

Research Article

Value-Based Reasoning in Autonomous Agents

Tomasz Zurek^{1,*}, Michail Mokkas²

¹*Institute of Computer Science, Maria Curie-Skłodowska University, plac Marii Curie-Skłodowskiej 5, Lublin, 20-400, Poland*

²*Polish-Japanese Academy of Information Technology, Koszykowa 86, Warsaw, 02-008, Poland*

ARTICLE INFO

Article History

Received 27 Mar 2020

Accepted 11 Jan 2021

Keywords

Reasoning
Model
Goals
Values
Expert systems
Autonomous agents

ABSTRACT

The issue of decision-making of autonomous agents constitutes the current work topic for many researchers. In this paper we propose to extend the existing model of value-based teleological reasoning by a new, numerical manner of representation of the level of value promotion. The authors of the paper present and discuss proofs of compatibility of both previous and current models, a formal mechanism of conversion of the parameters of the autonomous device into the levels of promotion of values, the mechanism of integration with machine learning approaches, and a comprehensive argumentation-based reasoning mechanism allowing for making decisions.

© 2021 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

A distinctive feature of modern technical devices is their increasing self-sufficiency. One may presume that in the future the user will only formulate the most rudimentary rules of how a device works (or goals), whereas the everyday aspects of system operation will be regulated completely autonomously. The most advanced types of such devices are self-driving vehicles, where the user merely sets the journey's destination and the vehicle develops the itinerary on its own and makes hundreds of traffic-related decisions.

The issue of decision-making by autonomous agents has been a research topic of a number of researchers representing various fields of science [1]. Typically, it is perceived as the selection of the best possible action which should be performed. Classical decision theory, developed mostly by economists, relies on the concept of *utility*, where utility is a way of representing the agent's preferences. Utility theory can be combined with probability theory to build the principle of maximum expected utility (PMEU) which states that an agent is rational if and only if it chooses the action that yields the higher expected utility, averaged over all the possible outcomes of the action [2,3]. In early 2000s an alternative approach to classical decision theory emerged where the basis for modeling a decision-making situation is the logic-based representation of human reasoning and argumentation and the so-called practical reasoning, that is reasoning regarding the action which should be performed by the agent (discussed from a various points of view: [4–8], and

others, where, e.g. [9,10], present it in a general discussion of argumentation). In this work we intend to focus on the latter approach where decisions are made on the grounds of a model of human reasoning and argumentation.

In major decision-making models it is usually assumed that the purpose of system operation is to accomplish some state of affairs pre-declared by the user [11–13] (and many others). Many authors (including [14,15]) believe, however, that increasingly complex devices must be endowed with the ability to not only find the best possible way to attain the state of affairs set by the user, but also the ability to set their own goals themselves; the model proposed in this paper is based on this assumption as we believe that the increasing autonomy of devices calls for the realization of systems able to self-sufficiently set themselves a goal and take measures to accomplish it.

Zurek [14] presents a framework allowing for autonomous (based on values which should be promoted) goal-setting by a device. The model relies on the differentiation between several kinds of goals: abstract (i.e., minimal levels to which values should be promoted) and material (i.e., particular states of affairs which fulfil abstract goals).

The objective of this paper consists in proposing for the model from [14] a new method of the representation of the levels to which particular decision options promote various values as well as modification of the reasoning mechanism allowing for autonomous setting and realization of goals. This study is an expansion of the research results presented during the FedCSIS 2017 conference and further discussed in [16]. The proposed mechanism may serve as a formal

*Corresponding author. Email: tomasz.zurek@poczta.umcs.lublin.pl

foundation for the realization of the autonomous agent and allow for an integration of the logic-based reasoning mechanism discussed in [14] with machine learning approaches. A simple example of a partially autonomous mobile device illustrates how the model operates.

1.1. Contribution

This study is an expansion of the research results briefly discussed during the FedCSIS 2017 conference [16]. The paper is built on the background of our two previous papers [14,16]. Sections 3 and 7 briefly summarize paper [14], sections 4.1 and (partially) 4.2 summarize paper [16].

Our main contribution to the previous results can be summarized as follows:

- We present and discuss proofs of compatibility of our model (basics of which were presented in [16]) with the general model presented in [14] as well as with rationality postulates (Sections 4.2 and 4.3);
- We introduce an exemplary mechanism of conversion of the parameters of the autonomous device and its environment into the levels of promotion of values (Section 6);
- We discuss the problem of the discrete and continuous decision space (Section 8);
- We introduce and discuss the mechanism of integration of our model with machine learning approaches (Sections 9.1 and 9.2);
- We introduce a general and comprehensive argumentation framework allowing for making value- and goal-based decisions (Section 10);
- The model is illustrated by the example (Section 11).

The key point of the model from [14] was the possibility of making decision (establishing the desired state of affairs and performing the action leading to such a goal) even for the decisive situations which were not predicted in advance. This could be realized on the basis of the abstract goal (the minimal required level of promotion of values) allowing for the choice of the best practical goal (action leading to the desired state of affairs). The agent, in order to make such a decision, required the evaluation (in terms of the values from his abstract goal) of a possible decisions and the orderings of those evaluations (obtaining of which was an important disadvantage of this approach). Our new approach extends the previous one by introducing the possibility of computing of the levels of promotion of values and ordering between them.

2. THE MAIN IDEA

The framework presented in [14] (and summarized in the following) provides strong possibilities of modelling reasoning in autonomous systems; however, its practical implementation requires the solution of a number of major problems. In our opinion, a serious disadvantage of the system is the requirement of declaration of a great number of orders indicating the relations between levels to which various situations promote various values and sets of values. This task proves simple only in the case of decision

support systems which contain a relatively low number of decision options and values. Unfortunately, for the majority of real decision-making situations, these sets are too large. We find it indispensable to extend the model (1) by a mechanism allowing for an automatic evaluation of the levels to which given values are promoted by various situations (2) and by a mechanism which would determine the cumulative evaluation of the promotion of various values. The key assumption of our work is the idea that the estimation of the levels of promotion of values by the set of possible actions and state of affairs is easier to obtain than a declaration of orderings between them (in Zurek [14] we show that the number of orderings can be significantly greater than the number of all values, actions, and states of affairs).

We plan to attain our goal by introducing a numeric representation of the levels of value promotion. Such a method of representation of value levels (for single values as well as for value sets) will facilitate their comparison, leading to easier and quicker (searching order sets *O* and *OR* will not be necessary) reasoning. Additionally, for systems in which values are connected with the physical parameters of a device or the environment (e.g., the battery level of a mobile device, transfer speed, temperature, etc.) we propose a basic mechanism for automatic translation of physical units into the levels of promotion of values, allowing the possibility of its development and adjustment to evaluate other values difficult to measure, e.g., the degree of resemblance to the pattern, security level evaluation (such as in [17–20]), etc. The completion of the goal requires: (1) definition of the representation method for the level of value promotion including each value's weight, (2) development of the mechanism of determining the cumulative evaluation of the promotion of various values in a given situation, thus making possible the comparison of various options, (3) proposition of relevant modifications of the inference rules discussed in [14], (4) discussion of possibility of integration of our mechanism with machine learning based approaches, (5) verification of whether the developed mechanisms are not contradictory with either intuition or the formal properties of the model presented in [14].

Some comments: Many real-life decision situations are connected with a huge number of options to choose. Since the computer cannot distinguish between “reasonable” and “non-reasonable” options, in most decision systems we have to analyze all possibilities. Since their number can be high, such analysis can be a serious challenge (especially because the analysis requires detailed knowledge about all the options, relations between them and because the result of the analysis has a large impact on the quality of the decision). This is a problem which exists in most (if not all) formal decision-making mechanisms. If we look at the decision-making mechanisms from a very general point of view we can distinguish two main approaches to collect knowledge necessary to make the decision:

- 1 Introduce the knowledge directly to the system (knowledge-based approaches);
- 2 Train the system by the examples of previous decisions (machine learning-based approaches).

Both approaches have their own advantages and disadvantages: a knowledge-based systems require preparation and formulation of the entire necessary knowledge, but they allow for full control

over the decision-making process (and can justify the decision). The machine learning-based approaches require an extensive set of examples to train the system (the quality of the decision depends on the size and quality of a training set) and, since most of them work as a black-box, they cannot justify their decisions, which may exclude such a system from many decision areas (e.g., in the legal matter, which usually requires justification of the decision).

In order to illustrate the problem, we present the knowledge requirements of some models of knowledge-based decision-making models: in the classical PME model [2,3], there is necessity of declaration (or computation) of the level of utility and assignment of the level of probability to every option. The AATS+V model initially [7] required declaration of the promotion or demotion of each value for every possible transition between states of affairs (note that there can be much more transitions than states) but in the later works ([21] and others) it allows for calculation of promotion or demotion. In the later versions, the model allowed for the calculation not only the promotion or demotion, but also of the levels of promotion and demotion. In Amgoud *et al.* [22] there is a necessity of declaration of the orders between all possible options (which represents the relation of preference). Additionally, the preference relations between arguments pro and con each option is computed on the basis of the strength of the knowledge supporting each of the arguments (which also should be declared in advance). In the model of practical reasoning presented in [11] states of affairs (decision options) require declaration of the list of values supporting and demoting each of the states of affairs as well as preorders on states representing the preferences of the agents which allows for analysis of the value system and decision-making (the choice of a state of affairs). In the popular BDI model there is also the necessity of the analysis and comparison of all possible decision options.

In all the above examples, the amount of knowledge which should be declared is (similar to our approach) very high and the quality of the knowledge has a crucial character for the quality of decisions. The difference between our approach and the approaches presented above is in the kind of knowledge which should be introduced to the system. In our model we introduce the levels of promotion of particular values and weight functions instead of various orderings between states, actions, arguments, etc. There is an open problem: which kind of knowledge is easier to obtain? In our opinion, the answer to this question is not trivial and requires further research, especially regarding the possibility of utilisation of machine learning mechanisms to support obtaining such knowledge. In section 6 we introduce the general proposition on how machine learning mechanisms can help to estimate the level of promotion of particular values.

3. GVR MODEL

We will begin with a summarized discussion of the model of teleological reasoning from [14], further referred to as the GVR model.

Firstly, the naming convention will be presented:

- By upper case letters we denote sets;
- By lower case letters we denote propositions;
- Subscripts denote names of propositions;

- Superscripts denote names of sets;
- Greek letters denote functions;
- Other symbols will be defined later (except trivial logical and set-theory ones).

