Machinery, New York, NY, USA, 2018, pp. 1–11.

[55] J. Banks, "Optimus primed: Media cultivation of robot mental models and social judgments," *Front. Robot. AI*, vol. 7, art. 62, 2020.

[56] C. Dindler and O. S. Iversen, "Fictional inquiry-design collaboration in a shared narrative space," *CoDesign*, vol. 3, no. 4, pp. 213–234, 2007.

[57] D. S. Syrdal, K. Dautenhahn, K. L. Koay, and W. C. Ho, "Views from within a narrative: Evaluating long-term human–robot interaction in a naturalistic environment using open-ended scenarios," *Cognit. Comput.*, vol. 6, no. 4, pp. 741–759, 2014.

[58] H. Rex Hartson, "Human–computer interaction: Interdisciplinary roots and trends," *J. Sys. Softw.*, vol. 43, no. 2, pp. 103–118, 1998.

[59] A. Cooper, *"The Inmates are Running the Asylum,* Indianapoli, IN, USA: Macmillan Publishing Co., Inc., 1999.

[60] I. Duque, K. Dautenhahn, K. L. Koay, L. Willcock, and B. Christianson, "A different approach of using personas in human–robot interaction: Integrating personas as computational models to modify robot companions' behaviour," in *2013 IEEE RO-MAN*, 2013, pp. 424–429.

[61] T. F. dos Santos, D. G. de Castro, A. A. Masiero, and P. T. Aquino Junior, "Behavioral persona for human–robot interaction: a study based on pet robot," in *Human-Computer Interaction. Advanced Interaction Modalities and Techniques, HCI 2014*, M. Kurosu, Ed., Lecture Notes in Computer Science, vol. 8511, Springer, Cham, 2014, pp. 687–696.

[62] J. Brooke, "SUS: a retrospective," *Journal of Usability Studies*, vol. 8, no. 2, pp. 29–40, 2013.

[63] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, United States: Elsevier Science, 2013.

[64] R. Rosenthal, "Parametric measures of effect size," in *The Handbook of Research Synthesis*, H. Cooper and L. Hedges, Eds., New York: Russell Sage Foundation, 1994, chapter 16, pp. 231–244.

[65] F. Corno, L. De Russis, and A. Monge Roffarello, "Empowering end users in debugging trigger-action rules," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–13.

[66] J. Good and K. Howland, "Programming language, natural language? Supporting the diverse computational activities of novice programmers," *J. Visual Lang. Comput.*, vol. 39, pp. 78–92, 2017.