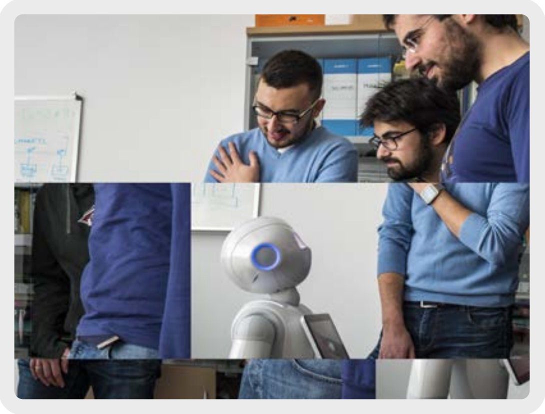
Planning for Human-Agent collaboration using Social Practices

Tim Miller, *University of Melbourne, Australia*
Virginia Dignum, *TU Delft, NL*
Frank Dignum, *Utrecht University, NL*

Can Robots be“social”?



Social Interaction with Artificial Systems

- The ability to exhibit social behaviour is paramount for collaboration.
- Human – Agent (- Robot) interaction:
  - Healthcare robots, intelligent vehicles, virtual coaches, serious game characters…

- social intelligent systems:
  - behaviour can be interpreted by other systems as the behaviour of perceiving, thinking, moral, intentional, and behaving selves; i.e. as individuals
  - can consider the intentional or rational meaning of others' field of expression, and that can form expectations about the others' acts and actions

- Interaction with humans
  - Account for a myriad of possible ways of acting
  - Account for the social expectations concerning collaboration

- Usually task and domain specific social behaviours are built into robots.

- Research on intelligent robots usually focuses first on making robots cognitive by equipping them with planning, reasoning, navigation, manipulation and other related skills necessary to interact with and operate in the non-social environment, and then later adding ‘social skills’ and other aspects of social cognition.

(Gal Kaminka, Curing robot autism, 2013)

Challenge:
building social intelligence blocks

Social skills are not a simple ‘add-on’ to human–agent interfaces

The behavior of the robot should be realized so that it can be adaptable to unexpected human reactions.

and that’s where we think social practices could be helpful

Gal Kaminka, Curing robot autism, 2013
Frank Dignum, From autistic to social agents, 2014

Social Practices (Reckwitz, 2002)

- A ‘**practice**’
  - is a routinized type of behaviour which consists of several interconnected elements
  - describes physical and social patterns of joint action as routinely performed in society and provide expectations about the course of events and the roles that are played in the practice
  - elements of a ‘practice’ are: Materials, Meanings, Activities

=> A practice is not a rigid schema but a sort of generalizable procedure in a particular context

SP model for « computer scientist » (Dignum, 2015)

Social Practice

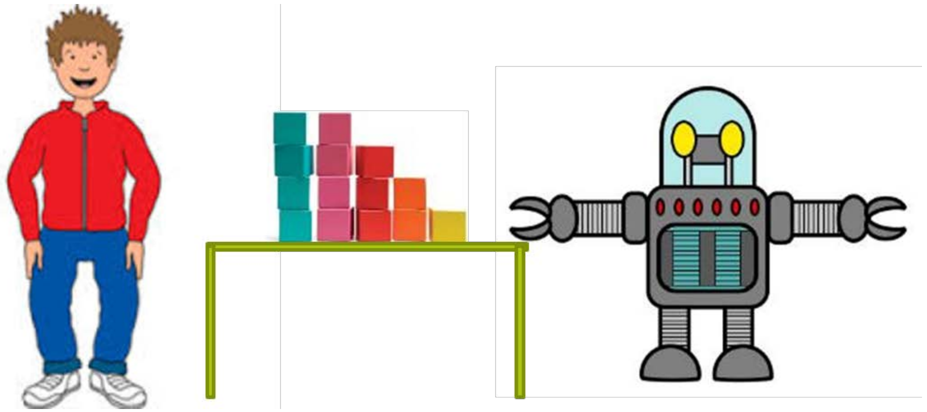
Context	Activities	Meanings	Expectations
Actors Roles Ressources Positions	Basic Actions Capabilities General Preconditions	Purpose Promote Counts-as	Plan patterns Norms Triggers Start condition Duration

4 groups of concepts that play a role in the social practice

An illustrative example

- A human and a robot have the goal to build a pile with 4 cubes and put a triangle at the top.
- One after the other, they should stack bricks in the expected order.
- Each agent has a number of cubes accessible in front of him and would participate to the task by placing its cubes on the pile.
- At the end, one of the agent should place a triangle at the top of the pile.

