Robust Collaboration: Enriching Decisions with Abstract Preferences

Loïs Vanhée[1,2], Frank Dignum[1], and Jacques Ferber[2]

[1] Utrecht Universiteit, The Netherlands
[2] LIRMM, University of Montpellier II, France

Abstract. Aspects of human societies provide a rich source of inspiration for influencing individual and social behaviors in order to achieve collaboration in a MAS. This article particularly investigates how human cultures and particularly human values can be used as an inspiration for achieving collaboration. Indeed, human values abstractly set what individuals consider as important, driving them towards similar individual and social outcomes, helping them to work together. We want to reproduce the same type of behaviors in MASs, even if we do not aim at faithfully reproducing human behavior.

Preferences are used for modeling values. But, specifically for values, preference functions order abstract yet driving criteria (e.g. "security vs. freedom" instead of "blue vs. red"). Values support abstract decisions, which drive agents to make local decisions that support some coherence at the collective level.

We show that integrating values as a design constraint have many benefits for designing collaborative MAS. In particular, they offer greater flexibility and robustness to the system. Furthermore, values provide a top-down perspective for designing MASs which can be combined with traditional methods (e.g. norms, organizations) for lowering overall design complexity.

Keywords: Agent Oriented Software Engineering, Methodology, Collaboration, Values, Preferences.

1 Introduction

"The firefighter agent is about to enter in the burning house in order to extinguish the fire and rescue victims. Should it immediately enter the house or spend precious seconds in order to first double check that tasks of other colleagues that support the agent's entrance have been completed?"

Current methods for supporting collaboration have troubles for solving such a dilemma, particularly in complex environments (e.g. quick evolution, adversarial agents, numerous interactions between environmental variables, partly visible dynamics, many possible contexts). Current methods for supporting collaboration specify decisions to be taken for concrete and expected choices. But by definition of complex environments, these choices are numerous and there is rarely a single simple rule which determines the best answer for any situation.

F. Dalpiaz et al. (Eds.): EMAS 2014, LNAI 8758, pp. 406–430, 2014.

How many norms have to be created in order to cope with those dilemmas? How large should a protocol be? With cooperation, how many details have to be considered before performing any action? Without mentioning the difficulty, as a system designer, to predict all those possible dilemmas which may lead to collaboration failures.

Looking at humans, they generally manage to cope quite well with such a dilemma. Maybe we can get some inspiration from their reasonings. Our goal consists in building models inspired by the way human solve some problems with the aim of transposing this solution to their artificial counterparts. Plenty of former methods use similar inspirations for designing agents and improving collaboration (e.g. BDI, norms, organizations, protocols). In human societies, norms and protocols[1], despite being particularly extensive in the domain of fire fighting, do not specify how to resolve the dilemma presented in introduction. Norms and protocols are used in human societies for coping with limited and well-expected technical issues (e.g. sensing pain in the hand while watering means electrical hazard. Change watering to "spread" mode in order to limit conductivity; techniques for manipulating the water hose). But, the relative success of human societies given this lack of formal control suggests that humans have other mechanisms for both to making decisions and creating expectations about others. But which ones?

As a possible answer, we propose to investigate at cultures. Cultures can be seen as a set of shared mental attitudes which exist within a society or a group. These mental attitudes have many influences: they can range from *values* which are big abstract principles about how to behave in life (e.g. timeliness, relationship with authority) to *practices* which are more local and concrete (e.g. greeting protocols). Those influences tend to support each other (e.g. if timeliness is important, concrete rules tend to support timeliness). As a rule of thumb, humans tend to use the most concrete rule available when making decisions and rely on values when no rule is available, for more exceptional decisions.

Back to our dilemma from that human perspective, consider that our team of firefighters has a culture which favors a value of timeliness. Assume also that for that specific dilemma, no practice or rule explicitly states how individuals should behave. In such a situation, individuals investigate their values and know that "timeliness" is an important value for the group. In other words, individuals are culturally willing to sacrifice local utility for being in time, considering for instance that group success is more important than individual success. In that case, any agent supporting the firefighter would do his or her best in order to be on time, possibly sacrificing some local utility (e.g. preferring to delay rescuing a victim for making sure that the water hose is operational on time). Thus, culture *influences decisions* of individuals. In addition, cultures also influence *expectations* that individuals can create about others. For instance, the firefighter agent which is about to enter the burning house, knows that timeliness is important. Thus, if he or she do the best to be on time and assumes that the work of

[1] See some documentations for training firefighters at `http://www.udsp34.org/index.php?p=pres&Ctt_Doc_Categorie=4`

others is done. So the firefighter can accurately decide to enter without having to double-check.

Back to the MAS-design perspective, we propose to inspire from cultures in order to solve MAS problems[2]. Practices are already well-handled by existing literature for promoting collaboration (e.g. norms, protocols). To that extent, practices are not further investigated in this article, even if they can be related with techniques for achieving collaboration which are further investigated in this article. Contrarily, values, which importance for driving collaboration has just been illustrated, are relatively new to the design of methods for supporting collaboration. Values are abstract but broadly influential. They offer principled and justifiable answers to dilemmas that agents can encounter. Values drive individuals and societies towards environmental or social outcomes which are culturally preferable without strongly constraining decisions. Values provide offer another possibility for system designers to drive agents, since system designers determine what agents culturally prefer.

Values are complementary with existing solutions for achieving collaboration. Indeed, values support abstract complex decisions that agents have to make (e.g. buying a house). Nevertheless, they are inappropriate for driving simple and more standardized decisions that agents have to make (e.g. moving to the house). Such decisions are more appropriately handled by practices. Thus, values are not an alternative but an additional method for achieving collaboration. Values cope with some problems which are hard to handle with more concrete approaches, while these latter approaches cope with problems hard to handle with values.

From a modeling perspective, we propose to model value systems in using preferences. Indeed, the core property of a value system drive what consider as important by determining the relative importance of their values. Nevertheless, human values are not any preference (e.g. preferring red over blue). Human values encompass abstract aspects which can be related to many decisions (e.g. timeliness, respect of authority). This article provides some principles for determining what values are to be integrated within a value systems and their possible influence of agents.

More technically, value systems are abstract preferences, which raise some technical challenges for designing agents. Indeed, they are inadequate for managing concrete agents decision, because values are abstract (e.g. no need to reason about one's values to determine which foot to start walking with). For those particular decisions, more appropriate tools should be used, such as BDI agents or protocols. Nevertheless, in order to avoid conflicting specifications, these concrete behaviors should support and thus be related with the value system of an agent. This article proposes solutions for bridging the gap in terms of abstraction between abstract preferences to concrete action.

[2] *Disclaimer*: this article aims at providing a solution for engineering MASs. Values are just inspirations, we do not aim at faithfully replicating their influence on behavior but instead at finding in which context they are useful. We prefer "incredible" working solutions than credible human-like failures.

The content of this paper is organized as follows. Section 2 describes a running example illustrating our concepts throughout the article. Section 3 describes the related work. Section 4 describes solutions for integrating preferences in agent decision processes. Section 5 describes the use of shared preferences for achieving collaboration. Section 6 describes examples of human cultures that can be used as inspiration for designing shared preference. The main contributions of this article correspond to the content of Section 4 and Section 5.

2 Running Example

A running example is used in order to better illustrate concepts and methods described throughout this article.