Definition 1. [State of affairs] Let $S = \{s_0, s_1, s_2, \dots\}$ be a finite, nonempty set of propositions. Each proposition represents one state of affairs. Let γ be a function which returns 1 if a given state of affairs is true and 0 if not. One and only one element from set S can be true: if $(\gamma(s_y) = 1)$, then $\forall_{s_x \in S: s_x \neq s_y} (\gamma(s_x) = 0)$. s_0 is the initial state of affairs. We assume that all state of affairs are separate. If we would like to model a case in which more than one state of affairs will be achievable simultaneously, then they should be divided into separate decisions: e.g., having two state of affairs s_a, s_b , an agent have such available state of affairs: $S = s_a, s_b, s_{a,b}$. Such an approach may cause a combinatorial explosion, but since set S contains only possible states, then the number of possibility will be significantly lower than 2^n .

Definition 2. [Actions] As an action we understand an activity which carries a transition from a certain state of affairs to another state of affairs. Actions will be represented by propositions from set $A = \{a_1, a_2, \dots, a_k\}$.

It is worth noticing that a particular action cannot be performed in every state of affairs. The set of all possible actions in all possible states of affairs we denote as AS ($AS \subseteq A \times S$). Set AS is a set of pairs $AS = \{as_{ij}, as_{k,w}, \dots\}$ in which $as_{ij} = \langle a_i, s_j \rangle$, $as_{k,w} = \langle a_k, s_w \rangle$ (the first subscript denotes the name of an action, the second subscript denotes the name of a situation). Each pair represents that a given action (e.g., a_i) can be performed in a given state of affairs (e.g., s_j). By AS^j (where $AS^j \subset AS$) we denote a set of actions possible to perform in a state of affairs s_j .

Function $\delta : AS \rightarrow S$ returns the result of performing an action a_i in a state of affairs s_j . By $\delta(as_{ij}) = s_y$ where $as_{ij} = \langle a_i, s_j \rangle$ we denote that the result of performing an action a_i in a state of affairs s_j is s_y . The presentation of our model will be illustrated by a simple running example (adopted from [14]):

Example 1. [Running example] John is in a bar and he is thirsty (state of affairs $s_{thirsty}$), he should drink a pint of lager (action $a_{drinkingLager}$), which will slake his thirst and leads to the state of affairs $s_{satisfied}$. By $as_{drinkingLager,thirsty}$ we denote the action-state pair (drinking lager in the state thirsty). Function $\delta(as_{drinkingLager,thirsty}) = s_{satisfied}$ shows that drinking lager in the state thirsty leads to the state satisfied.

Definition 3. [Transition process] Let $\varepsilon : AS \times S \rightarrow S$ be a partial function which represents performing an action a in a state of affairs s . If $\delta(as_{ij}) = s_y$ and $\gamma(s_j) = 1$ (the result of performing an action a_i in a state of affairs s_j is s_y), then performing $\varepsilon(as_{ij})$ causes changing $\gamma(s_j) = 0$ and $\gamma(s_y) = 1$.

Example 2. [Running example, cont.] If John, being thirsty ($\gamma(s_{thirsty}) = 1$), drinks a pint of lager (state action pair $as_{drinkingLager,thirsty}$), then, since drinking beer brings about satisfaction ($\delta(as_{drinkingLager,thirsty}) = s_{satisfied}$), the actual state of affairs will change:

$\gamma(s_{thirsty}) = 0$ and $\gamma(s_{satisfied}) = 1$.

Definition 4. [Situation] By a situation x_n we understand a particular state of affairs or a result of a particular action performed in a given state of affairs. A set of situations X is a union of sets of states of affairs and results of actions: $X = \{S \cup AS\}$. By $x_n \in X$ we denote an element from set X . By X^i we denote a set of situations available from a state of affairs s_j : $S^i = \{s_j \cup AS^i\}$ (we introduce the concept of situation because state of affairs and actions can become decision options and it is easier to use symbol X instead of $\{S \cup AS\}$)

Example 3. [Running example, cont.] In our case, the set of situations X contains actual state of affairs and one state-action pair: $X = \{s_{thirsty}, as_{drinkingLager, thirsty}\}$

Definition 5. [Values] Following Rohan [23], we have to separate the two meanings of the word value: a value may be understood as a concept or as a process.

- 1 Value as an abstract concept which allows for the estimation of a particular action or a state of affairs and influences one's behavior. V is a set of values: $V = \{v_1, v_2, \dots, v_n\}$
- 2 Value as a process of estimation of the level of extent to which a particular situation (state of affairs and / or action) x promotes a value v_i . By $v_i(x)$ we denote the extent to which x promotes a value v_i . By $V(X)$ we denote the set of all valuations of all situations.

It is important to emphasize that values can be promoted to a certain degree by a particular state of affairs or action: $v(as_{i,j})$ represents the degree to which a value v is promoted by a state of affairs which is the result of performing an action a_i in a state of affairs s_j .

Example 4. [Running example, cont.] We introduce two values: happiness ($V = \{v_{happiness}\}$) and frugality ($V = \{v_{frugality}\}$).

The levels of promotion of those values by both available situations ($X = \{s_{thirsty}, as_{drinkingLager, thirsty}\}$) are:

$$V(X) = \{v_{happiness}(s_{thirsty}),$$

$$v_{happiness}(as_{drinkingLager, thirsty}),$$

$$v_{frugality}(s_{thirsty}), v_{frugality}(as_{drinkingLager, thirsty})$$

By $V^i(X)$ we denote the set of all possible extents to which a value v_i from set V may be promoted by any possible situation $x \in X$.

Example 5. [Running example, cont.] For value $v_{happiness}$ the set $V^{happiness}(X)$ contains

$$\{v_{happiness}(s_{thirsty}), v_{happiness}(as_{drinkingLager, thirsty})\}$$

For value $v_{frugality}$ the set $V^{frugality}(X)$ contains

$$\{v_{frugality}(s_{thirsty}), v_{frugality}(as_{drinkingLager, thirsty})\}$$

A partial order $O_i = (\geq; V^i(X))$ represents the relation between extents to which values are promoted: $v_i(x_n) \geq v_i(x_m)$ means that $x_n \in X$ promotes a value v_i to a no less extent than $x_m \in X$. If $v_i(x_n) \geq v_i(x_m)$ and $v_i(x_m) \geq v_i(x_n)$, then extents to which a situation x_n and x_m promotes a value v_i are equal ($v_i(x_n) = v_i(x_m)$). If $v_i(x_n) \geq v_i(x_m)$ and $v_i(x_n) \neq v_i(x_m)$, then $v_i(x_n) > v_i(x_m)$.

Example 6. [Running example, cont.] We assume that:

$$O_{happiness} \text{ contains: } \{v_{happiness}(as_{drinkingLager, thirsty}) > v_{happiness}(s_{thirsty})\},$$

which means that drinking lager promotes happiness to a greater extent than staying thirsty.

$O_{frugality}$ contains

$$\{v_{frugality}(s_{thirsty}) >$$

$$v_{frugality}(as_{drinkingLager, thirsty})\}$$

which means that staying thirsty promotes frugality to a greater extent than drinking lager. In real-life reasoning people do not rely only on a comparison of the levels of promotion of one value; usually, they compare the levels of promotion of various values. Theoretically speaking, they are incompatible, but practically, people compare not only the levels of promotion of various values, but also the levels of promotion of various sets of values.

Definition 6. [Sets of values] By $V^Z \subset V$ we denote a subset (named Z) of a set of values V which consists of values: $v_i, v_j, \dots \in V^Z$.

By $V^{x_i} \subset V$ we will denote a set of values promoted by a situation X_i .

Definition 7. [The level of promotion of a set of values] By $V^Z(x_n)$ we denote a set of estimations of the levels of promotion of values constituting set V^Z by a situation $x_n \in X$. If $V^Z = \{v_z, v_t\}$, then $V^Z(x_n) = \{v_z(x_n), v_t(x_n)\}$.

By $V^{x_i}(x_i)$ (when the upper script contains the name of a situation) we denote a set of estimations of the levels of promotion of all values promoted by a situation $x_i \in X$.

Example 7. [Running example, cont.] Since both available situations in our example promote the same values (but to different extents), then $V^Z(s_{thirsty}) = V^{s_{thirsty}}(s_{thirsty}) =$

$$V^Z(as_{drinkingLager, thirsty}) =$$

$$V^{as_{drinkingLager, thirsty}}(as_{drinkingLager, thirsty})$$

Definition 8. [Value-extent preference] A partial order $OR = (\triangleright; 2^{V(X)})$ represents a preference relation between various values and various sets of situations: $V^Z(x_n) \triangleright V^Y(x_m)$ means that the extent to which values from set V^Z are promoted by a situation x_n is preferred to the extent to which values from set V^Y are promoted by a situation x_m . Properties of the OR relation:

- Relation OR is a strict partial order, hence it is irreflexive, asymmetric, and transitive.
- If V^Z is a set of values promoted by a situation x_1 ($V^Z \subseteq V^{x_1}$) and $V^X \subseteq V^Z$, then:
 $V^X(x_1) \triangleright V^Y(x_2) \Rightarrow V^Z(x_1) \triangleright V^Y(x_2)$.

Example 8. [Running example, cont.] By $V^Z(x_{as_{drinkingLager, thirsty}}) \triangleright V^Z(x_{s_{thirsty}})$ we denote that John prefers the extents to which drinking lager promotes both values ($v_{happiness}(as_{drinkingLager, thirsty})$ and $v_{frugality}(as_{drinkingLager, thirsty})$) to the extents to which staying thirsty promotes both values ($v_{happiness}(s_{thirsty})$ and $v_{frugality}(s_{thirsty})$), which means that he prefers the drinking lager to staying thirsty.

Note that John prefers beer even though frugality is promoted to the lower extent than it is in staying thirsty. It reflects a case in which someone have to balance and accept decision in which some values are promoted to the maximal at the expense of other values.

How to determine whether the extents to which all values promoted by one situation are preferred to the extents to which all values are promoted by another situation? In the simplest case, we can assume some properties connecting orders O and OR , which may allow for reasoning about preferences; however, we have to note that order O is not sufficient to infer all necessary orders OR . Firstly, we have to assume some basics:

- By $V^{x_1 \geq x_2}$ we denote values which x_1 promotes to a no less extent than x_2 : $V^{x_1 \geq x_2} = \{v_i \mid v_i(x_1) \geq v_i(x_2)\}$.
- By $V^{x_2 > x_1}$ we denote values which x_2 promotes to a higher extent than x_1 : $V^{x_2 > x_1} = \{v_i \mid v_i(x_2) > v_i(x_1)\}$.
- By $V^{x_1 \setminus x_2}$ we denote values which are promoted by x_1 , but not promoted by x_2 .
- By $V^{x_2 \setminus x_1}$ we denote values which are promoted by x_2 , but not promoted by x_1 .
- $V^{x_2 \supseteq x_1} = \{V^{x_2 \geq x_1} \cup V^{x_2 \setminus x_1}\}$
- $V^{x_1 \supseteq x_2} = \{V^{x_1 \geq x_2} \cup V^{x_1 \setminus x_2}\}$
- For the sake of consistency, I also assume that the order declarations are rational, i.e., there are no declarations leading to situations in which OR includes both $V^A \triangleright V^B$ and $V^B \triangleright V^A$ (e.g., there are no declarations like: $V^A \subseteq V^{x_1}$, $V^B \subseteq V^{x_2}$, $V^A(x_1) \triangleright V^{x_2}(x_2)$ and $V^B(x_2) \triangleright V^{x_1}(x_1)$).