- Available actions
  - *pickup*: pick up block;
  - *stack*: put block in top of tower;
  - *place*: put block on the table;
  - *give*: give block to the other actor;
  - *stabilize*: support tower such that the other actor can stack block;
  - *request*: ask other actor to perform action.



Planning

- Using Muise's et al.'s ***first-person multi-agent planning*** (FPMAP)

$$\langle Ag, F, I, \{A_i\}_{i=1}^{|Ag|}, \{G_i\}_{i=1}^{|Ag|} \rangle$$

- *Ag* defines a set of agents,
- *F* defines a set of fluents or propositions,
- A_i is a set of actions for each agent i,
  - (1) $pre(a) \subseteq F$ describes the fluents that need to hold for a to be executed;
  - (2) $add(a) \subseteq F$ describes the fluents that will become true if a is executed;
  - (3) $del(a) \subseteq F$ describes the set of fluents that will become false if a is executed;
  - (4) $cost(a) > 0$ is the cost of executing a.
- $I \in F$ is the initial state,
- $G_i \subseteq F$ characterises the goal for each agent i.

- A solution for an FP-MAP is a policy — a mapping from (partial) states to actions — for a single agent *i*, rather than a policy that orchestrates a set of agents

Social Practices

Context	
	Actors
	Roles
	Resources
	Positions
Activities	
	Basic actions
	Capabilities
	Preconditions
Meanings	
	Purpose
	Promote
	Counts-as
Expectactions	
	Plan pattern
	Norms
	Triggers
	Start Condition
	Duration

Social Practices

Context	
Actors	Robot, Human
Roles	Stacker
Resources	blocks, pyramids, table
Positions	Position in space of resources and actors
Activities	
Basic actions	*pickup*, *stack*, *place*, *give*, *stabilize*, *request*
Capabilities	The set of actions an actor is capable of performing.
Preconditions	All actors are at the table with blocks.
Meanings	
Purpose	Intended result of an action, E.g. *place(block)* has the purpose to increase the stack size, but it could lead to the whole stack falling
Promote	Social values promoted by an action, E.g. waiting for your turn promotes cooperation.
Expectations	
Counts-as	Executing an action is seen as another action or aim E.g. putting the pyramid on a block counts-as ending the scenario
Norms	E.g. the robot is forbidden to place the pyramid
Plan pattern	Landmarks (goal states) for each part of the interaction E.g. pickup(b); place(b);.... Place(p)
Start Condition	
Duration	

FPMAP ??

Planning for social interaction

- **Definition 1: Normative Action**. A normative action is a standard action, except it has a normative proposition ϕ, which specifies the norm condition, and a violation cost $\omega > 0$, which defines the cost of violating the norm.
  - Actions that violate norms have a higher cost than those that do not.
  - The planning problem is then simply a standard cost-minimising problem.

- Planning with normative actions.

  $\langle F, A, I, G_i \rangle$ with normative actions in $A_{norm} \subseteq A$
- replace each action $a \in A_{norm}$ with a'_{norm} and a'_{viol} such that:

– $pre(a'_{norm}) = pre(a) \wedge \phi$
– $cost(a'_{norm}) = cost(a)$
– $del(a'_{norm}) = del(a), add(a'_{norm}) = add(a)$

and

– $pre(a'_{viol}) = pre(a) \wedge \neg\phi$
– $cost(a'_{viol}) = cost(a) + \omega$
– $del(a'_{viol}) = del(a), add(a'_{viol}) = add(a)$

Normative Actions

- Turn taking: actors act one at the time
  - Meaning: Prevents conflicts
  - Promoted value: Politeness

  - Encoding
    - Ordering fluents (actor a) (next a b) (next b a)
    - Normative constraint that the actor ?a satisfies (actor ?a), penalising if not

- Finishing touch: the human will place the pyramid in top of the stack.
  - Promoted value: Achievement

  - Encoding

    Normative constraints:
    - (1) the block being stacked is the last block (a pyramid)
    - (2) the agent stacking it is the human collaborator.