Consider a MAS supporting a team of fire fighters. Each fire fighter has his own agent. Each agent keeps track of the information of the fire fighter's situation and can confer with the other agents about which information or action advise to give to its fire fighter. We may also assume that agents can be involved within the system for supporting humans. In the following we identify the agents with the persons they support for ease of reference. The mission (or goal) of the agents consists in extinguishing fires and rescuing people who got injured due to the crisis. In addition to fire fighters, a special agent called the "fire commander" (represented by a_{fc}) located in the firetruck can communicate with the fire fighters using point to point communication.

In this setting, an agent (indicated by a_1) is about to make a decision. The situation of a_1 is as follows: *The fire commander planned for me. I have to be at the fire place at time T. There, I will support agent a_2 for extinguishing the fire. While moving to the fire, I spotted a person nearby.*

a_1 has to chose between three available options:

1. *Rescue*: a_1 delays its action to move towards the fire and rescues the victim instead. The time required to rescue the victim is unpredictable: if the victim is healthy, the action can be very quick (ask the victim to leave), if the victim is injured this action can take much longer (the agent has to stabilize and to carry the victim out of the danger zone). The fire fighter regulations state that a_1 is forbidden to leave a victim if a victim is injured.
2. *Report*: a_1 delays its action to move towards the fire and warns the fire commander about the presence of a person. This action takes some time but is quicker than helping.
3. *Ignore*: the agent stores the information that a person has been spotted and keeps moving towards the fire.

If a_1 has some available time before T, the situation is referred to as d_t. Otherwise (a_1 is short in time), the situation is referred to as $d_{\bar{t}}$.

A warning has to be issued before going further. In order to be easily understandable, we keep that example simple. But simple examples are easily coped by other methods (e.g. a single norm can state: "obliged to report spotted victims"). Nevertheless, we aim at considering complex and dynamic environments.

To that extent, when considering this running example, consider that the decision to be made can be done in many different contexts (e.g. in an isolated house, in a skyscraper, in a warzone). In that case, methods for designing simple constraints require much more design efforts. Indeed, these methods are more appropriate for driving behavior in well-expected scenarii but at less adequate when the context have multiple influences on many decisions (e.g. need three norms just depending on the location: rescuing victims in an isolated house; rushing to the fire in a skyscraper; determining origins of casualties in warzones). Contrarily, we aim at showing that such a complex problem is better handled by values. While this example is purposefully very simple and specific in order to be understandable, desirable solutions for this outcome are expected to be adaptable for a wide variety of contexts.

3 Previous Work

This section introduces two categories of related work. First, Section 3.1 presents existing work for achieving collaboration. This work is encompasses cultural practices that we do not further model in this article. Furthermore, presenting this previous work allows to better display how our work contributes to that field. Second, Section 3.2 presents existing frameworks for modeling preferences which are used in order to model our values.

3.1 Driving Collaboration

Former research in MAS extensively investigated the design of multi-agent solutions for reaching system goals via the collective action of individual agents [10,13,15,27]. Indeed, these methods are particularly useful for solving collective problems or for supporting interactions of self-interested agents, which are the main practical applications of MASs. Since this article aims at driving collective action, these methods require to be introduced. They are introduced from a decreasing order of the influence from system designers on collective action.

Determining which method should be selected highly depends on the problem complexity. As a general rule, the more system designers restrict behaviors of individuals, the more system designers can drive collective action, the more complex is the task of the system designer. To that extent, methods which give the most influence to system designers are also the ones which are the most limited by environmental complexity. In addition, those methods allow to reach the highest efficiency but their closeness to environmental constraints tend to limit their flexibility and robustness.

Methods. From an extreme perspective, system designers can be totalitarian by completely restricting agent behavior. The main frameworks of this perspective can be related to MDP-like approaches (DEC-MDP, DEC-POMDP, POSG [3]) for collaborative agents and Game Theory [5] for selfish agents.

Some methods give further methods by only partly constraining agent behaviors. More pragmatically, these methods aim at being just enough constraining for making sure that agents cannot go against desirable collective outcomes. Some of them are inspired by natural social systems [1], stygmergy [23] or human societies (e.g. norms, organizations, protocols). Norms [6] are rules which forbid collectively harmful behaviors. Organizations [12] allocate roles and create obligations between individuals. Protocols [14] standardize patterns of interactions.

Some other methods offer even further freedom to agents. These methods rely on abstract rules. These abstract rules do not aim at tightly enforcing concrete collective action (e.g. be at the meeting point at time T). Instead they aim at providing abstract and general rules to agent (e.g. forbidden to be late). Agents are free enough to circumvent the rules, but they are expected not to and to be intelligent enough to comply with norms. To our knowledge frameworks for modeling abstract representations are limited to rule-based methods (norms, organizations and protocols) such as OperA[2].

On another extreme, system designers completely hand off direct control on agents. In that case, agents do not have any behavioral restrictions by design. Desirable collective outcomes are expected to be reached by the action of benevolent (namely, being cooperative [9]) agents. In general, these agents are provided with important reasoning capabilities (e.g. explicit representation of the environment, capable of automatically proposing coordination solutions).

Relation with Our Work. There are two relations between these methods and ours. The first relation concerns the integration of some of these methods in our framework of cultures. Indeed, our cultures are composed of two parts: values and practices. The former is explored by this paper but the latter is encompassed by that previous work. Practices correspond to standardized restrictions on behaviors. Practices conceptually directly encompass (abstracted or not) norms and protocols. By extension, practices can contain any other approaches which aim at restraining concrete agent behaviors.

The second relation consists in integrating values within this framework of methods. Values belong to the set of abstract methods. Thus, values should be used in complex in dynamic environments. They aim at promoting flexible and robust collaboration but are not the best tool for achieving efficiency.

3.2 Preferences

Preferences [7] are used for ordering a set of objects $\mathbf{\Omega}$. Formally, preferences are a transitive binary relation $\succ$ over $\mathbf{\Omega}$. For instance, "I prefer apple over oranges" can be represented by "apple$\succ$orange". In our setting, $\mathbf{\Omega}$ is the set of expected outcomes that can result from a decisions made by agents.

Representing preferences can be a difficult task when $\mathbf{\Omega}$ is large, because many objects have to be compared with one another. In spite of the cost of time, humans have difficulties to express object-to-object comparisons and prefer instead to use more generic statements (e.g. I prefer blue over red). In order to efficiently designing preference functions in using human-like descriptions, former research [7,25] proposes some core principles and solutions.

Efficient representations generally rely on the existence of criteria[3] which evaluate objects (e.g. cost in time, money or human lives). If criteria are independ, each independent criteria can be internally ordered (e.g. saving 200€ from the flames is preferred than saving 100 € from the flames), allowing to use the *Ceteris Paribus* ordering method (e.g. any outcome which saves 200€ from the flames is better than any other outcome saving 100 € from the flames with all other criteria being equal). Furthermore, ordering outcomes can also be achieved in integrating evaluations of criteria within the order (e.g. saving lives is more important than anything else, saving one hour of activity for an agent is as important as saving the equivalent of 100€ from the fire). These methods for efficiently representing preferences are particularly interesting in our setting, since they offer a more efficient and justifiable representation of desirable outcomes.