Then we formulate relations between orders O and OR :

- The first and basic mechanism connecting orders O and OR is very simple: on the basis of the ordering of extents to which a given value is promoted by two situations, preferences between these extents can be derived:

$$v_i(x_1) > v_i(x_2) \Rightarrow \{v_i(x_1)\} \triangleright \{v_i(x_2)\} \quad (1)$$

Example 9. [Running example, cont.] This can be illustrated by a simple example:

Since

$$v_{\text{happiness}}(as_{\text{drinkingLager}, \text{thirsty}}) >$$

$$v_{\text{happiness}}(s_{\text{thirsty}}) \text{ then:}$$

$$\{v_{\text{happiness}}(as_{\text{drinkingLager}, \text{thirsty}})\} \triangleright$$

$$\{v_{\text{happiness}}(s_{\text{thirsty}})\}$$

Because drinking lager promotes $v_{\text{happiness}}$ to higher extent than staying thirsty, then in the light of this value drinking beer is preferred to staying thirsty.

- The next one allows to derive preferences between the sets of values promoted by given situations: if every value promoted by a situation x_2 is also promoted by a situation x_1 , x_1 promotes it to a no less extent than x_2 , and at least one of these values is promoted by x_1 to a higher extent than by x_2 , then the extents

to which values are promoted by x_1 are preferred to x_2 :

$$(\forall_{v_z \in V^{x_2}} (v_z(x_1) \geq v_z(x_2))) \wedge \quad (2)$$

$$\exists_{v_t \in V^{x_2}} (v_t(x_1) > v_t(x_2)) \Rightarrow V^{x_1}(x_1) \triangleright V^{x_2}(x_2)$$

Example 10. [Running example, cont.] In order to illustrate the above we have to add a new value: social relations ($v_{\text{socialRelations}} \in V$). Both available situations promote this value to the extents:

$$v_{\text{socialRelations}}(as_{\text{drinkingLager}, \text{thirsty}}),$$

$$v_{\text{socialRelations}}(s_{\text{thirsty}}), \text{ where}$$

$$(v_{\text{socialRelations}}(as_{\text{drinkingLager}, \text{thirsty}}) =$$

$$v_{\text{socialRelations}}(s_{\text{thirsty}})) \in O_{\text{socialRelations}} \text{ (which means that both situations promote this value to the same level).}$$

If we assume that there are only 2 values in set V ($v_{\text{socialRelations}}, v_{\text{happiness}} \in V$), then sets $V^{as_{\text{drinkingLager}, \text{thirsty}}}$ and $V^{s_{\text{thirsty}}}$ are equal and contain only those 2 values.

Since both values in $V^{s_{\text{thirsty}}}$ are promoted by $as_{\text{drinkingLager}, \text{thirsty}}$ at least to the same level ($v_{\text{socialRelations}}$ to the same level, $v_{\text{happiness}}$ to the higher extent) then

$$V^{as_{\text{drinkingLager}, \text{thirsty}}}(as_{\text{drinkingLager}, \text{thirsty}}) \triangleright$$

$$V^{as_{\text{drinkingLager}, \text{thirsty}}}(as_{\text{drinkingLager}, \text{thirsty}})$$

- If a set of values promoted by a situation x_2 (V^{x_2}) consists of two sets: values which are promoted by x_1 to a higher extent than by x_2 ($V^{x_1 \geq x_2}$) and other values which are promoted by x_2 ($V^{x_2 \setminus x_1}$), and if there exists a set of values $V^s \in V^{x_1 \setminus x_2}$ whose promotion by x_1 is preferred to levels to which values from a set $V^{x_2 \setminus x_1}$ are promoted by a situation x_2 , then the extent to which values are promoted by a situation x_1 is preferred to the extent to which values are promoted by a situation x_2 :

$$(V^{x_2} = \{V^{x_1 \geq x_2} \cup V^{x_2 \setminus x_1}\}) \wedge \exists_{V^s \subseteq V^{x_1 \setminus x_2}} (V^s(x_1) \triangleright V^{x_2 \setminus x_1}(x_2)) \Rightarrow V^{x_1}(x_1) \triangleright V^{x_2}(x_2) \quad (3)$$

Example 11. [Running example, cont.] In order to illustrate the above we assume that drinking lager promotes social relations and happiness ($V^{as_{\text{drinkingLager}, \text{thirsty}}} = \{v_{\text{socialRelations}}, v_{\text{happiness}}\}$, note that drinking lager does not promote frugality) and staying thirsty promotes social relations and frugality ($V^{s_{\text{thirsty}}} = \{v_{\text{socialRelations}}, v_{\text{frugality}}\}$, note that happiness is not promoted by staying thirsty). Additionally we assume that

$$(v_{\text{socialRelations}}(as_{\text{drinkingLager}, \text{thirsty}}) =$$

$$v_{\text{socialRelations}}(s_{\text{thirsty}})) \in O_{\text{socialRelations}} \text{ and}$$

$$(v_{\text{happiness}}(as_{\text{drinkingLager}, \text{thirsty}}) \triangleright$$

$\{v_{\text{frugality}}(s_{\text{thirsty}})\} \in OR$, which can be interpreted that John prefers level of promotion of happiness by drinking beer to the level of promotion of frugality by staying thirsty.

On the basis of the above assumptions and formula 3 we can say that

$$V^{as_{\text{drinkingLager}, \text{thirsty}}}(as_{\text{drinkingLager}, \text{thirsty}}) \triangleright$$

$$V^{as_{\text{drinkingLager}, \text{thirsty}}}(as_{\text{drinkingLager}, \text{thirsty}})$$

The above can simply justified: Since social relations are promoted to the equal level by both situations and drinking beer is preferred to staying thirsty in terms of happiness and frugality, then the agent prefers drinking lager.

- In any other case, it is impossible to derive a preference order.

As we have noticed earlier, orders from set O do not allow for the inference of all necessary orders from set OR , hence some of them must be declared in advance. The main challenge regarding the modeling of the above relation of preference lies in the necessity of declaration of the number of orders between the extents to which situations promote all subsets of values from set V .

3.1. Goals

As it has already been noticed, we need to draw a distinction between a few different kinds of goals [14]:

Definition 9. [Abstract goals] Abstract goals are goals represented by the minimal extents to which a particular situation promotes a given set of values (in our model the valuation of any target state of affairs can be considered only with the valuation of the action which leads to it, hence by a situation we will understand a particular action performed in a particular state of affairs):

- $GA = \{ga_1, ga_2, \dots\}$ - a set of abstract goals.
- By $v_n \min(ga)$ we denote the minimal extent to which the promotion of a value v_n satisfies a goal ga .
- By $v_n(x_1) \geq v_n \min(ga)$ we denote that a goal ga is satisfied by a situation x_1 with respect to a value v_n .
- By $v_n \in ga$ we denote that the minimal extent of a given value v_n is declared in a goal ga : $v_n \in ga \leftrightarrow \exists v_n \min(ga)$.

Definition 10. [Abstract unreachable goals] Abstract unreachable goals are abstract goals where the agent desires one value to be promoted as much as possible, while other values to be promoted no less than to a certain extent:

- $GUA = \{gua_1, gua_2, \dots\}$ - a set of abstract unreachable goals.
- By $v_n \min(gua)$ we denote the minimal extent to which the promotion of a value v_n satisfies a goal gua .
- Function $\omega : GUA \rightarrow V$ returns the value which should be promoted to the maximal possible extent. A value $v_m : \omega(gua) = v_m$ will be called the key value of an abstract unreachable goal gua . Note that value v_m is also the ordinary goal of a goal gua ($v_m \in gua$) and it has its own minimal extent to which it should be promoted.

Definition 11. [Material goals] Material goals are particular situations which satisfy given abstract goals.

- $GM = \{gm_1, gm_2, \dots\}$ - a set of material goals.
- A material goal is a particular action performed in a given state of affairs where: $gm_k = (as_{t,m})$, while $as_{t,m} \in AS$.

- By $sat(gm_k, ga_l)$ we denote that a material goal gm_k satisfies an abstract goal ga_l and is possible to achieve:
 $sat(gm_k, ga_l) \leftrightarrow (\forall_{v_i \in ga_l} (v_i(as_{t,m}) \geq v_i \min(ga_l)) \wedge as_{t,m} \wedge (gm_k = (as_{t,m}) \in AS \wedge \gamma(s_m) = 1)$.

Definition 12. [Practical goal] A practical goal is a material goal which is achievable, which satisfies the agent's abstract goal, and which the agent is going to reach. A practical goal will be denoted as gp . The agent can only have one practical goal at a time.

Example 12. [Running example, cont.] Let's assume that John has one abstract goal $ga = \{v_{happiness} \min(ga), v_{frugality} \min(ga)\}$ expressed by two minimal acceptable levels of promotion of two values: happiness and frugality.

We have three decision options $as_{drinkingLager, thirsty}$, $as_{drinking2Lagers, thirsty}$, $s_{thirsty}$ (drinking beer, drinking 2 beers, staying thirsty). We assume that:

$O_{happiness}$ contains: $\{v_{happiness}(as_{drinkingLager, thirsty}) >$

$v_{happiness} \min(ga)\}$,

$\{v_{happiness}(as_{drinking2Lagers, thirsty}) >$

$v_{happiness} \min(ga)\}$,

$\{v_{happiness}(s_{thirsty}) <$

$v_{happiness} \min(gua)\}$,

which means that drinking lager (or two) promotes happiness to the satisfactory levels, while staying thirsty not.

$O_{frugality}$ contains

$\{v_{frugality}(s_{thirsty}) >$

$v_{frugality} \min(gua)\}$

$\{v_{frugality}(as_{drinkingLager, thirsty}) >$

$v_{frugality} \min(ga)\}$

$\{v_{frugality}(as_{drinking2Lagers, thirsty}) <$

$v_{frugality} \min(ga)\}$

which means that staying thirsty or drinking one lager promotes frugality to the satisfactory levels, while drinking 2 beers not (two beers are too expensive).