Landmarks

- Landmark Plan Pattern. A landmark plan pattern consists of a landmark condition lc (a proposition), and three disjoint sets of actions $A_{pre} \cup A_{post} \cup A_{both} = A$, where the set A_{pre} represents the set of actions that can only occur before the landmark is reached, A_{post} the set of actions that can only occur after the landmark is reached, and A_{both} represents actions that can occur both before and after.
- Given a landmark lc and planning model <F, A, I, G> we produce a new planning
- model <F', A', I', G> in which

  - $F' = F \cup \{L\}$, in which L is a single token fluent representing whether the landmark has occurred.
  - $A' = A'_{pre} \cup A'_{post} \cup A_{both} \cup \{assess\_L, assess\_not\_L\}$, in which:
    For each $a \in A_{pre}$, there exists an action $a' \in A'_{pre}$ such that:
    - $pre(a') = pre(a) \wedge \neg L$
    - $add(a') = add(a)$, $del(a') = del(a)$

    For each $a \in A_{post}$, there exists an action $a' \in A'_{pre}$ such that:
    - $pre(a') = pre(a) \wedge L$
    - $add(a') = add(a)$, $del(a') = del(a)$

    There is a new action $asses\_L$ that 'flips' the landmark token L from false to true:
    - $pre(assess\_L) = lc$ (note, lc is the proposition, not the token)
    - $add(assess\_L) = L$
    - $del(assess\_L) = \neg L$

    There is a new action $assess\_not\_L$ that 'flips' the landmark token L from true to false (thus allowing landmarks to be iterated within a single plan):

<F', A', I', G>

- $F' = F \cup \{L\}$, in which L is a single token fluent representing whether the landmark has occurred.
- $A' = A'_{pre} \cup A'_{post} \cup A_{both} \cup \{assess\_L, assess\_not\_L\}$, in which:
  For each $a \in A_{pre}$, there exists an action $a' \in A'_{pre}$ such that:
  - $pre(a') = pre(a) \wedge \neg L$
  - $add(a') = add(a)$, $del(a') = del(a)$

  For each $a \in A_{post}$, there exists an action $a' \in A'_{pre}$ such that:
  - $pre(a') = pre(a) \wedge L$
  - $add(a') = add(a)$, $del(a') = del(a)$

  There is a new action $asses\_L$ that 'flips' the landmark token L from false to true:
  - $pre(assess\_L) = lc$ (note, lc is the proposition, not the token)
  - $add(assess\_L) = L$
  - $del(assess\_L) = \neg L$

  There is a new action $assess\_not\_L$ that 'flips' the landmark token L from true to false (thus allowing landmarks to be iterated within a single plan):
  - $pre(assess\_not\_L) = \neg lc$
  - $add(assess\_not\_L) = \neg L$
  - $del(assess\_not\_L) = L$
- $I' = I \wedge \neg L$

Landmark: unstacking blocks

- Social practice of first unstack all the currently stacked blocks, and then start to re-stacking blocks to achieve the goal.
- While this may be suboptimal - some stacked blocks may be part of the goal - it is the type of simplification that humans make to simplify planning.

- Encoding
  - landmark *unstacked* $\in$ F.
  - The action of picking up a block from on top of another block contains the precondition $\neg$*unstacked*.
  - The actions of stacking, picking up from the table, and putting on top of another block have the precondition *unstacked*.
  - a new action (with no actor) is added, called *assess unstack*, which has the precondition:

$$\neg unstacked \wedge \forall\ ?b \in block \cdot (clear\ ?b)$$

Experiments

- Using planner MA-PRP and planning language PDDL
  - Muise, et al: Planning for a single agent in a multi-agent environment using FOND, IJCAI 2016

- Scenarios
  1. Turn-taking (with finishing touch)
  2. Landmarks: as 1 with Unstacking
  3. Baseline: no social practices, the agent plans for every contingency

Results – planning time

In turn-taking, at each choice point, branching factor is reduced by a factor of $|A_g|$
Unstaking also simplifies planning