Preferences and Self-oriented Reasoning. Preferences can drive self-oriented agents for making complex decisions. Indeed, preferences can provide principle for supporting agent decisions. These principles are particularly relevant for resolving dilemmas (e.g. I have to visit A and B. Shall I start with A or B?). In particular, using criteria for designing preferences can help justifying their decisions made using values (e.g. human victims are more likely to be found in the housing location A there than in the warehouse B which contains many precious items. Since saving human lives is preferred on saving wealth, I go first in A). [25] propose a framework for supporting decisions using user values. [17] propose a framework for integrating preferences for making decisions in GOAL.

Preferences are inadequate for making very concrete decisions. Indeed, preferences require to *estimate the outcome* of decisions. When evaluating very concrete decisions (e.g. starting to walk using the right or left foot), preferences either have to integrate overly concrete aspects (e.g. I prefer starting walking using the right foot) or to include complex predictions about preference outcomes for these decisions (e.g. starting with the left foot is quicker in order to go to A but may induce extra costs), even if they may be adequate for punctual decisions. In the first case, the complexity of the preference function explodes with environment complexity. In the second case, this is the complexity of the estimation function which explodes.

Instead, preferences are appropriate for making more abstract decisions. These decisions which tend to be less frequent, lowering the number of outcomes to be estimated and the number of preferences this function should encompass. In addition, abstract decisions can be more easily connected with abstract preferences, which allow to keep the preference function simple.

Preferences are to be distinguished with goals. Goals correspond to concrete situations which can be achieved. They are either achieved, failed or ongoing. Preferences, instead, correspond to preferable situations. Preference cannot be "achieved" or "satisfied" but they permanently pursued and optimized against (unlike maintenance goals which are either maintained or failed). The same applies with norms which are either violated or not.

[3] Also known as perspectives.

Making preferences public help agents to create expectations about each other and to use preferences of other agents for adequately interacting with them. As an illustration of the difficulties triggered by hiding preferences, [4] propose a negotiation framework for agreeing on individually preferred outcomes without revealing private preferences. The effort which is deployed in this article in order to efficiently reach an agreement in spite of benevolent agents show the inherent difficulties for working together with private preferences. This inherent difficulty incited us to assume that in our framework preferences are visible by other agents.

Preferences and Collective Reasoning Preferences can be used for making decisions which takes into consideration preferences of other agents. This topic is particularly investigated by Game Theory [5]. Game Theoretical representations model for each agent in a group the desirability of collective outcomes for each possible collective actions of that group. These outcomes are ordered in terms of preferences for each agent. In this setting, agents are assumed to be rational, i.e. they should select an action which allow them to get the best outcome assuming that other agents are also rational. To that extent, Game Theoretical frameworks allow to make decisions while taking other agents into consideration. Nevertheless, Game Theory suffer multiple restrictions in terms of modeling: outcomes, other agents and their preferences should be known. In addition, predicting possible outcomes is intractable in presence of many agents with multiple choices.

Preferences can be used for directly driving collective action. This framework corresponds to mechanism design or implementation theory [19]. These frameworks extensively rely on Game Theory frameworks and thus suffer similar limitations caused by complexity.

Relation with Our Work Our work is tightly connected to these former works in several points. Indeed, our model of values relies on those models of preferences. Furthermore, we also want to use values for driving decisions of individuals but also to create expectations about behaviors of others.

We also aim at contributing to that domain. We propose a method for achieving collaboration in complex environments. We somehow expanding the challenges tackled by mechanism design but for complex problems.

4 Integrating Values in Decisions

This section presents how to integrate values within agent decisions. As a first step, we present as an inspiration how values influence humans decisions. Then, we use this inspiration for proposing a model and a possible implementation of the integration of values within agent decision processes. Next, present some advantages sharing values when designing agents: creating expectations about other agents and better interacting with humans. Finally, we conclude this section by relating the decision process with our running example.

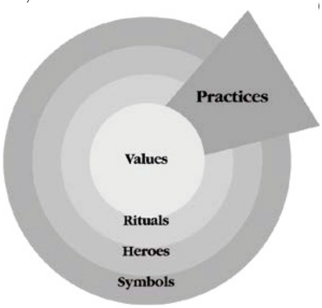

Fig. 1. Onion diagram, abstractly modeling the contents of cultures, from [18]

4.1 Inspiration

Before going into details of modeling values, let us first investigate more in details what human values are about. Values belong to cultures. Cultures are broadly investigated in social sciences [16,18]. These studies acknowledge that cultures are particularly fuzzy and difficult to grasp (unlike emotions which are relatively easier to pinpoint for instance). Nevertheless, these studies propose some simplified models in order to see the main influences of cultures which can be easily used as an inspiration for designing MAS.

These studies model cultures as collectively shared values (representing what individuals consider important, such as being normal or being rational) and practices (e.g. greeting by bowing or shaking hands), as illustrated in Figure 1. Practices are more visible, standardized, easy to change and situation dependent. Values are more internalized, implicit and all encompassing.

In this article, we discard the creation of models for driving practices. Indeed, these models have been extensively studied by the MAS community through norms, protocols, partly organizations and so on. Thus we leave interested readers consulting the rich and available literature about that topic.

Values are the most subtle part of cultures. Let us consider how they impact on decisions in order to model them. Cultural studies state that decisions made by individuals are influenced by human nature and cultures and personality. Human nature can be considered as individual rationality (e.g. selecting options leading to goal achievement). Personality corresponds to an individual variance for considering problems. This aspect does not seem relevant with regard to our goal for supporting collaboration, so it is left out in this article. Cultures are values and practices. As a rule of the thumb, when making a decision, agents consider how to achieve their goals (driven by human nature and possible external constraints such as protocols) while being conform with their practices (e.g. not violating cultural norms). If there are still multiple options available, values influence which option to select. This rule is of course not an absolute truth (e.g. values sometimes drive people to go against their self-interest or against practices, a model of such decisions is proposed in [11]), but it covers the standard decision process while remaining simple enough to model.

The term "values" can introduce some confusion and requires to be further introduced. A value is an abstract and broad perspective for considering a situation (e.g. achievement, self-direction, more examples are given in Section 6). The value*s* of an individual are informally the set of values which are given some importance by an individual. Nevertheless, values of an individual are not binary (e.g. either caring about achievement or discarding it) but relative with each other (e.g. giving relatively more importance to achievement than to self-direction). In the following value*s* are referred to as "value systems".

With that new information in mind, value systems seems to match well with preferences. [17,25] propose to use preferences as soft constraints, which allow to decide when multiple rationally and norm-compliant choices are available. Nevertheless, value systems are more specific than any preference function. Indeed, value systems should order abstract and driving values.

4.2 Integrating Value Systems in Agent Decision Processes

In order to make decisions which are streamlined with their value systems, agents require two capabilities: they have to be capable of *estimating the outcomes* of their decisions and order these outcomes using their *preference function*. These two capabilities are combined in order to make the decision which achieve the preferred estimated outcome according to agent's value system.

Estimating Outcomes Decision outcomes model the estimated effects of making a decision (e.g. performing an action) in a given situation. We talk about estimated effects, because they cannot be predicted for sure (e.g. actions might fail, the environment or other agents can interfere). Several solutions exist for representing and estimating about the consequences of decisions.

[22] uses a planning approach, assuming a bounded search span. The outcome of action corresponds to the expected satisfaction of the final situations which would be reached assuming that the agent selects the most satisfactory actions in the future. This representation implies some assumptions for the model. First, this model discards the presence of other agents. Second, expected outcomes are limited to single-dimensional real numbers in order to limit complexity. Third, a complete environment model is required.