Since our agent has abstract goal ga then we can derive that only one option (drinking lager) promotes all values from goal ga to the satisfactory level, thanks to which $as_{drinkingLager, thirsty}$ becomes a material and practical goal.

Let's assume that John has an unreachable abstract goal $gua = \{v_{happiness} \min(gua), v_{frugality} \min(gua)\}$ in which $\omega(gua) = v_{happiness}$. We assume that thresholds in gua are the same as in ga and the key value of gua is happiness (happiness should be promoted as much as it is possible having other values promoted at least to the satisfactory level).

3.2. Inference Mechanism

Zurek [14] discusses a number of inference rules as well as a simple argumentation framework which allows for reasoning about goals

and making decisions concerning the fulfilment of these goals. Our paper includes a discussion of several inference rules (Section 7) and a slightly modified argumentation framework (Section 10). An exhaustive discussion of the inference rules' structure and properties is included in [14].

4. NUMERIC REPRESENTATION OF THE LEVELS OF VALUE PROMOTION BY PARTICULAR SITUATIONS

The underlying objective of this section is to discuss a new semantics (basics of which were presented in [16]) for the model described in [14], which would allow for a numeric representation of the levels to which given situations promote various values. This task consists of two main parts: first, a proposal and discussion of the numeric method of representation of the level to which given values are promoted; then the development of the mechanism of determining this level, of determining the cumulative evaluation of value sets, the mechanism of comparing them (equivalents of the *O* and *OR* relations), a discussion of the properties of the proposed semantics and verification of whether the new propositions are not contradictory to intuition or the formal assumptions of the model from [14].

4.1. Numeric Representation of Levels to Which Various Situations Promote Particular Values

In the model presented in [14], any level to which a given value (e.g., v_i) is promoted by a given situation (e.g., x_j) is presented as $v_i(x_j) \in V(X)$ (without a definition of the kind and range of values which can be taken). The relations between the levels to which a given value (v_i) is promoted by situations x_1 and x_2 are regulated by a partial order O_i . In our model (firstly presented in [16]), we assume that $v_i(x_j)$ will be expressed with a number from the range $\langle 0; 1 \rangle$, where $v_i(x_j) = 0$ means that a value v_i is not promoted by a situation x_j , a value $v_i(x_j) = 1$ means that a value v_i is promoted by x_j to the maximal possible level, though we assume that the accomplishment of the maximal fulfilment of the value is not possible—the value may be promoted very close to the maximal value, but cannot reach it. Such representation allows us to compare the levels to which the same value is promoted by various situations (order *O* will be a total order).

Assuming a numeric representation of the level of promotion of values one cannot miss the fact that not all values are equally important to each user. The simplest solution would be to assign some weight to each value (from the range $\langle 0; 1 \rangle$), though this proposition is a far cry from the way humans reason. In real decision-making, a person does not assign constant weights to various values: e.g., for a mobile device user wishing to run a web application, the battery level seems less important than transfer speed, but if the battery is nearly exhausted, this fact will become the critical value for the user. In most cases, the weight of a given value depends on the user's preferences, external factors, and the level to which the value is promoted (like in the abovementioned example with the battery). This phenomenon is noticed by the author of [24]. The numeric representation of values in the legal decision-making was discussed in [25]. The authors introduced a system dedicated strictly to the

legal reasoning with factors. Although some common elements (the numbers represent the *weight* of value), this model is different to the our one. Firstly, because it was designed strictly to the legal reasoning where are only two possible and strictly opposite decision options (plaintiff-defendant), secondly because the authors of [25] use factors as a case representation (which are absent in our system), thirdly because the system uses different calculus than we are (e.g., we have a dedicated function Θ allowing for cumulative evaluation of different values, where in [25] pro-plaintiff values' weights are added and subtracted from pro-defendant values' weights). Obviously there are some general similarities, e.g., the numerical representation of values allows (similarly to our approach) for comparison of cases (or decisions) which promote different values. We also agree with the authors of [25] that not all values are equally important. This is the reason of introducing the functions of weight (Ω_i). The reason of introducing weight function comes from the observation that the importance of a given value can depend on the level of promotion of this value (and, possibly, other values).

Definition 13. (Function of the weight [16]) Let $\Omega_i : v_i(x_j) \rightarrow \langle 0; 1 \rangle$ be a function of the weight referring to a value v_i . We assume that every function of the weight in the range $\langle 0; 1 \rangle$ will be a continuous and increasing function (this assumption is indispensable for the preservation of the features of the GVR model). For every value $v_i \in V$ exists maximally one function Ω_i . Let Ω be a set of weights' functions. By $vo_i(x_j) = \Omega_i(v_i(x_j))$ we will denote the level of promotion of a value v_i by situations x_j taking into account weight Ω_i . A value $vo_i(x_j)$ will denote the relative level to which a situation x_j promotes a value v_i . Let $VO(x) = \{vo_i(x_j) | vo_i(x_j) = \Omega_i(v_i(x_j)) \wedge v_i(x_j) \in V(X)\}$ be the set of all values vo_i in all situations.

$VO^Z(X)$ will denote any subset $VO(X)$ called *Z*.

The most basic kind of function Ω is a linear function $\Omega_i(v_i(x)) = a(v_i(x))$, where a is a constant from the range $\langle 0; 1 \rangle$; it is, however, possible to define more complex functions which would better express the relative preferences between values. For example, in the above presented battery case we can use such a function:

$$\Omega_{batt}(v_{batt}(x)) = \begin{cases} 0.1v_{batt}(x) & \text{where } v_{batt}(x) \leq 0.2 \\ 1.6v_{batt}(x) - 0.3 & \text{where } 0.2 < v_{batt}(x) \leq 0.45 \\ 0.1v_{batt}(x) + 0.45 & \text{where } v_{batt}(x) > 0.45 \end{cases}$$

If the level of promotion of the battery charge (v_{batt}) is above than 45%, then the changes of its value do not influence strongly the relative level of promotion of the value (vo_{batt}). If the level of promotion of battery charge decreases below 45%, then the relative level of promotion decreases rapidly almost to 0, which means that such a low level of battery charge becomes critical. The above example presents a situation where for the low level of battery charge (below 0.2), the weight of the v_{batt} value becomes very low (close to zero), which strongly decreases the cumulated evaluation of a given situation (see Section 4.2) If for a given value v_i it is impossible to determine function Ω_i , then we can assume its default for $\Omega_i(v_i(x)) = a(v_i(x))$, where $a = 0.5$.

4.2. Cumulative Evaluation of the Level of Promotion of a Value Set by a Given Situation

In the model discussed in [14], a given situation may promote various values to various extents. It is represented by set $V^Z(x_n)$, where V^Z is a subset (named Z) of value set V . By $V^Z(x_n)$ we denote a set of estimations of the levels of promotion of values constituting set V^Z by a situation $x_n \in X$. If $V^Z = \{v_z, v_t\}$, then $V^Z(x_n) = \{v_z(x_n), v_t(x_n)\}$. The GVR model also introduces the order relation OR between promotion sets to which various values are promoted by various situations (def. 8). Properties of the OR order are discussed in [14].

Since our model assumes that the grounds for evaluation are relative levels to which particular values are promoted by situations, in our model the correspondent of set $V(X)$ will be set $VO(X)$ and its subsets [16].

For the realization of value-based reasoning to be possible, it is indispensable to define the correspondent of order OR for the new semantics. Introducing the numeric representation of the level to which a given situation promotes a given value ($v_i(X_n)$) and the weight function Ω provides the possibility to develop a mechanism able to determine the cumulative evaluation of the promotion of a value set by a given situation.

Firstly, we assume that the cumulative evaluation of the level to which a given value set will be promoted by a given action will be a number from the range $\langle 0; 1 \rangle$, where 0 is interpreted as the minimal level of promotion and 1 is interpreted as the maximal level of promotion (impossible to attain). Such a relation will enable the comparison of various situations, including those which promote different values.

Definition 14. (Function Θ [16]) Let $\Theta : VO^Z(x) \rightarrow \langle 0; 1 \rangle$ be a function returning the cumulative evaluation of the level to which a situation x promotes a value set V^Z .

If:

- $V^Z = \{v_1\}$ then $\Theta(VO^Z(x)) = vo_1(x)$;
- $V^Z = \{v_1, v_2\}$, then $\Theta(VO^Z(x)) = vo_1(x) + vo_2(x) - vo_1(x)vo_2(x)$;
- $V^Z = \{v_1, v_2, v_3\}$, then the value returned by function Θ is determined in the following manner: first, we determine $\Theta(VO^{v_1, v_2}(x))$ for $V^{v_1, v_2} = \{v_1, v_2\}$, then we determine $\Theta(VO^Z(x)) = \Theta(VO^{v_1, v_2}(x)) + vo_3(x) - \Theta(VO^{v_1, v_2}(x))vo_3(x)$;
- In case of a higher number of values in set V^Z , the cumulative value $\Theta(VO^Z(x))$ is determined analogously to the previous case.

The main idea of the function Θ is to introduce the joint evaluation of a particular situation. Such evaluation allows for comparison of various situations (even those which promote different values). For example, we have two options to choose: to stay at home and work (x_{home}) or to go to a party (x_{party}). We know that staying at home promotes productivity to a certain level (e.g., $vo_{prod} = 0.5$) and saves our money ($vo_{money} = 0.4$). Going to the party promotes fun ($vo_{fun} = 0.6$) and allows for contact with friends ($vo_{friends} = 0.4$).

The cumulative evaluation of staying at home is

$$\Theta(VO^{home}(x_{home})) = vo_{prod}(x_{home}) + vo_{money}(x_{home}) - vo_{prod}(x_{home})vo_{money}(x_{home}) = 0.7$$

The cumulative evaluation of going to the party is

$$\Theta(VO^{party}(x_{party})) = vo_{prod}(x_{party}) + vo_{money}(x_{party}) - vo_{prod}(x_{party})vo_{money}(x_{party}) = 0.76$$

The cumulative evaluation of the values promoted by a party is higher than the cumulative evaluation of values promoted by staying at home (note that the basis of the evaluation are the relative levels of the values' promotion), which can be interpreted as the situation in which a particular agent prefers going to the party to staying at home. The concept of the function Θ is a simulation of a real-life problem in which we have to compare situations promoting different and sometimes difficult to compare values.

