New Developments for ROBERT

Assisting Novice Users Even Better in DIY Projects

Gregor Behnke[1,2], Pascal Bercher[1,3], Matthias Kraus[1], Marvin Schiller[4], Kristof Mickeleit[1], Timo Häge[1], Michael Dorna[4], Michael Dambier[4], Dietrich Manstetten[4], Wolfgang Minker[1], Birte Glimm[1], Susanne Biundo[1]

[1]Ulm University, Germany, firstname.lastname@uni-ulm.de
[2]University of Freiburg, Germany, behnkeg@cs.uni-freiburg.de
[3]The Australian National University, Australia, pascal.bercher@anu.edu.au
[4]Corporate Research Sector, Robert Bosch GmbH, Germany, firstname.lastname@de.bosch.com

Motivation

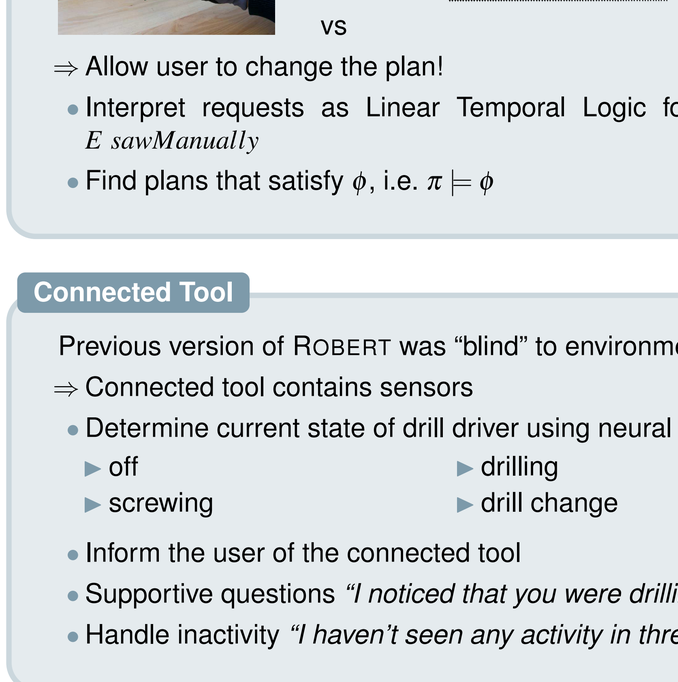
- Do-It-Yourself projects can be hard for novices – due to lack of knowledge, fear, or inexperience.
- ROBERT helps them to complete projects, allowing them to successfully perform other DIY projects in the future.
- ROBERT guides its user with a step-by-step instruction through the DIY project. The instructions are generated by a planner – each action is one step in the instruction.

Abilities

- HTN Planning – generates plan and abstraction for orientation
- Ontology Manager – stores background knowledge, selects appropriate texts, images, and videos for each action
- Dialogue Manager – controls dialogue between user and system, sends request of the user to appropriate component

We added two new abilities to ROBERT

- Changing plans if the user requests
- Proactivity using connected tools

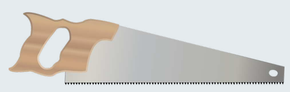







Changing Plans

Users might have preferences which we can't elicit before using ROBERT, but planner has to choose one alternative.


vs


⇒ Allow user to change the plan!

- Interpret requests as Linear Temporal Logic formulae e.g. $\phi = G\neg sawElectric \wedge E\ sawManually$
- Find plans that satisfy ϕ, i.e. $\pi \models \phi$

Connected Tool

Previous version of ROBERT was "blind" to environment.

⇒ Connected tool contains sensors

- Determine current state of drill driver using neural network
  - ▶ off
  - ▶ drilling
  - ▶ battery change
  - ▶ screwing
  - ▶ drill change
  - ▶ other
- Inform the user of the connected tool
- Supportive questions *"I noticed that you were drilling. Did that work?"*
- Handle inactivity *"I haven't seen any activity in three minutes. Do you need help?"*

```
(:action Drill_Screw
 :parameters (?drill - Drill
              ?o1 - Connectable ?o2 - Connectable
                          ?screw - Screw)
 :precondition (and
  (usable ?screw) (usable ?o1) (usable ?o2)
  (imply (typeOf ?o1 HomObj) (fixated ?o1))
  (imply (typeOf ?o2 HomObj) (fixated ?o2))
  (exists (?b - Battery) (and (AttachedBattery ?drill ?b) (hasEnergy ?b)))

  (exists (
   ?sb - ScrewBit
   ?sbh - ScrewBitHolder
   ?screwType ?screwBitType - Type
   ?rpm - Number
   ?ds - DrillSettings
  ) (and
   (AttachedShank ?drill ?sbh) (AttachedShank ?sbh ?sb)
   (typeOf ?sb ?screwBitType)
   (typeOf ?screw ?screwType)
   (Drill_settings ?drill ?ds)
   (DrillSettings_direction ?ds right)
   (DrillSettings_rotarySpeed ?ds ?rpm)
   (exists (?sc - ScrewingConfig) (and
    (ScrewingConfig_screwType ?sc ?screwType)
    (ScrewingConfig_screwBitType ?sc ?screwBitType)
   ))
   (exists (?scc - ScrewConnectionConfig ?ot1 ?ot2 - Type) (and
    (typeOf ?o1 ?ot1)
    (typeOf ?o2 ?ot2)

    (ScrewConnectionConfig_screwType ?scc ?screwType2)
    (ScrewConnectionConfig_materialType1 ?scc ?ot1)
    (ScrewConnectionConfig_materialType2 ?scc ?ot2)
    (ScrewConnectionConfig_rotarySpeed ?scc ?rpm)

    (forall (?hs1 - HoleShape) (and
     (imply (ScrewConnectionConfig_holeShape1 ?scc ?hs1) (holeShape ?o1 ?hs1))
    ))
    (forall (?hs2 - HoleShape) (and
     (imply (ScrewConnectionConfig_holeShape2 ?scc ?hs2) (holeShape ?o2 ?hs2))
    ))
   ))
  ))
 )
 :effect (and
  (not (usable ?screw))
  (connected ?o1 ?o2 ?screw)
 )
)
```