[25, p. 109-136] uses a more logic-based approach, which consists in estimating "by hand" the consequence of a decision using qualitative tags. For instance, the expected effect of performing the "rescue" action is that the victim will be rescued, the agent will probably be late and the agent will be near the victim. Such representation should also describe longer term or uncertain consequences of decisions. For instance, hiding in the fire truck is immediately "safe" but "unsafe" situation if the fire is expected to spread.

Estimating long-term consequences of decisions can be difficult if these decisions are too concrete. This difficulty is caused by possible incompleteness of the environmental model, partial information and collective action. If all these elements would be modeled through some uncertainty factors, one would soon reach a point where the effects of actions are completely unknown.

The representation of [25] allows to cut short the search for all possible effects (which is intractable) by heuristically estimating consequences of decisions.

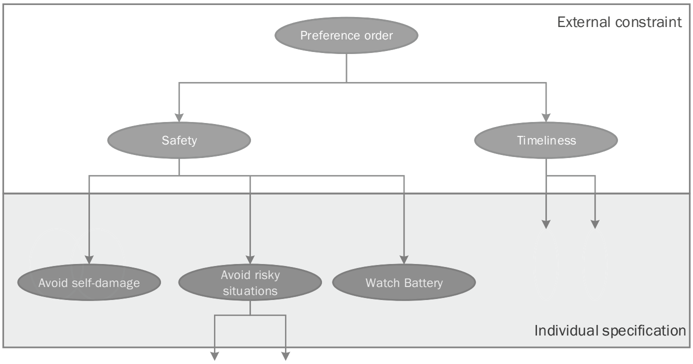

Fig. 2. Preference decomposition of firefighter agents. Each node of the thee is further specified by its children. Leafs are to be directly connected with evaluation outcomes.

However, this approach is more suitable for making strategic actions (e.g. whether to collaborate with a specific agent) rather than low-level decisions (e.g. whether to send a specific message or another the other agent). The reason being that the range of outcomes and of possible situations explodes when being concrete, thus making intractable the design of heuristics.

Ordering Preferences. Preferences are used for modeling the value system of agents. Technically, preferences are used for ordering the values of agents.

Ordering outcomes require to connect the value system to outcomes. Nevertheless but value systems are too abstract to be easily connected with concrete decisions (e.g. evaluating the safeness of the route to take). In order to better evaluate the desirability of an outcome for a given value, we propose to use more concrete criteria for supporting this evaluation. For instance, the evaluation of the safeness of a route can be supported by the three criteria: “avoiding risky situation”, “avoiding self-damage” and “watching battery”. Of course, these criteria are just a guidance and the decision tree may not be complete. Decisions which badly fulfills these more specific criteria can still be preferred (e.g. going through the fire without recharging is bad for all three options but still better than getting around the fire which would certainly lead to burning the agent).

Similarly, these criteria can be further related with more concrete criteria. Indeed, these criteria can still be abstract and evaluating them can rely on other and more concrete criteria (e.g. relating “avoid risky situations” with “avoid fire” and “avoid collapse risks”). From a more global perspective, value systems can be represented as a generalization tree[4], in which children of a criteria are specific criteria which can be used as a support for evaluating its parent. An illustration of that tree is proposed in Figure 2.

From a collective perspective, this generalization tree is partly shared by agents. More precisely, the highest part of the tree is shared by all agents. Then, this tree can be refined locally by specifying criteria.

---

[4] Or directed acyclic graph, since multiple criteria can rely on similar more concrete criteria.

From a design perspective, this representation combines a top-down and a bottom-up approach. Indeed, abstract value systems can be connected to more concrete decisions in further specifying them. The other way around, designers can aim at making abstract decisions which outcomes can be connected to more abstract criteria.

Using such a generalization tree has multiple advantages. First, designers of preference functions can connect their abstract decisions in a step-wise way, by adding criteria in an increasing order of concreteness. This decomposition has the advantage of avoiding to connect a value to far too many criteria for making principled decisions. Furthermore, each criteria can be related to sub-criteria which are not too far enough in terms of abstraction, helping to better justify decisions. Third, each criteria can be reused used as a sub-criteria multiple times, lowering the cost of designing criteria (e.g. "avoiding self-damage" can be used as a sub-criterion for both "safety" and "limiting costs" criteria). Last but not least, this representation leaves a lot of freedom about using criteria when designing preferences. For instance, just by changing how a criteria relies on its sub-criteria, agents can be driven to extreme safety ; extreme punctuality ; or a balanced combination of the two.

4.3 Towards Hybrid Agents Using Values

Value systems are adequate for supporting decisions with abstract reasoning, but they are impractical for handling concrete behavior. Concrete behavior is better handled by traditional solutions for designing agents or coordinating them (like plain code, BDI, planning, protocols), but those solutions are less adapted for integrating abstract drives. These two solutions are complementary and there is a clear gain in connecting both together. This section proposes some principles for an implementation model of the influence of value systems on decisions.

A first solution consists in implicitly integrating value systems within decisions, i.e. without explicitly relating a model preferences with decisions. This approach has the advantage of shortcutting the design of a complete value function and avoiding to evaluate outcomes of actions. But, this approach has multiple limitations. First, value systems cannot be changed without having to directly change agent decision process. Second, value-related decisions are less justifiable, introducing subjectivity. This subjectivity can be difficult to handle if preferences are related to multiple criteria and each criteria can encompass a wide range of evaluations (e.g. if timeliness and safety are both relatively important and enter into consideration for a decision, hard to compare each solution). In such a case, it may become difficult to determine which decision to select in a principled way.

Most of these issues can be solved by explicitly integrating value systems within agent decision processes. Multiple solutions can be investigated for achieving that. For instance, [17] proposes to integrate preferences for determining which plan to select when multiple plans can be fired in a given situation.

As an extension, we suggest to use a hybrid model. This model would be composed of three layers. A *tactical* layer would cope with concrete decisions. A *strategical* layer using abstract value systems would manage abstract longer-term decision, as suggested by [25]. These two layers would be connected through an interaction layer. In this layer, strategical decisions can influence tactical decisions, for instance by changing goals, activating a module[8] or executing a protocol. In return, strategical decisions are influenced by tactical outcomes, for instance through belief updates, goal fulfillment, or a specific procedure state is reached. Hybrid agent architectures are not new. The more adequate one for that purpose that we found in the literature is inteRRaP [21]. inteRRaP proposes different reasoning layers with different internal logics (reacting to the environment, planning from a single agent point of view, planning from a group point of view). Nevertheless, such architectures tend to focus on different perspective than ours (social versus individual or system reactivity). They do not seem immediately applicable in our context but propose an interesting inspiration.

As a closing word, the aim of this subsection consists in showing the type of decisions which are adequately connected with value systems. In particular, value systems appear particularly useful for supporting and influencing abstract strategical decisions. This type of decisions show the ease to connect value systems with hybrid models. Nevertheless, the global aim of this article is not to provide a very concrete implementation of one solution for using value systems, even if such an implementation is an immediate follow-up. Instead, we want to show how value systems can be beneficial for designing MAS and this subsection highlights how easily they can be implemented.