The properties of function Θ :

- 1 The value returned by function Θ is independent of the order in which particular values from V^Z are reviewed (proof can be found in the Appendix A.0.1).
- 2 Function Θ is monotonic (here as monotonic we understand not only its being non-decreasing, but also the fact that adding a new value which promotes a given situation increases the cumulative evaluation $\Theta(VO^Z(x))$: If $VO^Z(x)$ is a set of relative levels of promotion of the values from set V^Z in a situation x and if a situation x also promotes a value $v_j \notin V^Z$, then $\Theta(VO^{Z+v_j}(x)) > \Theta(VO^Z(x))$ (the cumulative evaluation of the level of promotion of the values from V^Z and of a value v_j by a situation x will be higher than the evaluation of the level of promotion of the values from V^Z). Proof can be found in Appendix A.0.2.

As we have already noticed, values should not be treated equally (they are not all equally important), and therefore we proposed a set of weight functions Ω . Based on that, we assume that in our model the equivalent of order OR will be order ORO :

Definition 15. (Value-extent-weight preference [16]) A total order $ORO = (\geq; 2^{VO(X)})$ represents a preference relation between various values, their weight functions, and various sets of situations. $VO^Z(x_n) \geq VO^Y(x_m)$ means that the relative extent to which values from set V^Y are promoted by a situation x_m is not preferred to the relative extent to which values from set V^Z are promoted by a situation x_n .

We assume that $\Theta(VO^Z(x_n)) \geq \Theta(VO^Y(x_m)) \Rightarrow VO^Z(x_n) \geq VO^Y(x_m)$.

Since function Θ allows to determine the cumulative evaluation of the promotion of all value sets and all situations, then on the basis of the above formula one can unambiguously determine order ORO . Hence in practice there is no necessity to declare relations in order ORO .

If $VO^Z(x_n) \geq VO^Y(x_m)$ and $VO^Y(x_m) \geq VO^Z(x_n)$, then we say that $VO^Z(x_n) \equiv VO^Y(x_m)$.

If $VO^Z(x_n) \geq VO^Y(x_m)$ and $VO^Y(x_m) \not\geq VO^Z(x_n)$, then we say that $VO^Z(x_n) \blacktriangleright VO^Y(x_m)$.

4.3. Relations Between Orders O and ORO

Zurek [14] (as well as this paper's sections 3, def. 8 and formulae 1, 2, 3) introduce the properties of order OR and the relations between orders O and OR . These relations may be used to determine unknown relations of order OR . Since in our model the bases for the evaluation of a situation are relative (considering the weight function) levels to which values are promoted, the equivalent of order OR for the numeric representation of the levels of value promotion will be order ORO . To preserve the features of the mechanism from [14], we have to presume that order ORO will have the same properties as OR and will preserve the relations between OR and order O . We assumed that functions in Ω are continuous and increasing, due to which we know that if $v_i(x_j) > v_i(x_k)$, then $\Omega_i(v_i(x_j)) > \Omega_i(v_i(x_k))$. The proof is trivial.

Knowing the properties of the functions from the Ω set, we will examine the behavior of function Θ and, in consequence, the features of order ORO . Firstly, we will investigate whether order ORO has features similar to OR (def. 8).

- 1 Since order ORO is a total order, it fulfills the assumptions of a partial order. The only difference is that order ORO is not a strict order.
- 2 The other feature of order OR is preserved: If V^Z is a set of values promoted by a situation x_1 ($V^Z \subseteq V^{x_1}$) and $V^X \subseteq V^Z$, then $V^X(x_1) \triangleright V^Y(x_2) \Rightarrow V^Z(x_1) \triangleright V^Y(x_2)$. Since function Θ which supports order ORO is monotonic (see 2 property of function Θ), this condition is fulfilled (proof can be found in appendix A.0.3).

Furthermore, we will analyze whether the relations between order O and ORO are the same as the relations formulated in [14] (recalled in this paper in Section 3, formulae 1, 2, 3) between orders O and OR (proofs can be found in Appendix A.0.4)

- 1 The first relation will be fulfilled if $v_i(x_1) > v_i(x_2) \Rightarrow \{v_{o_i}(x_1)\} \geq \{v_{o_i}(x_2)\}$.
- 2 The second relation will be fulfilled if $(\forall_{v_z \in V^{x_2}} (v_z(x_1) \geq v_z(x_2)) \wedge \exists_{v_i \in V^{x_2}} (v_i(x_1) > v_i(x_2)) \Rightarrow VO^{x_1}(x_1) \geq VO^{x_2}(x_2)$
- 3 The third relation will be introduced differently than in [14]. It will be fulfilled if $(V^{x_2}(x_2) = \{V^{x_1 \geq x_2}(x_2) \cup V^{x_2 \geq x_1}(x_2)\})$
 $(V^{x_1}(x_1) = \{V^{x_2 \geq x_1}(x_1) \cup V^{x_1 \geq x_2}(x_1)\})$
 $\exists_{VO^s(x_1) \subseteq VO^{x_1 \geq x_2}(x_1)} (VO^s(x_1) \blacktriangleright VO^{x_2 \geq x_1}(x_2)) \Rightarrow VO^{x_1}(x_1) \blacktriangleright VO^{x_2}(x_2)$

5. GOALS

The four kinds of goals defined in [14] (and discussed in Section 3.1 of this paper) remain the same. The only changes are caused by the fact that the level of promotion of values is expressed in numbers, and therefore the threshold values: $(v_n \min(ga))$ and $(v_n \min(gua))$ will also be numbers from the range $(0; 1)$. These values may be declared directly by the user or determined from function Φ .

Although we focused on the 4 main kinds of goals presented in [14], we do not exclude the possibility of adding other types of goals. For example, should a need arise for a more sophisticated kind of goal allowing for the balancing between the levels of promotion of values, we can introduce a new kind of abstract goal in which the threshold values $(v_n \min(ga))$ are not fixed, but they depend on the levels of promotion of other values (such kind of a goal gives the possibility for representing the situation in which we allow for a small demotion of the level of promotion of one value in order to have a strong promotion of the other value). The discussion of such new types of goals will be presented in future work.

6. DECISION SPACE

Similarly to other argumentation-based mechanisms of practical reasoning and decision-making, our model assumes a discrete space of available events. When making a decision, the agent is faced with a known number of available options. It is worth noticing, that In our model we assume that all actions are certain (i.e., the result of a given action is always certain), the problem of uncertain results of actions and the behavior of other agents, was discussed in [26]. Adding such a possibility to our model is possible, but requires further developments and will be discussed in the future works.

In the presented model, the decision-making options available for the state of affairs s_j have been described in the form of a set of situations X^j which allows for maintaining the current state of affairs (s_j) or performing an action included in the set of available actions (AS^j). For a closer analysis of the decision-making process, we need to differentiate between the decision space (i.e., the total number of options available to the agent) and the respective decision variables. To date, our model provided the agent with a set AS^j of actions available in a given situation s_j . Building upon the same, we now wish to expand our model in such a way that the respective actions in set AS^j will be combinations of various values of the particular decision variables.

Definition 16. [Decision variables] Let $\vec{x}_i = (d_1, d_2, \dots, d_n)^T$ be a vector of decision variables referred to as i and corresponding to an action taken in accordance with the parameters expressed as $(d_1, d_2, \dots, d_n)^T$. Let \vec{X} be a set of decision variable vectors. Vector \vec{x}_i represents the selection of a certain specific action with the parameters expressed as (d_1, d_2, \dots, d_n) , with one exceptional case entailing the action of taking no action whatsoever and maintaining the current state of affairs. Let D_1 be the domain (set of possible values) of variable d_1 , D_2 the domain of variable d_2 , etc. If the domain for each variable is finite and discrete, the set \vec{X} (decision space) will be a subset of the Cartesian product of variable domains d_1, \dots, d_n : $\vec{X} \subseteq D_1 \times D_2 \times \dots \times D_n$. In other words, any situation from the X^j can be expressed as a vector from set \vec{X} (note that not every combination of decision parameters represent a possible decision, some combinations are not available). In the subsequent parts of these deliberations, whenever it is necessary to refer directly to the vector of decision variables, the same will be denoted as \vec{x}_i , otherwise, we will refer to actions ($as_{i,j}$) and situations (x), bearing in mind, however, that the same relate to a vector of decision variables.

Example 13. [Running example cont.] The above definition can be illustrated on an example. Let us assume that the following decision-making situation occurs: we are in a pub (state of affairs s_{pub}), feel like having a glass of beer, and can choose between two types of beverage (lager and stout) as well as two glass sizes: 0.5 or 0.3 liters. We also have the choice of taking no action whatsoever. Given the above, we are therefore dealing with two decision variables: d_{beer} and d_{glass} , where the domain of d_{beer} is $D_{beer} = \{lager, stout, nothing\}$, and the domain of d_{glass} is $D_{glass} = \{0.5, 0.3, nothing\}$. Hence, the decision space is as follows:

$$as_{pub,(lager,0.5)}^T, as_{pub,(stout,0.5)}^T, as_{pub,(lager,0.3)}^T, as_{pub,(stout,0.3)}^T, as_{pub,(nothing,nothing)}^T.$$

If the number of decision variables and the number of values in the domain of each variable are high, then we can face the problem of combinatorial explosion of the number of options to choose (the number can be decreased because not all combinations of decision options are available, e.g., there is no $as_{pub,(nothing,0.3)}^T$ and $as_{pub,(nothing,0.5)}^T$ in AS). Such a situation can be the origin of some computational problems, but (as we notice in the comments to Section 2) this is a problem which appears in most (if not all) formal methods of decision-making: the rational choice of option requires the analysis of all possibilities, no matter how we describe them (as decision vector, state of affairs, transition between states, actions, etc.) In Zurek and Mokkas [16], we proposed a very simplified model for the transformation of the physical quantity pv_i into a certain level of promotion of a given value v_i by employing the Φ_i function. The Φ_i function allows the value of one physical quantity (or any given decision parameter) to be converted into the level of value promotion. Unfortunately, the same does not account for that fact that any decision-making situation entails numerous decision variables (expressed in our model by the variable vector). Therefore, because the model proved too simplistic to be applicable to a more complex decision-making reality, we expanded it as described below.

Definition 17. Let $\Phi_i : \vec{X} \rightarrow \langle 0; 1 \rangle$ be a function that transforms the vector of decision variables into the level of promotion of value v_i : $v_i(\vec{x})$. Let Φ be the set of transformation functions. Our model can make use of several different transformation functions which can be additionally declared, depending on the nature of the values.

If action $as_{k,j}$ is represented by decision vector \vec{x} , then $\Phi_i(as_{k,j}) = \Phi_i(\vec{x})$.

The thus determined level of promotion can be used for further inference.