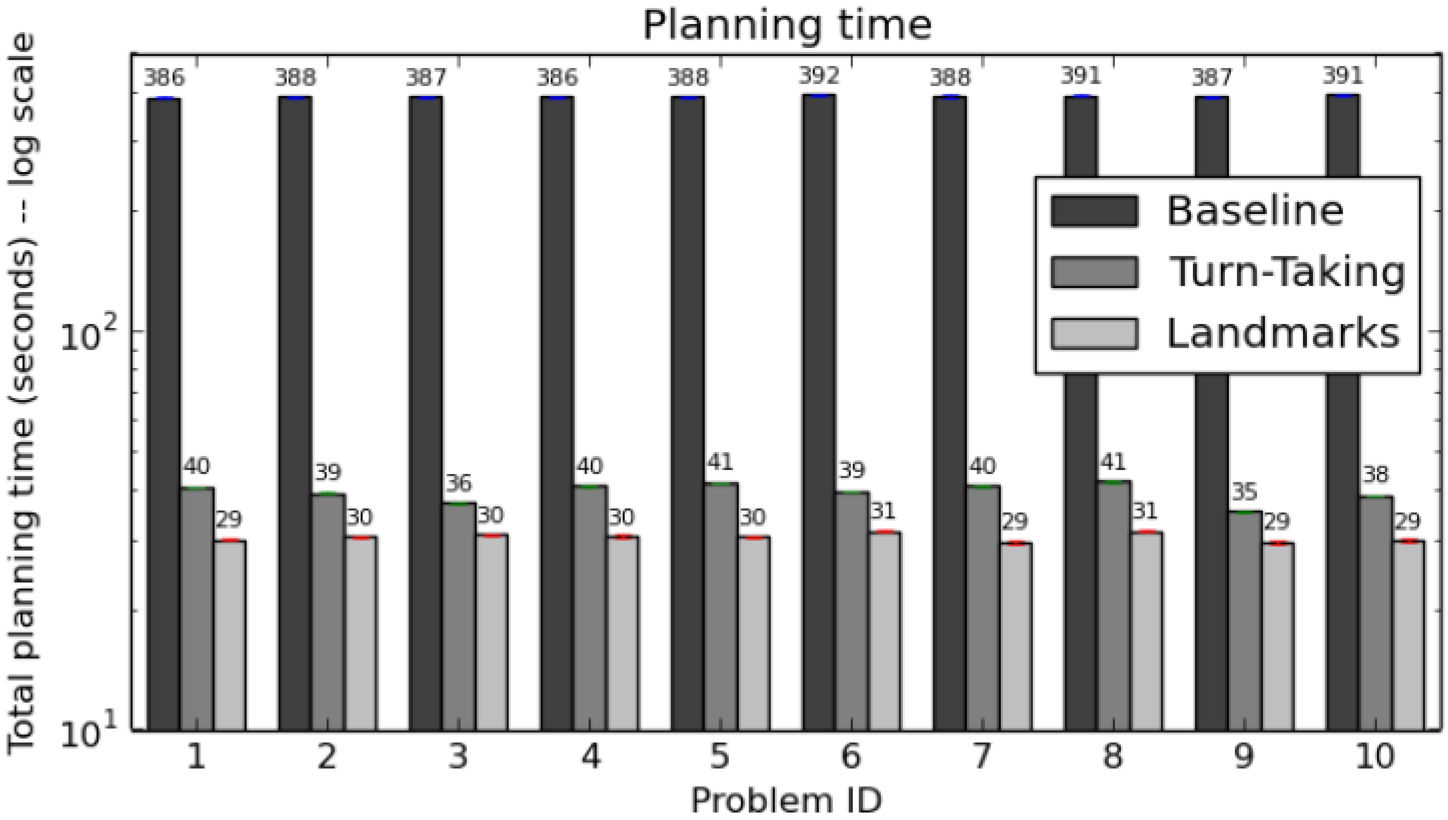


Fig. 1: Average planning time (in seconds)

Results – policy size

The optimal solution is often not to unstack blocks, because they are already stacked in the desired position.