4.4 Benefits of Value-Based Agents

Creating Expectations about Other Agents. Since value systems are expected to be shared within the agent community, agents can create expectations about each other drives and resulting behaviors. Furthermore, value systems can be used for creating expectations about the environment and the society. For instance, if timeliness is important, then other agents are assumed to be on time. As a result, agents can create the social assumption that schedules are reliable. Thus agents can plan while tightly optimizing their schedules, leading to overall higher performance. The same can be applied for environmental assumptions (e.g. if timeliness is important, resources are assumed to be available on time).

Expectations can be integrated in several ways. A first solution consists in integrating them by design. Designers can integrate them knowing what is collectively considered as important (e.g. if timeliness is important, then other agents prefer not to accept too many tasks at once). This solution requires human intervention, but it allows to shortcut the need for creating these expectations on the fly. Furthermore, designers can integrate their insights, providing a richer variety of expectations within agents.

A second solution consists in automatically creating expectations about behaviors of other agents. This solution can be achieved in relying on the assumption that value systems are shared. Agents can use these systems for making Game-Theoretical expectations about behaviors of other agents in estimating their preferences, if the setting is simple enough. Agents can also be used for evaluating the responses of other agents. Such an expectation is particularly useful for achieving cooperation. Indeed, preferences can be used for driving the generation and the selection collective plans as well as justifying them to other agents, which is a basic task of cooperation. Nevertheless, agents only share the most abstract part of their value systems and only partly their beliefs. Thus, estimating other's preferred outcomes may be imperfect but should still be relatively close to reality.

Making the comparison with other approaches, expectations created by value systems are more abstract and less certain than those resulting from concrete approaches for collaboration (e.g. norms). Indeed, expectations for value systems are not adequate for sharply predicting concrete behaviors. For instance, "saving humans" does not mean than agents will rescue victims as soon as possible. Instead, agents may believe that more lives can be saved by extinguishing the fire (e.g. as in the running example in a skyscraper). Nevertheless, if some option is clearly better than other, agents are very likely to create adequate expectations about other agents. Furthermore, the more the value tree is shared, the more agents are more likely to create adequate expectations. At the opposite, expectations provided by other methods for achieving collaboration (e.g. norms) are more concrete and specific to a situation or interaction (e.g. protocols). Thus, these expectations tend to be relatively precise and accurate. As a concluding word, both expectations are complementary. Combining them offers useful perspectives.

Value Systems: Towards Human-Agent Interaction? Value systems are particularly useful for supporting interactions between human and agents. Using value systems for linking human and agents is relatively little cumbersome for humans. Indeed, value systems are relatively intuitive to grasp (e.g. efficiency, comfort, timeliness). They avoid the learning curve induced by understanding the possibilities of more concrete approaches which restrict behaviors of agents. Furthermore, humans can intuitively adapt value systems of agents to their needs, without requiring specific programming capabilities. From the agent side, given an allocated value system, agents can determine which expectations human users make about them. They can also use value systems for creating more concrete rules for interacting with humans.

Value-based systems are also appropriate for avoiding systems to be rejected due to their complexity. Indeed, decisions are supported by value systems. Thus, decisions can be explained in terms that humans can understand (e.g. "I extinguished the fire because I think it will save more lives"). Consequently, agent decisions appear less arbiraty to humans, which can then better support collaborative interactions with humans, in human-agent societies.

Finally, value systems provide a simple model of human psychology. Consequently, agents can use value systems of humans in order to create expectations about them, as they would with other agents with value systems. Value systems are abstract enough to create some abstract expectations about human behavior, while not creating expectations about concrete human behavior. These latter expectations are likely to mismatch, unless strong restrictions are imposed to humans behaviors.

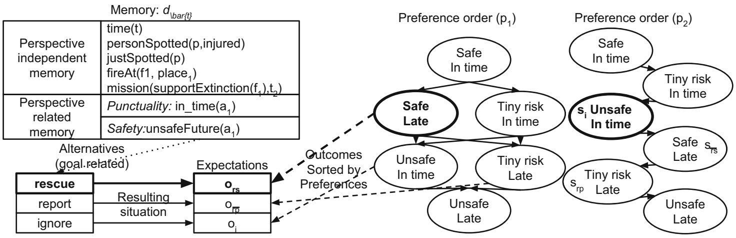

Fig. 3. Decision process for an agent with preference function p_1 in situation $d_{\bar{t}}$ Bold lines and text highlight choices made by the agent. If a_1 uses preference p_2, then the selected action is "ignore".

4.5 Running Example

Perspectives In this example, four preferences are considered: safety p_1, punctuality p_2 and combinations of these two. The whole decision process is illustrated in Figure 3.

Estimating Outcomes. If "rescue" tactical action is performed, a_1 expects o_{rs}: $late(a_1), at(injured\_person)$. If "report" is performed in d_t, a_1 expects o_{rp}: $in\_time(a_1), reported(a_1, person), at(unknown\_position)$. If "report" is performed in $d_{\bar{t}}$, a_1 expects $o_{\bar{rp}}$: $late(a_1), at(unknown\_position)$. If "ignore" is performed, a_1 expects o_i: $in\_time(a_1), unreported(person), at(fire)$.

Preference Functions. Safety is represented by the following order: situations with the property *safe* are better than those with the property *tiny_risk* which are better than those with the property *unsafe*. *safe* is true if the agent is far from fire (e.g. rescuing the injured person, thus $at(injured\_person)$ is true), *tiny_risk* is true the agent may be have to move to the fire (e.g. when $at(unknown\_position)$ is true, for instance when the agent waits for leader instructions) and *unsafe* if the agent is near a fire (thus $at(fire)$ is true). Punctuality is represented by the following order: situations with $in\_time(a_1)$ are better than those with $late(a_1)$.

p_1 and p_2 are combinations of safety and punctuality. p_1 compromises punctuality and safety. p_2 drastically favors punctuality over safety: for two situations s_1 and s_2; s_1 is better than s_2 if, for timeliness s_1 is better than s_2 or they are incomparable with regard to timeliness and for safety s_1 is better than s_2.

5 Integrating Value Systems in Collaboration

This section investigates the benefits of integrating value systems for improving collaboration from a collective perspective. As a first observation, value systems alone insufficient for achieving collaboration alone. First, because values support drives which are not directly related to system goals. Second, because value systems are too abstract for constraining concrete agent behaviors, which are particularly important for conducting concrete interactions with other agents (e.g. value systems are not adequate for determining whether to drive on the left or on the right). These more concrete interactions are better supported by more traditional methods (e.g. norms, organizations, cooperation). In the following, we aim at combining value systems with such methods.

This section investigates first the influence of value systems over collaboration in human societies using concrete methods for supporting collaboration. Then, we use this link as an inspiration for improving collaboration in artificial societies using value systems.

5.1 Inspiration

The relationship between cultures and collective performance has been broadly studied for human societies. These studies focus on generalist influences of cultures on collective behaviors but some studies focus more specifically of this influence in the context of organizations or corporations. Figure 4 illustrates such an observed correlations between some cultural features (power distance and uncertainty avoidance) and preferred organizational patterns. These patterns drive in turn the type of collective performance profiles which can be achieved (e.g. bureaucracies are fitter for simple and static environments while adhocracies are fitter for more complex and dynamic environments [20]). This section aims at replicating such a property, in investigating how value systems can improve collaboration in a MAS.