The above definition can be illustrated by an example:

Example 14. [Running example, cont.] Let's assume that John have such decision options

$$as_{pub,(lager,0.5)}^T, as_{pub,(stout,0.5)}^T, as_{pub,(lager,0.3)}^T, as_{pub,(stout,0.3)}^T, as_{pub,(nothing,nothing)}^T.$$

Let's assume function $\Phi_{happiness}(\vec{X})$:

$$\Phi_{happiness}(\vec{X}) = \begin{cases} 0 & \text{where } as_{pub,(nothing,nothing)}^T \\ x_2 & \text{where } as_{pub,(lager,x_2)}^T \text{ and } x_2 < 0.8 \\ 1.2 * x_2 & \text{where } as_{pub,(stout,x_2)}^T \text{ and } x_2 > 0.8 \\ 0.3 & \text{elsewhere} \end{cases}$$

The above function shows that staying without drinking does not promote John's happiness, he prefers to drink more than less, but drinking more than 0.8 liter (or drinking unknown drink) promotes happiness to 0.3 only. John also prefers stout to lager. Drinking 0.5 liter of stout promotes his happiness to 0.6, while drinking 0.5 liter lager promotes happiness to 0.5. This example shows that in a new, unexpected, situation (e.g., there is a new menu in the pub in which beer is served in different glasses: 0.2 and 0.6 l.) John can assign the level of promotion of value happiness to the possible decisions.

Under real-world circumstances, it is not always possible to determine separate functions Φ and Ω for each respective value, particularly if these are obtained inductively (with the use of machine learning mechanisms) rather than predeclared. In such a case we can assume the existence of a single function combining the properties of functions Φ_i and Ω_i :

Definition 18. Let $\Lambda_i : \vec{X} \rightarrow \langle 0; 1 \rangle$ be a function that transforms the vector of decision variables into the relative level of promotion of value v_i : $vo_i(\vec{x})$. Let Λ be the set of transformation functions. Our model can make use of several different transformation functions which can be additionally declared, depending on the nature of the values.

Let us assume that $\Lambda_i(\vec{x})$ is an equivalence to $\Omega_i(\Phi_i(\vec{x}))$.

If action $as_{k,j}$ is represented by decision vector \vec{x} , then $\Lambda_i(as_{k,j}) = \Lambda_i(\vec{x})$. Some comments: The issue of establishing of the functions from the sets Φ or Λ is an important and nontrivial problem. The simplest (but very often difficult or impossible to perform) way to establish the functions is to declare them manually. Although each function can have an individual character, sometimes we can use a default function. For example, if we know that only one decision variable of the decision space \vec{x} (e.g., d_1 , where by pd_1 we denote the value of the decision variable d_1) influences value v_i , we know the maximal and minimal possible value of d_1 (which we denote $max(d_1)$ and $min(d_1)$), and we know that the influence is linear, then we can compute $v_i(\vec{x})$ on the basis of the normalization function used in [16]: In order to transform the values of decision variables into the levels of value's promotion we can use the following normalization function:

$$v_i(\vec{x}) = \Phi_i(\vec{x}) = (pd_1 - min(d_1)) / (max(d_1) + e - min(d_1)) \text{ where}$$

- pd_1 - is the actual level of the decision variable d_1 .
- x - is the situation that promotes a certain value.
- $min(d_1)$ - is the minimal level of value pv_i .

- $\max(d_1)$ - is the maximal level of value pv_i .
- e - is an arbitrarily small positive quantity.

The result of $\Phi_1(\vec{x})$ is the level of the corresponding value $v_i(\vec{x})$.

For decision variables in which higher levels indicate a lower value's promotion we inverse the result of the normalization function: $1 - \Phi_1(pd_1)$.

Much more problems will appear if maximal or minimal values of a given decision variable are known partially (we can estimate $\max_e(d_1)$ or $\min_e(d_1)$ but we cannot exclude examples which are above max or below min.). In such situations we can introduce an imperfect (disrupting the linearity of function Φ_1) solution: If pd_1 is between $\max_e(d_1)$ and $\min_e(d_1)$, it is computed on the basis of the modified normalization function:

$$v_i(\vec{x}) = \Phi_1(\vec{x}) = 0.8(pd_1 - \min_e(d_1)) / (\max_e(d_1) + e - \min_e(d_1)) + 0.1$$

where e , 0.8 and 0.1 are coefficients.

If pd_1 is higher than the previous known maximum, the level of promotion of value v_i can be computed on the basis of equation:

$$v_i(\vec{x}) = v_i(\max(d_1)) + (1 - v_i(\max(d_1))) / 2 \text{ (where } v_i(\max(d_1)) \text{ is the level of promotion of value } v_i \text{ by an actual maximal value of } d_1).$$

If pd_1 is lower than the previous known minimum, the level of promotion of value v_i can be computed on the basis of equation:

$$v_i(\vec{x}) = v_i(\min(d_1)) - (v_i(\min(d_1))) / 2 \text{ (where } v_i(\min(d_1)) \text{ is the level of promotion of value } v_i \text{ by an actual minimal value of } d_1).$$

If a next new pd_1 has value between $\max_e(d_1)$ and new maximal value of v_i , then $v_i(\vec{x})$ should be between these values. Such a solution is imperfect, but in our opinion, it can be sufficient in many situations.

Alternatively, functions from the sets Φ or Λ can be obtained by machine learning mechanisms. Such a task requires a relatively large training set including various decision vectors with assigned to them proper estimation of each value. This problem is presented in a more detailed way in Section 9.2.

7. INFERENCE RULES

In this work, inference rules will be presented in the form of argumentation schemes, which are forms of argument representing stereotypical patterns of human reasoning.

Argumentation schemes have an antecedent part and a consequence part separated by a double bar which stands for the sign of defeasible inference. Argumentation schemes in computational models are usually interpreted as defeasible inference rules and they constitute a part of a whole argumentation framework, providing a basis on which the process of reasoning is conducted (a detailed description of the framework is presented in Section 10).

The author of [14] introduces a number of such argumentation schemes allowing for the realization of value-based and teleological reasoning. The issue of argumentation schemes was widely discussed in many papers and there is a number of different views on their nature. The author of Prakken [27] presents an in-depth analysis of various approaches to argumentation schemes. From the most general point of view he divided them to the two approaches:

the logical one, when argument is treated as a defensible inference rule and the dialogical approach in which arguments are dialogical devices. Since in our approach we do not introduce any dialog moves with shifting the burden of proof, we understand the argumentation schemes as a defeasible inference rule, but unlike it is discussed in the H. Prakken paper, our model is not purely logical because it contains some procedural elements (especially the function ε which changes the state of affairs).

Note that, unlike it is in standard approaches to argumentation schemes, we do not introduce critical questions (the ways of challenging of arguments). It was caused by purely technical character of our model (we do not model the human reasoning process, but we are going to create a reasoning framework for autonomous agents). The ways of challenging of arguments were implemented into reasoning framework.

Below are presented mechanisms from [14] and [16] which have been adapted to our reasoning model with a numeric representation of the level of value promotion (the names will correspond to the mechanisms from [14]). Since the AS1 scheme from [14] is very basic and is practically included in AS2, it will be omitted.

AS2 Generalized practical reasoning: If in circumstances s_m performing an action a_i is preferred to remaining in s_m and $as_{t,m} \in AS$, then an action a_i should be performed:

$$\frac{\exists_{s_m \in S} \exists_{as_{t,m} \in AS} : VO^{as_{t,m}}(as_{t,m}) \geq VO^{s_m}(s_m)}{\varepsilon(as_{t,m})}$$

In the above example, relation \gg from [14] (def. 9) is expressed by means of order $ORO (\geq)$ which takes into consideration the weight function.

AS3 Reasoning with abstract goals: If in the current circumstances s_m achieving an abstract goal ga_k is possible by an action a_i performed in s_m , then an action $as_{t,m}$ becomes the practical goal gp :

$$\frac{\exists_{ga_k \in GA} \exists_{s_m \in S} \exists_{as_{t,m} \in AS} : \gamma(s_m) = 1 \wedge sat(as_{t,m}, ga_k)}{gp = as_{t,m}}$$

AS4 Reasoning with unreachable abstract goal: In the current circumstances s_m , a state of affairs s_m does not satisfy the goal or promotes the key value of gua_m to a lesser extent than $as_{t,m}$, both $as_{t,m}$ and $as_{k,m}$ satisfy ordinary values of a goal gua , but $as_{t,m}$ satisfies the key value of gua_m to a greater extent than $as_{k,m}$, hence $as_{t,m}$ becomes the practical goal gp :

$$\frac{\begin{aligned} &\exists_{gua_m \in GUA} \exists_{s_m \in S} \exists_{as_{t,m} \in AS} : (\gamma(s_m) = 1 \wedge \\ &\exists_{v_z \in gua_m} : (v_z(s_m) < v_x \min(gua_m)) \vee (v_z(s_m) < v_z(as_{t,m})) \wedge \\ &\forall_{v_x \in gua_m} : (v_x(as_{t,m}) \geq v_x \min(gua_m)) \wedge \omega(gua_m) = v_y \wedge \\ &\neg \exists_{(as_{k,m}) \in AS} : ((v_y(as_{k,m}) > v_y(as_{t,m})) \wedge \\ &\forall_{v_x \in gua_m} : (v_x(as_{k,m}) \geq v_x \min(gua_m))) \end{aligned}}{gp = as_{t,m}}$$

The abstract unreachable goal can be understand as a specific kind of abstract goal in which one value should be

promoted to the maximal possibly extent. The idea is that the agent chooses the decision (action) which promotes this (key) value to the highest possible extent, while rest of the values are promoted at least to their satisfactory levels.

For the sake of clarity we explain the subsequent verses of the above scheme:

In the first verse, we assume that there exist an abstract unreachable goal, and the available action.

The second verse confirms the actual state of affairs.

Third and fourth verses examine if the actual state of affairs does not satisfy the goal (are the ordinary values promoted below the thresholds or exists available action which promote key value to the higher extent). If not, then

Fifth verse examine if exists available action ($as_{t,m}$) which promotes the ordinary values to the satisfactory levels. If yes, The last three verses examine if there exists other available actions which promote the key value to the higher extent than $as_{t,m}$ and promote the ordinary values at least to the declared thresholds. If not then $as_{t,m}$ may become a practical goal.

AS5 Goal-driven practical reasoning In the current circumstances s_m , in order to achieve the practical goal gp , an action a_t should be performed:

$$\frac{\exists_{s_m \in S} \exists_{as_{t,m} \in AS} : (\gamma(s_m) = 1) \wedge (gp = as_{t,m})}{\mathcal{E}(as_{t,m})}$$

Although we focused on the inference rules from [14], note that new inference rules can be added do the above list.

8. DECISION-MAKING AS AN OPTIMIZATION PROBLEM

One can notice that our model has some common elements with multi-attribute utility theory [28]. Obviously, the topic of MAUT was widely discussed and developed and it would be difficult to discuss here a relations between our approach and all versions and approaches to MAUT, but we will try to present a brief look on the similarities and differences between our model and the most basic approach to multi-attribute utility theory.