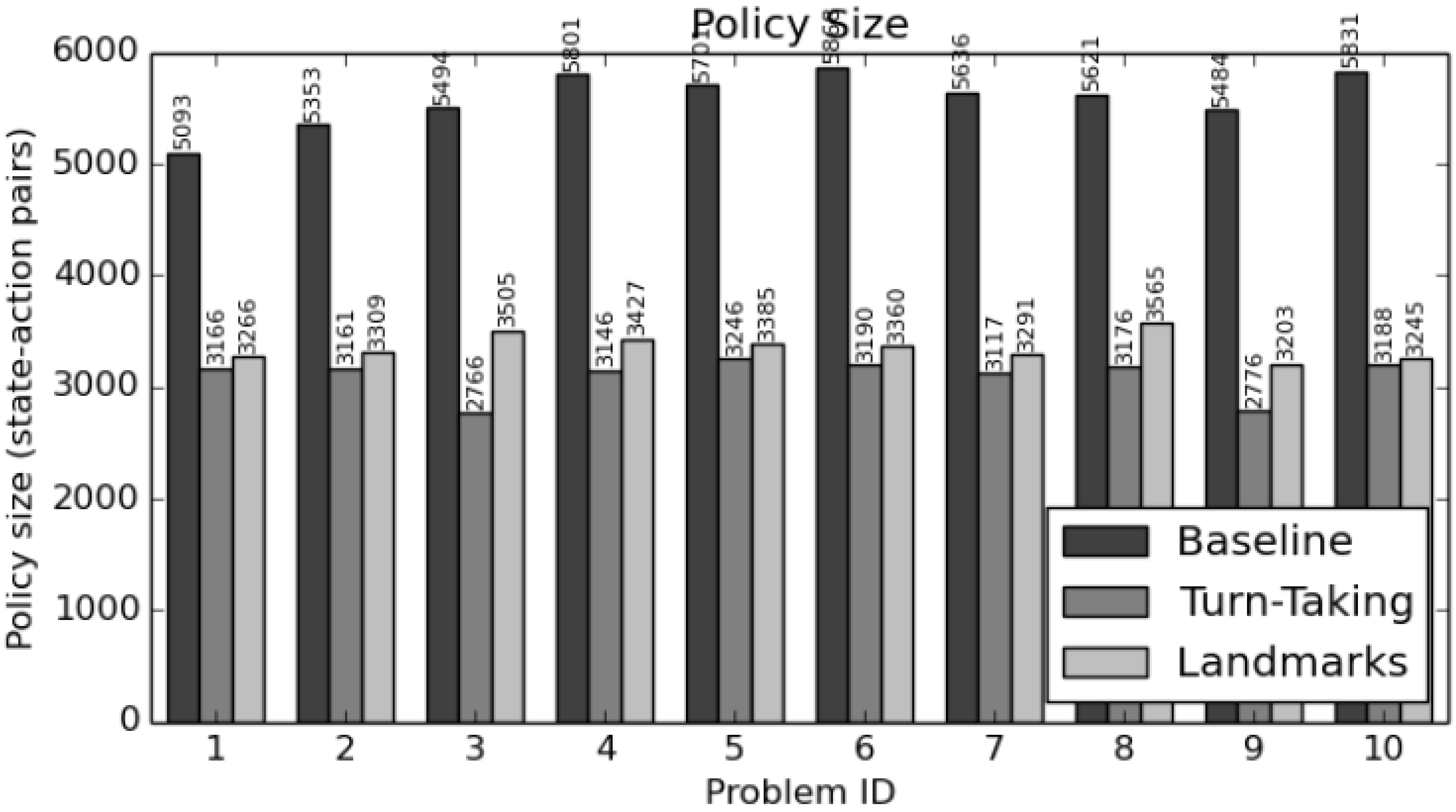


Fig. 2: Policy size

Results – plan size

Baseline and turn-taking exploit the fact that some blocks may be already stacked in the desired position.