These studies give high importance to the influence of culture, for the main reason that they drive individuals towards common individual and collective outcomes. This drive helps individuals to understand and create expectations about each other. Furthermore, value systems also highlight what is important for individuals and drive them towards similar outcomes. Thus, individuals decisions tend to be streamlined when having multiple possible choices (e.g. 5 minutes late for a meeting versus gaining 10 minutes for oneself), allowing to support abstract collective properties.

5.2 Combining Values and Former Approaches for Collaboration

Value systems and concrete approaches tackle very different problems. This section aims at considering these differences and how they can be combined with each other.

Abstract Values and Concrete Collaboration The higher is a criteria in the value tree, the more this criteria is abstract and thus the least environment-dependent it is. Value systems are relatively reusable, even if they may require to be locally adapted to specific environments. Nevertheless, they are not perfectly adequate for making very concrete decisions.

Value systems enforce properties which are abstract and relatively independent from a given system (e.g. being in time, not causing others do be delayed). Conversely, traditional methods for achieving collaboration tend to be more related to environmental or interaction properties (e.g. concrete norms, protocols). Thus, these approaches tend to better enforce concrete decisions.

Value systems appear to combine well with other solutions for achieving collaboration. They offer generic and abstract drives which lead to useful system properties (e.g. timeliness, degree of independence of agents) without strictly forbidding any concrete behaviors. In addition, value systems help agents to make abstract actions which influence more concrete decisions (e.g. which goal to select, which abstract method to use in order to tackle a given problem). Nevertheless, collaboration solutions are still crucial for enforcing concrete environmental and interaction properties, which are required for tightly connecting agent actions.

Drives versus Constraints. Value systems determine what agents should consider as important, given some abstract and generic perspectives. Value systems can be seen as drives. In particular, value systems are shared within the community. Thus, agents are all driven towards similar outcomes and all consider the same things as important. This shared drives is crucial for avoiding individuals go against expectations of others. Furthermore, this shared drive helps behaviors to conflict with each other, given value-driven properties. For instance, with timeliness all agents know the importance of deadlines, which drive agents and are used for creating assumptions. Thus, no agent jeopardizes these deadlines without a good reason.

Value systems systems and other approaches complement well with regard to this aspect. Indeed, value systems can be used for determining *what* is important for agents in general. Conversely, other approaches determine *how* to behave in given expectable situations. Value systems help to abstractly and collectively drive agents towards desirable collaborative behavior. Value systems systems provide to agents some high-level guidelines about what agents should pursue when using the more concrete tools for collaborating (e.g. do not circumvent a safety rule by doing something dangerous). This link is particularly relevant for achieving cooperation. Indeed value systems can be used for determining collective goals and plans and for creating expectations about behavior of individuals with regard to that plan (e.g. with timeliness, sub-plans are likely to be assigned deadlines, which are highly used for optimizing global plans).

Furthermore, value systems guide agents in situations where agents are given some freedom. Such a situation can be desired by system designers. But this situation can happen if the agent has to perform an unexpected decision, in which rules do not apply. In that case, supporting agent decision is crucial in order to prevent the agent to go against the system due to a lack of guidance.

Performance. The influence of value systems on collaboration is relatively independent from the environment. To that extent, collaboration promoted by value systems tend to be relatively less sensitive to failures or to unexpected events, supporting robust and flexible collaboration. Nevertheless, this relative independence with the environment makes difficult to tightly optimize interactions of agents, making more difficult to pursue high efficiency.

This influence is relative to other approaches and provided in a general context. Indeed, value systems support a wide variety of behaviors with different performance profiles (e.g. timeliness tend to improve time to completion at the expense of robustness) and other approaches can be particularly robust or flexible.

Value systems are adequate for handling complex and dynamic environments. Indeed, value systems do not restrain agent behavior for resolving complex problems. To that extent agents are not restricted in the way they resolve problems and can thus develop more freely adaptive solutions. Such possibility is crucial for complex and dynamic environments for which restricting concrete behaviors may be hard to determine or can become inadequate.

Designing Collaboration with Value Systems. Integrating value systems raises now perspectives for designing collaboration.

Top Down vs. Bottom Up Concrete approaches restrain local behaviors and generally aim at achieving more abstract properties, using a bottom-up approach. Abstract approaches provide abstract constraints and are made more concrete, in a top-down approach. Value systems expand the latter category, which is at the moment composed of abstract normative and organizational systems (e.g. OperA [2]).

These two approaches can be combined together in order to leverage design costs. Indeed, reaching concreteness with abstract methods and abstraction with concrete methods is particularly expensive. Combining the two allows to use each method for the problems they are is adequate for.

Design Guidelines for Agents Value systems determine what agents consider as important. To that extent, they provide clear guidelines about what agents should focus on and thus the type of mechanisms they should encompass (e.g. time management for timeliness). From a collective perspective, value systems help to determine the set of concepts that agents can use for interacting with each other (e.g. deadlines for timeliness).

Designing Drives Designing value systems differ from former approaches because system designers have to drive agents instead of restricting their behavior. With value systems, system designers can only determine agents preferences and connect these preferences with the environment and decisions to be made. This form of design can appear difficult on first glance because designers cannot directly affect behaviors but instead about drives which would lead to desired behaviors. Idem, when debugging, system designers have to investigate whether agents making undesired decisions wrongly estimated outcomes or whether agents are not inclined enough towards a more adequate value.

Designing the Whole System As a recommendation, value systems and other methods for collaborating should be streamlined by design. For instance, if timeliness important, then time should play a crucial role in norms and organizations. Norms should determine time limitations, such that agents which rely on these norms can integrate them in their schedule in order to be sure they do not miss deadlines. If that is not the case, value systems may conflict with other methods of collaboration, leading to counter-productive behaviors.

As a future work, we consider the integration of value systems with systems including dynamic constraints for supporting collaboration (e.g. dynamic environments and norm base). In such scenario, we may expect agent to adapt the constraints base to the environment but also to value systems (e.g. removing safety rules leading to further danger). If value systems are also assumed to change, we suggest to make the value-base less dynamic than more concrete aspects, since value systems should be less sensitive to the environment.

6 Example of Useful Values for Improving Collaboration

In previous sections, we presented how value systems can be integrated in the design of MASs. This section proposes more concrete examples of what kind of objects can be integrated as a value. In particular, these examples of value systems are inspired by in human cultures. The core aim of showing these examples, in addition to provide immediate solutions, is to give an idea of the level of abstraction that we have in mind when we discuss about value systems. In addition, we want to show that each value has an important impact potential on plenty of individual and collective decisions. This impact has in turn plenty of consequences on design perspective (e.g. which norm to combine with a given preference).

As a disclaimer, these examples should serve as an inspiration of solutions which can be used for improving collaboration. Nevertheless, the model presented in this article does not aim at faithfully replicating human behaviors. Consequently, we kept only examples of cultural influences which are relevant for improving collaboration in MAS. Aspects which are too human oriented (e.g. desire from hedonism, ways to express emotions) are left out in this article.

Value systems are not a new topic and former researchers have empirically established them for human societies. The most used model is the Schwartz value model [24]. Schwartz empirically recognizes 10 value systems: stimulation, self-direction, universalism, benevolence, conformity, tradition, security, power, achievement and hedonism. These value systems can directly be used as criteria for modeling artificial value systems, as proposed in this article.