Similar to MAUT we evaluate possible decision options in the light of different values (in MAUT utilities) and then we compute the cumulative evaluation of a decision (in MAUT the aggregate utility function) which is the basis of the decision. The process of aggregation of utility in MAUT can be additive or multiplicative. In our work we use the Θ function, which can be seen as a variation of the additive rule (but normalized to the range $< 0 : 1$).

Unlike it is in MAUT, we do not use weights (as numbers), but we use weight functions which can be more complex and we do not implement the reasoning with uncertainty (but we plan to do it in the future works). The goal of MAUT is to find option with the highest aggregate utility function. In our work the goals of the model can be more complex, similar to MAUT we can maximize the cumulative evaluation of all values (such an approach can be achieved with the use of AS2), but we also introduce the thresholds (minimal acceptable levels of promotion of values) and the possibility of

maximization of one value having other above the threshold (like it is in unreachable abstract goals). Moreover, our system allows for utilization of other (to appear in future works) argumentation schemes.

Concluding, since there is a number of similarities (and some differences) between our model and MAUT, we believe that our approach can be seen as a bridge between a value-based decision-making and the utility theory.

The model presented in our paper was designed, above all, for situations involving a finite and discrete space of events wherein a finite AS set can be presented.

A considerably more complex situation is observed where at least one of the decision-making variables relates to an infinite set of values. If it is infinite but discrete, it is often possible to limit the range of variability for the given decision variable without prejudice to the quality of the decision. If the domain of at least one of the variables is continuous, the decision space will also be continuous or semi-continuous. In such a situation, the variable should be, if it is possible, discretized. If it is not possible, the argumentation-based approach fails: the number of possible decision-making options is infinite and no viable arguments for or against the respective options can be formulated. In this context, the most effective decision-making models are based not on argumentation, but on single- and multi-criterion optimization aimed at maximizing the target function. Although the mechanism presented herein was developed with a discrete event space in mind, its underlying premise can, to a certain extent, be also applied in the development of an optimization model. The key premise of our model is the assumption that decisions are rooted in values that should be promoted or a goal comprising a set of values that should be promoted up to a certain minimum level. Should we treat such values as criteria in a decision-making process, the entire sequence could be interpreted as single- or multi-criterion optimization.

In Section 6, we introduced a set of functions Φ which allowed us to evaluate the level to which particular values are promoted by different values of the vector \vec{x} (and therefore the respective actions from the set AS). The assessment can be processed by employing the weight function (Ω_i), which subsequently allows the application of the Θ function for the purposes of a joint function-based evaluation of the promotion of the selected set of values V^Z , which in turn will allow a comparison between the promotion levels for various values promoted by different values of decision variables, as described by the vector \vec{x} .

At the beginning of our paper, we distinguished several types of goals that an agent can achieve. Abstract goals and unreachable abstract goals correspond to the agent's needs with respect to the levels up to which particular values ought to be promoted. Based on the same, material goals (i.e., specific actions facilitating particular states of affairs) which fulfil abstract goals are defined, and based on these, the practical goal, i.e., the action to be performed by the agent, is determined. Presented below is the method of approaching the decision-making problem, based on the thus defined goals, as an optimization task.

Let us begin by assuming the following:

Let the model of a decision-making situation D comprise the pair:

$D = (F(\vec{x}), l(\vec{x}))$, where \vec{x} is the vector of decision variables (d_1, d_2, \dots, d_n) , $F(\vec{x})$ is the goal function which ought to be maximized, and $l(\vec{x})$ is the set of formulae limiting the values of the input vector. Explained below is the method for developing a model of the optimization situation in three distinct cases:

- 1 The first case is one where no preimposed goal exists. The decision is based on a choice of the option (option vector) which ensures the best total promotion of all values (taking into account the weight of the respective values).

In the analyzed case, we have a defined goal function $F(\vec{x})$ (maximizing the promotion of all values relative to their weight), but with no limitations: $l(\vec{x}) = \emptyset$.

Assuming that function Φ allows us to determine the level of promotion of values: $v_i(\vec{x}) = \Phi_i(\vec{x})$, function Ω determines the relative level of value promotion: $vo_i(\vec{x}) = \Omega_i(v_i(\vec{x}))$, and $VO(\vec{x})$ is the set of relative promotion levels for all values promoted by the vector \vec{x} , then the goal function will be as follows: $F(\vec{x}) = \max(\Theta(VO(\vec{x})))$.

- 2 In the second case, we have a declared an abstract goal $ga \in GA$, i.e., a set of minimum levels for values within a given set V^{ga} . This case is not an optimization task as it has no declared increase of the goal function. The abstract goal itself does not define the value or values that should be maximized. An abstract goal allows the definition of a set of limitations $l(\vec{x})$. The set will contain the number of limitations declared in the goal ga . Each limitation will pertain to a single value. For each value v_n whose minimum value is declared in the goal ga ($\forall_{v_n \in ga}$), the following must be true: $\Phi_n(\vec{x}) \geq v_n \min(ga)$. Because the abstract goal itself does not define a goal function as such, the function from case 1 can be used adopted as the goal function: $F(\vec{x}) = \max(\Theta(VO(\vec{x})))$.

- 3 In the third case, the agent has a declared unreachable abstract goal $gua \in GUA$. An unreachable abstract goal is a set of minimum levels up to which the respective values ought to be promoted with one (key value) requiring the maximum possible level of promotion.

In this case, the goal function will entail maximizing the level of promotion for the key value whereas the minimum levels of promotion for ordinary values of the goal gua will constitute the limits:

If $\omega(gua) = v_m$, then the goal function is to maximize the level of promotion for the value v_m : $F(\vec{x}) = \max(\Phi_m(\vec{x}))$.

The limit set will be defined on the basis of the ordinary values of the unreachable abstract goal:

For each value v_n whose minimum level is declared in the goal gua ($\forall_{v_n \in gua}$), the following must be true: $\Phi_n(\vec{x}) \geq v_n \min(gua)$.

Having obtained, after employing one of the aforementioned approaches, the set l and function $F(\vec{x})$, we can use the adequate

optimization mechanism to determine the vector \vec{x}_i (or vectors, where more than one applies) for which the function $F(\vec{x})$ reaches its maximum, thus rendering it (or one of the same) the best possible decision.

As the development of the optimization mechanism is outside the goal and scope of this paper, and because we do not introduce any specific functions Φ or Ω (which can differ significantly depending on the actual decision-making problem at hand), it is not possible to introduce here a single optimization mechanism that would allow us to identify the maxima of the goal function $F(\vec{x})$ in each of the three cases.

9. SOME COMMENTS ON DECISION-MAKING

In the introduction to this paper, we observed that the ethical context of decisions made by an autonomous agent becomes more important where the environment in which it is to function influences other people and becomes more unpredictable. This has been confirmed by the famous example of the Tay bot [29] which very quickly “learned” to become a racist. The example relates also to the problem of trust placed in learning mechanisms: how far can we trust a device whose actions in the real world can be potentially dangerous to humans? This is particularly true of mechanisms that are not capable of explaining their actions (such as neural networks): A device can “learn” bad behaviors by observing the behavior of other agents (similarly to humans who can be persuaded to do something they would not normally do through the influence of “bad company” or peer pressure).

The proposed framework allows us to include ethical concerns in the decision-making process. It also allows us to view ethics in a somewhat broader perspective, i.e., not only as based on deontic principles of mandatory action and prohibition, but also focused on the promotion of values and their adequate, relative proportions. Indeed, some authors [15,30] go as far as to directly suggest that under certain circumstances deontic rules should be broken and indicate values as a viable justification for the decision to break a rule.

The in-depth discussion of the problem of ethical behavior of autonomous systems was presented in [bench-capon2020]. The author discussed the problem of autonomous system in the light of three main ethical theories: consequentialism, deontological ethics, and virtue ethics. Although such an analysis is outside the scope of this paper, in our opinion (similar to the mentioned in [bench-capon2020] AATS+V model), our model can be treated as an attempt to model the virtue ethics, especially with the use of values in abstract goals and unreachable abstract goals which can represent virtues (however, we have also implemented some elements of consequentialism, e.g., the analysis of the actions in the light of the future consequences).

Moreover, although the author of [bench-capon2020] points out that his approach is focused on modeling of ethical behavior rather than ethical reasoning, we believe that autonomous setting the practical goal (action and state of affairs to be achieved) from abstract

or abstract unreachable goal can be seen as a first step to model the ethical reasoning.

9.1. Two Sources of Knowledge

People acquire knowledge necessary in decision-making processes in various ways. Overall, in the most generalized perspective, these can be divided into two groups: knowledge acquired directly from sources: from books, teachers, parents, etc.; and knowledge acquired through experience: a person acquires knowledge through induction, by interacting with the world and evaluating the effects and consequences of certain behaviors. For instance, the first method can allow us to learn road traffic rules, what is and what is not allowed, whether it is acceptable to drive 100 km/h in an urban area, etc. Meanwhile, the second method allows us to learn how to determine the distance from the car in front of us and decide whether it is safe, will the speed with which we enter a corner allow us to negotiate it safely, etc.

The development of formal models of knowledge representation, inference, and argumentation allows us to represent knowledge acquired using the first of the above methods. On the other hand, the spectacular advances made in machine learning in recent years allowed the development of very complex systems focused on the second method of knowledge acquisition.

Based on the above, it is our belief that autonomous systems acting in the real world should take advantage of both of the above approaches. The model presented herein is based on formal reasoning, i.e., it allows the representation of knowledge acquired in the first manner; however, it has been structured in such a way so as to facilitate the implementation of a hybrid system that would draw upon both approaches to the use of artificial intelligence in decision-making.

9.2. Hybrid Model of Decision-Making

In this section, we will propose a way to combine the described model with machine learning systems. The idea of integration of the two approaches entails the division of functions between the respective types of knowledge:

- 1 On the one hand we have knowledge in the form of straightforward and clear principles of reasoning and rules for their application (developing arguments, solving arguments, etc.). This type of knowledge can be input into the system directly as it allows a clear and unambiguous choice of the best option based on the values of the agent's goals (what is more, the model is also capable of explaining why this is the case).
- 2 On the other hand we have knowledge which allows us to recognize whether the conditions for applying the inference rules have been satisfied, to what extent the respective values are indeed promoted, etc. This type of knowledge is less about inference and more about pattern recognition, classification, etc.