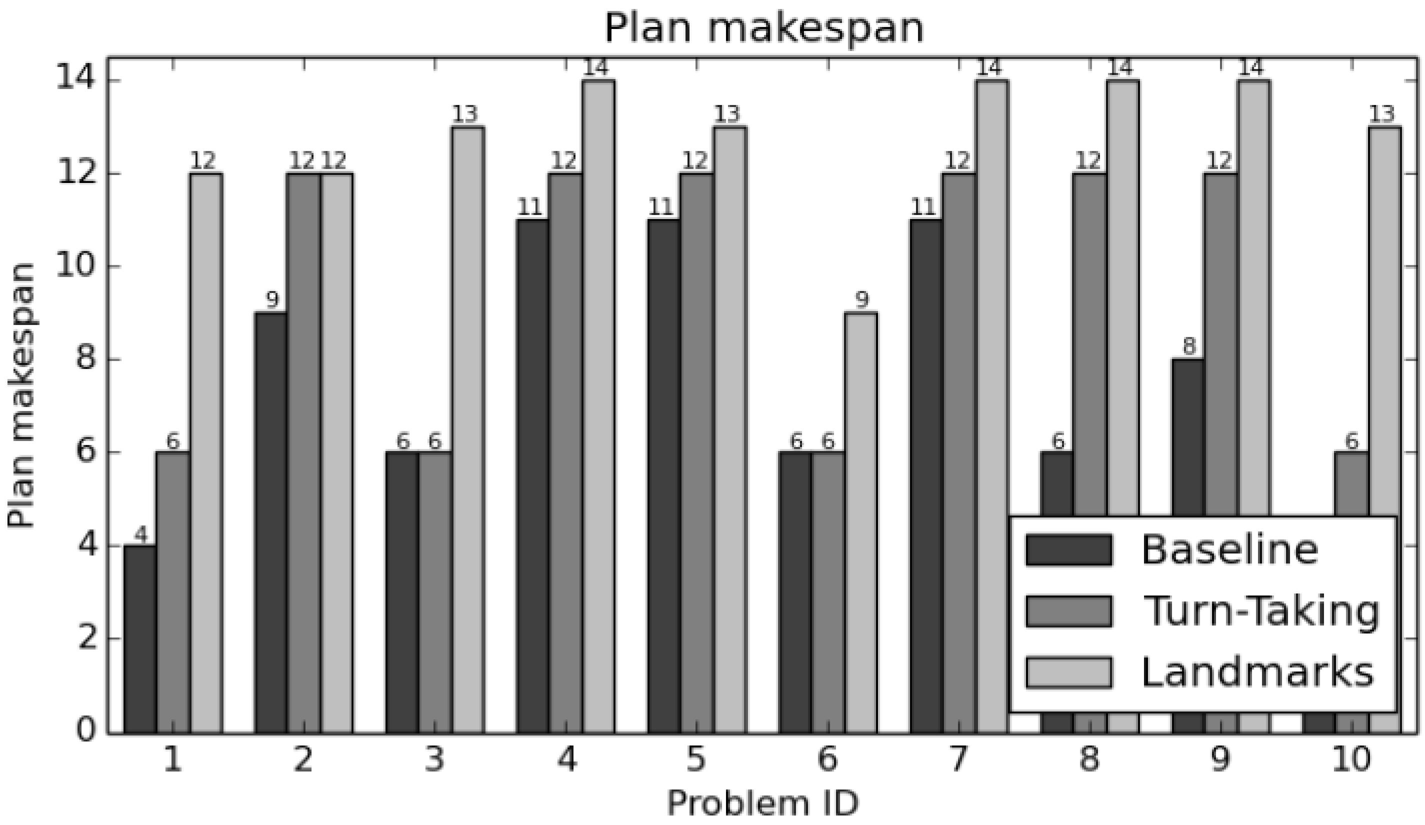


Fig. 3: Plan size

Conclusions

- Social practices define expectations and constraints that can be used in a consistent way.
  - support predictability and directability in interactions.
- Human-agent/robot interaction: social practices describe natural interactions for humans.
- Social practices indeed lead to quicker planning and re-planning.
  - plans can sometimes be longer than optimal plans.
  - nut more robust and following common patterns of interactions of humans, thus increasing acceptability.
- In future work
  - richer scenarios and richer encodings of social practice.
  - human behavioural experiments to demonstrate the benefits of social practice in human-agent interaction.
  - automatically integrate a description of a social practice as input for the planner and also use it to direct the execution and possible re-planning.
  - enough for 10 years of issues in SCS journal ☺

Conclusion

Using social practices for human-robot interaction will lead to the following advantages:

1. The robot will be able to monitor its environment on the basis of the expected events of the social practice; It can filter its perceptions for those elements that are meaningful for a particular social context
2. The social practice can speed up robots behaviour-selection by promoting the behavioural patterns that are best suited for that social context;
3. The interactions will be more robust, because the robot has an explicit context which it can use to recover from failures and unexpected events.
4. The social practice provides a basis for explanation of the robots behaviour
5. From the users perspective, social-aware robots will be perceived as more socially realistic.

Conclusions

- Successful joint action depends on both physical aspects of the domain, but also, and primarily, on its social characteristics.
- We presented a check-list proposed to support the identification of the aims and steps of the joint action
- Social practices can support agent planning and deliberation in different contexts
- Future work: enough issues for 10 years of research in all kinds of aspects (theoretical, engineering,…)