Cultural dimensions provide another source of inspiration for designing value systems. Each dimension evaluates cultural responses to dilemmas, like "What is more important, rules or relationships?" [16]. These dilemmas highlight some abstract choice which impact plenty of decisions. These choices indicates the influence of underlying value systems which can be integrated in our value-model (even if cultural dimensions are not value systems). On the track of linking culture and collective behavior in MASs, [26] conceptualize links between cultural dimensions, individual behavior, emerging collective behavior and performance.

In the rest of this section, we briefly introduces cultural dimensions from [16,18] which highlight crucial dilemma can be considered when integrating value systems in collaborative MASs.

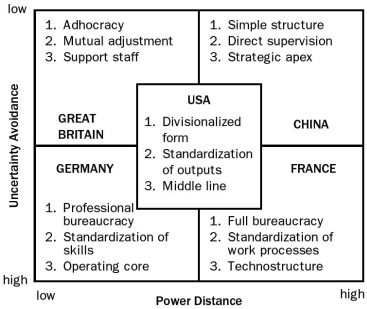

Fig. 4. Culture and preferred organizational pattern, from [18]

6.1 Power Distance (PDI)

[18] defines Power Distance as the cultural relative importance given to formal and informal statuses.

In high PDI, subordinates prefer to give information and decision power to leaders. Leaders are expected to decide and assign clear orders for subordinates. As a result, in such a culture, leaders tend to have the more information and can thus make well-informed decisions. In addition, leaders can further optimize subordinate schedules because subordinates tend to be expected to wait for and comply with instructions, allowing to increase collective efficiency. From the perspective of system performance, high PDI tends to lower system robustness: leaders are bottlenecks (in particular in information-rich or complex environments). Thus, failing or missing leaders leads to a collapse of the communication and decision structures. In the running example, agents with high PDI value systems are likely to perform the "report" action. Indeed, this action makes sure that leaders possess the adequate information without going against orders given by agents. If agents run out of time or the leader is assumed to already have the information, they can also "ignore" the victim. "Rescue" is unlikely triggered because they are not supported to take that initiative without receiving the authorization.

In low PDI, individuals give less power to statuses. They consider themselves as independent and of equal value with regard to information and decisions. They are likely to take more initiatives and carry their own tasks. As a result, individuals have locally more information but leaders are less informed to technical details, providing with higher-level feedback. Such a culture is likely to increase system robustness, since no agent is critical to the system. Nevertheless, the lack of centralization of information and decisions tends to lower efficiency. In the running example, the lack of strong leadership is likely to let a_1 determine which action to pick, maximizing utility from an individual perspective. To that extent, "rescue" the victim is the most likely option, unless the task to be achieved by the a_1 particularly important. In the latter case, the agent may "report" if given enough time or "ignore" if lacking time.

6.2 Uncertainty Avoidance (UAI)

[18] depicts uncertainty avoidance the cultural sensitivity of individuals towards the certainty of their situations and their decisions.

In high UAI, individuals prefer to can create strong assumptions about their beliefs. To this extent, they either try to lower this uncertainty either by getting more information or by making assumptions about it (e.g. someone will support me when I will enter the burning house). As a result, individuals prefer to behave according to standards, further reducing uncertainties for itself as well as for others. Thus, as a an emerging property, individuals can expect less variability from actions of other agents or environmental states, further enforcing the benefits of making assumptions. High UAI is very efficient for static environment because a lot of assumptions can be made about the environment allowing to optimize collective action. Nonetheless, this preference is not flexible: if the environment is dynamic, either agents constantly update their procedures or they may try to apply mis-adapted procedures leading to failures. In the running example, an agent with high UAI is likely to pick a solution which minimizes generated uncertainty. "Rescue" is the most unlikely option, because it may prevent the agent to be at the fireplace while it should to be there, creating uncertainty for others which is particularly undesirable. "Ignore" may lead to the casualty of the victim which is mixed feelings. "Report" seems the best option, since it would lower uncertainties of the fire commander without creating so many uncertainties for the firefigter.

In low UAI, individuals are less sensitive to uncertainty. Their behavior is more directed by goals than by procedures. To that extent, behaviors are likely to be more adaptive, leading to more variability in environmental situation. In that case, this variability is not problematic because other agents expect the environment to be uncertain. They do not make inappropriate assumptions. This adaptability tends to raise collective flexibility but lowers efficiency due to difficulty for standardizing. In the running example, an agent with a low UAI value system has little incentive for following standards. Such an agent is likely to make similar decisions as a low PDI agent, thus maximizing local utility.

6.3 Sequential versus Synchronous Perception of Time

[16] describes two paradigms to consider time management: sequential and synchronous.

In sequential time, individuals consider time as a sequence of events. Respecting deadlines is very important to not delay this time-line. As consequence, timeliness is expected from other individuals. From a collective perspective deadlines and schedules are expected to be more reliable. Thus, such an APF is likely to improve efficiency and lower time to completion in allowing accurate planning of tight schedules. But, this approach fails when time considerations cannot be estimated accurately (lower flexibility) and is sensitive to failures, missing agents and congestion (lower robustness). In the example agents with sequential value-systems will do their best in order to be on time. In any situation they are likely to “ignore” the victim. Nevertheless, if timeliness is not set to an extreme importance, they are likely to “report” in d_t.

In synchronous time, time is considered as a resource to be planned against. To that extent, individuals prefer to locally maximize their efficiency, for instance by taking opportunities. With this consideration of time, timeliness is less important than lowering efficiency, so individuals tend to be late. Other individuals can expect delays and thus can, for instance, prepare activities for filling waiting time. This form of time management can also lead to high efficiency, if the environment is suitable for “filling in” waiting time. A negative point concerns the unpredictability of time to completion: an agent can continuously delay a task because of getting opportunities to perform other tasks more efficiently. In the example, synchronous agents select which action to perform in comparing the time cost incurred by selecting one of the other option (time for extinguishing a wider fire if “help” and time for getting back and rescuing for “ignore”, estimated cost for sending someone else rescuing for “report”).

7 Conclusion

With a similar idea than BDI, norms, organizations and protocols, this article proposes to use aspects of human societies, namely cultures, as an inspiration for improving collaboration in problem-solving MAS. Cultures provide a rich inspiration for MAS, by distinguishing two levels of influences: abstract values and concrete practices. The latter being already well studied (e.g. norms, protocols), we focus on the former. We propose solutions for expanding agent design in order to incorporate value systems and we investigate the benefits of integrating value systems on top of practices for driving agent societies.

Value systems are abstract drives shared by agents. Value systems specify uniformly to all agents some abstract aspects that they should consider as important when making decisions. By sharing a similar emphasis on what is important, agents can more easily determine which decisions are streamlined with collaboration. From a collective perspective, this drive allows the emergence of abstract desirable properties (e.g. deadlines tend to be reliable). Value systems can also be used by agents for creating some weak expectations about drives of other agents, their behaviors, the environment and the society (e.g. an agent in a culture promoting safety can expect support from the others).

Value systems offer a complementary perspective to existing approaches for supporting collaboration. Existing methods for supporting collaboration would gain in also integrating value systems. From an individual perspective, value systems are abstract enough for driving decisions in possibly any situation, particularly in unexpected scenarii, which may be above the limits of more concrete methods. From a collective perspective, value systems provide abstract directions about *what* agents shall pursue while practices describe more concretely *how* agents should behave. This complementarity has numerous benefits. The main one being that they provide cross views for tackling problems. This crossed view allows to avoid inherent explosion in terms of design complexity which happens for solving problems with an inappropriate approach. For instance, individual sharp and well expected behavior is relatively captured by norms while abstract collective patterns are more easily driven by abstract value systems. From a performance perspective, concrete approaches are appropriate for achieving high efficiency by providing tight guidance in standard situations. Value systems offer high flexibility and robustness in providing agents core principles for making principled decisions when rules fail to direct them. This decision support makes of value an adequate solution for coping with complex and dynamic environments.

Concerning possible applications of value systems, we identified three categories of applications: *Unknown, evolving environments* (e.g. exploration, building dynamic sensor networks). In such an environment, system designers cannot easily determine beforehand adequate patterns of collaboration. Agents should rather do it on the fly, depending on the situation. Value systems offer an adequate guidance for driving collaboration in these many possible situations.

Adversarial environments (e.g. military applications, game-oriented applications). In such an environment, concrete approaches are risky because they tend to force some behavioral patterns for achieving collaboration. These patterns put the system at risk of being exploited (e.g. trigger an emergency call in order to attract all the drones around, weakening the main entrance). Instead, value systems offer versatile and adaptive behavior which still aims at promoting collaboration.

Human-machine interactions (e.g. health-care robots, serious gaming). As further advocated in this article, value systems are numerous advantages for connecting humans with software agents. Value systems are intuitive, easy to adapt, provide a simple model of human drives and agents can use value systems for justifying their decisions.

For future work, we plan to integrate value systems within an agent. We plan on using the hybrid agent, which combines a value-driven strategical layer with a tactical layer influenced by traditional design tools. The layer for practices should encompass a high-level BDI representation such as $2APL$ or GOAL. Then, we plan to create a society of such hybrid agents and investigate their individual and collective behavior on a concrete problem. This implementation allows us to confront ourselves to technical issues raised by real problems and to investigate how far value systems can be used. From this confrontation, we expect to gain further knowledge about methodologies which are relevant for the designing value-based agents and about the connexion between value-systems with other methods for supporting collaboration.

References

1. Abdallah, H., Emara, H.M., Dorrah, H.T., Bahgat, A.: Using Ant Colony Optimization algorithm for solving project management problems. Expert Systems with Applications 36(6), 10004–10015 (2009)

2. Aldewereld, H., Dignum, V.: OperettA: Organization-Oriented Development Environment. In: Dastani, M., El Fallah Seghrouchni, A., Hübner, J., Leite, J. (eds.) LADS 2010. LNCS (LNAI), vol. 6822, pp. 1–18. Springer, Heidelberg (2011)
3. Amato, C.: Cooperative Decision Making. In: Kochenderfer, M.J. (ed.) Decision Making Under Uncertainty: Theory and Application, ch. 7, pp. 159–187. MIT Press (2014)
4. Baarslag, T., Fujita, K., Gerding, E.H., Hindriks, K., Ito, T., Jennings, N.R., Jonker, C., Kraus, S., Lin, R., Robu, V., Williams, C.R.: Evaluating Practical Negotiating Agents: Results and Analysis of the 2011 International Competition (2012)
5. Binmore, K.: Playing for Real: A Text on Game Theory. Oxford University Press (2008)
6. Boella, G., Torre, L., Verhagen, H.: Introduction to normative multiagent systems (2006)
7. Brafman, R.I., Domshlak, C.: Preference Handling An Introductory Tutorial. AI Magazine 30, 58–86 (2013)
8. Cap, M., Dastani, M., Harbers, M.: Belief/goal sharing modules for BDI languages. In: 2011 CSI International Symposium on Computer Science and Software Engineering (CSSE), pp. 87–94 (2011)
9. Castelfranchi, C.: Formalizing the Informal? In R. Demolombe and R. Hilpinen, editors. In: Proceedings of the 5th International Workshop on Deontic Logic in Computer Science (DEON 2000), Toulouse, France, January 20-22. Informal Proceedings, pp. 3–5. ONERA (2000)
10. Di Marzo Serugendo, G., Gleizes, M.-P., Karageorgos, A. (eds.): Self-organizing Software, from Natural to Artificial Adaptation. Springer (2011)
11. Dignum, F., Dignum, V.: Emergence and enforcement of social behavior. In: Anderssen, R.S., Braddock, R.D., Newham, L.T.H. (eds.) 18th World IMACS Congress and MODSIM 2009 International Congress on Modelling and Simulation, pp. 2942–2948 (2009)
12. Dignum, V., Dignum, F.: A Logic of Agent Organizations. Logic Journal of IGPL 20(1), 220–240 (2011)
13. Durfee, E.H.: Distributed Problem Solving and Planning. In: Luck, M., Mařík, V., Štěpánková, O., Trappl, R. (eds.) ACAI 2001. LNCS (LNAI), vol. 2086, pp. 118–149. Springer, Heidelberg (2001)
14. Durfee, E.H., Lesser, V.R.: Using Partial Global Plans to Coordinate Distributed Problem Solvers. In: Bond, A.H., Gasser, L. (eds.) Proceedings of the Tenth International Joint Conference on Artificial Intelligence IJCAI 1987, pp. 875–883. Morgan Kaufmann Publishers (1987)
15. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley (1999)
16. Hampden-Turner, A., Trompenaars, C.: The seven cultures of capitalism: Value systems for creating wealth in the United States, Japan, Germany, France, Britain, Sweden, and the Netherlands. Currency Doubleday (1993)
17. Hindriks, K.V., van Riemsdijk, M.B.: Using Temporal Logic to Integrate Goals and Qualitative Preferences into Agent Programming. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2008. LNCS (LNAI), vol. 5397, pp. 215–232. Springer, Heidelberg (2009)
18. Hofstede, G., Hofstede, G.J., Minkov, M.: Cultures and Organizations: Software of the Mind, 3rd edn. McGraw-Hill Professional (2010)
19. Maskin, E.: Implementation theory. In: Aumann, R., Hart, S. (eds.) Science, vol. 1, pp. 237–288. Elsevier (2002)

20. Mintzberg, H.: The structuring of organizations: A synthesis of the research. Prentice-Hall (1979)
21. Müller, J.P., Pischel, M.: The agent architecture inteRRaP: Concept and application, p. 99. Citeseer (1993)
22. Nguyen, T., Do, M., Gerevini, A., Serina, I., Srivastava, B., Kambhampati, S.: Planning with Partial Preference Models. CoRR, abs/1101.2 (2011)
23. Omicini, A., Ricci, A., Viroli, M., Castelfranchi, C., Tummolini, L.: Coordination artifacts: Environment-based coordination for intelligent agents. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004 (2004)
24. Schwartz, S.H.: A Theory of Cultural Value Orientations: Explication and Applications. Comparative Sociology 5(2), 137–182 (2006)
25. van der Weide, T.: Arguing to motivate decisions. PhD thesis, Utrecht Universiteit (2011)
26. Vanhée, L., Dignum, F., Ferber, J.: Towards Simulating the Impact of National Culture on Organizations. In: Alam, S.J., Van Dyke Parunak, H. (eds.) MABS 2014. LNCS (LNAI), vol. 8235, pp. 151–162. Springer, Heidelberg (2013)
27. Wooldridge, M.: Introduction to Multiagent Systems. John Wiley & Sons, Inc. (2002)