On the basis of the above division, it can be assumed that functions from the set Φ need not be directly provided. They can take the

form of any mechanisms assigning to a decision vector a number from the range $(0; 1)$, and can therefore be a machine learning based mechanism, even one based on the so-called “black box” principle. The ML-based function from the set Φ can be obtained on the basis of the so-called supervised machine learning (linear or nonlinear regression or, for the cases with a small number of discrete options, classifier). In order to build the function, the training set of various decision vectors (\vec{x}) with assigned to them proper estimation of the level of promotion of a given value should be prepared. On the basis of the training set the regression function (or classifier) can be trained (however, it is important that the estimated value should be a number from the range $(0; 1)$). Naturally, the development of such a training set and a mechanism is anything but easy, but this far exceeds the scope of the discussion at hand. In our opinion, this can be an interesting direction of future research. Similarly to the functions from the set Φ , the product of the machine learning process can be the activation of the weight function from the set Ω (as a filter processing the level of value promotion, i.e., a number in the range of $(0; 1)$, to the weighted level of promotion, i.e., a number in the same range), or function δ returning the result of a given action being performed under a specific state of affairs (function δ can take the form of a classifier assigning a certain decision-making vector \vec{x} to one of the available states of affairs $s_k \in S$). It is noteworthy that under real-world circumstances, the determination of separate functions Φ_i and Ω_i can prove very difficult, indeed unnecessary, and such situations are best handled by considering them jointly and determining, through the application of machine learning mechanisms, a single function Λ_i .

Another, alternative way of obtaining Φ_i and Ω_i or Λ_i is learning from agent's own experience through some form of reinforcement learning functions. However, this a perspective topic to be discussed in a future work.

Concluding, the framework presented in our paper allows the integration of machine learning based solutions with traditional solutions involving logical inference modeling.

10. ARGUMENTATION FRAMEWORK

In this section we introduce a reasoning mechanism allowing for decision-making. The mechanism allows for establishing of the agent's goal and performing actions allowing for fulfilment of the goal (practical reasoning). The argumentation-based practical reasoning framework introduced in our work is based on the simple reasoning mechanism presented in [14].

- 1 We assume that sets S , A , \vec{X} , and V are elements of knowledge base KB .
- 2 By $KB \vdash E$ we denote that E is derived from knowledge base KB . E can be inferred from the knowledge base where E will be an element of any of the subsets of KB , or E will be an outcome of the operation of one of the functions $(\gamma, \delta, \varepsilon, \Phi, \Omega, \Theta)$ or standard mathematical (set-theory, logical, etc.) operators on the elements of any of the subsets of KB .
- 3 Let IR be a set of inference rules. As inference rules we understand the argumentation schemes described in Section 7.

- 4 We assume that arguments are constructed on the basis of inference rules from set IR and knowledge base KB . An argument will be constructed when $KB \vdash form$ and $form$ satisfies the conditional part of an inference rule. By $Conc(arg_x) = E$ we denote that E is the conclusion of an argument arg_x .
- 5 ARG is the smallest set of all finite arguments constructed from KB and IR .
- 6 Two arguments are in conflict if one attacks the other.
- 7 There is only one kind of attack: rebuttal, which is an attack on the conclusion of an argument. The attack has a symmetrical character: if $arg1$ attacks $arg2$, then $arg2$ also attacks $arg1$.
- 8 An attack on the conclusion of an argument $arg1$ occurs if:
 - $\exists_{arg1, arg2 \in ARG} : Conc(arg1) = \epsilon(as_{t,m}) \wedge Conc(arg2) = \epsilon(as_{t,m}) \wedge a_t \neq a_z$
($arg1$ concludes that $\epsilon(as_{t,m})$ and there exists argument $arg2$ which concludes $\epsilon(as_{z,m})$, where $a_t \neq a_z$), or
 - $\exists_{arg1, arg2 \in ARG} : Conc(arg1) = gp = as_{t,m} \wedge Conc(arg2) = gp = as_{z,m} \wedge a_t \neq a_z$
($arg1$ concludes that $gp = as_{t,m}$ and there exists argument $arg2$ which concludes that $gp = as_{z,m}$, where $a_t \neq a_z$).
- 9 We assume a partial ordering between arguments $OARG = (ARG, >)$ where if $arg1 > arg2$, then it means that $arg1$ is stronger than $arg2$.
- 10 We assume that the basic grounds for determining an order ($OARG$) between arguments is the inference rule on the basis of which the argument is constructed. We assume that $AS5 > AS2$, meaning that if arguments $arg1$ and $arg2$ are in conflict and if $arg1$ is built on the basis of $AS5$ and $arg2$ is built on the basis of $AS2$, then $arg1 > arg2$.
- 11 In order to eliminate conflicts between arguments constructed on the basis of $AS3$ and $AS4$ (root of this conflict is in the point that unreachable abstract goal is also abstract goal) we assume that if there is declared unreachable abstract goal, then the system does not generate arguments on the basis of $AS3$.
- 12 Argument $arg1$ defeats argument $arg2$ when argument $arg1$ rebuts argument $arg2$ and $arg2 \not> arg1$.
- 13 Reasoning about priorities: we assume that priorities between conflicting arguments built on the basis of the same inference rule depend on values whose application the argument promotes.
 - If both arguments ($arg1$ and $arg2$) are built on the basis of inference rule $AS2$, argument $arg1$ attacks argument $arg2$ (or vice versa), the conclusion of $arg1$ is $\epsilon(x_1)$, the conclusion of $arg2$ is $\epsilon(x_1)$, and $VO^{x_1}(x_1) \blacktriangleright VO^{x_2}(x_2)$, then $arg1 > arg2$.
 - If both arguments ($arg1$ and $arg2$) are built on the basis of inference rule $AS3$, argument $arg1$ attacks argument $arg2$ (or vice versa), the conclusion of $arg1$ is $gp = gm_i$, where $gm_i = x_1$, the conclusion of $arg2$ is $\epsilon(x_1)gp = gm_k$, where $gm_k = x_2$, and $VO^{x_1}(x_1) \blacktriangleright VO^{x_2}(x_2)$, then $arg1 > arg2$.
- 14 If two arguments ($arg1, arg2$) are built on the basis of the same inference rule ($ARG2, ARG3$, or $ARG4$), $arg1 \not> arg2$ and $arg2 \not> arg1$, then both conclusions equally fulfil the agent's goal and it is impossible to solve the conflict on grounds of value preferences. Such a situation will appear if it is impossible to derive preference between situations (e.g., if multiple situations promote cumulative levels of values' promotion to the same extent, where individual values are promoted to different levels). Such a conflict can be solved with some other (not listed in Section 7) inference rule, remain unsolved, or it can be solved randomly.
- 15 Argument $arg1$ is not defeated if it is not attacked by any argument or all arguments which attack $arg1$ are defeated.
- 16 If an argument whose conclusion sets a practical goal (i.e., $gp = as_{t,m}$) is not defeated, then its conclusion will be added to the knowledge base.
- 17 If an argument (i.e., $arg1$) is defeated, then its conclusion will be deleted from KB and action connected with its conclusion will not be performed. If the conclusion of a defeated argument is a premise of the other argument (i.e., $arg2$) and $\neg \exists_{arg_x \in ARG} : Concl(arg_x) = Concl(arg1)$, then $arg2$ will also be defeated.
- 18 If one of the arguments concludes that $\epsilon(as_{t,m})$ and the argument is not defeated, it brings about performing an action $as_{t,m}$.

Generally speaking, the argumentation framework used in the example was inspired by the ASPIC+ argumentation framework [31].

11. EXAMPLE

This section illustrates the process of deciding which action should be performed by the agent. We assume that there exists a smart device Dev_1 in the form of "smart glasses" equipped by a person in an industrial environment. The device Dev_1 is fully autonomous and can perform a variety of actions without the active participation of the user that wears it. One of the functionalities of such a device is scanning for objects of interest and when found taking pictures of the field of view of the person. Dev_1 should back up the pictures on an external storage server Ser_1 when a stable connection is found. We also assume that Ser_1 is accessible in the public network (Internet) and Dev_1 is capable of connecting to that network as well (i.e., via WiFi or 4G infrastructure). Such a scenario involves a lot of parameters that affect the image upload speed, the reliability of the connection, the energy supply of the device, carrier fees, the security and quality of the resource being sent. For a better presentation of our mechanism, we assume that the Dev_1 device focuses only on the security, upload speed and quality of the resource. One of the main strengths of our method that we would like to point out in this example is its compatibility with the self-governing capabilities of the agent (Dev_1). This is partly achieved via the use of a numerical representation for most of the parameters that the evaluation process relies on. Most of the similar methods don't focus on the autonomy of the agent. Dev_1 sets the practical goals autonomously basing on the current environment state. After the evaluation process it decides on whether to perform an action to achieve a given goal or not.

Three values are important for the agent:

- $v_{security}$
- $v_{quality}$
- v_{speed}

The agent declares an abstract unreachable goal gua in which it declares the minimal levels of promotion of three values ($v_{security} \min(gua)$, $v_{quality} \min(gua \in gua)$, $v_{speed} \min(gua \in gua)$) in which:

$$v_{security} \min(gua) = 0.85$$

$$v_{quality} \min(gua) = 0.5$$

$$v_{speed} \min(gua) = 0.45$$

The key value of the goal gua is *security* ($\omega(gua) = v_{security}$), which means that the level of promotion of security should be as high as possible.

The process of deciding which action should be performed is carried out by setting a practical goal by the device Dev_1 , which involves 5 stages. Each stage is described in detail below:

11.1. STAGE 1: Decision Space

This stage involves the definition of the decision space. First of all, we have to define a set of the decision vectors which we will denote as \vec{x} .

In our case, Dev_1 has not yet sent the image for back up, therefore it is in the following initial state of affairs: s_{ins} .

Now we declare the decision space:

We assume that we have four decision variables (parameters):

- encryption algorithm,
- encryption key length,
- quality level,
- network speed

Next, we assume the domains of particular decision parameters:

- $x_1 = D_{security-algorithm} = \{none, Twofish, AES\}$
- $x_2 = D_{security-key} = \{0, 128, 192, 256\}$
- $x_3 = D_{compression-level} = \{low, medium, high\}$
- $x_4 = D_{network-speed} = \{slow, average, fast\}$

These domains specify all the possible physical values that a parameter can hold.

The next step is to define the vectors for all combinations of possible actions.

These vectors are defined as follows:

$$as_{(none,0,low,slow)}^{\rightarrow} = (none, 0, low, slow)^T$$

$$as_{(none,0,low,average)}^{\rightarrow} = (none, 0, low, average)^T$$

$$as_{(none,0,low,fast)}^{\rightarrow} = (none, 0, low, fast)^T$$

$$as_{(none,0,medium,slow)}^{\rightarrow} = (none, 0, medium, slow)^T$$

$$as_{(none,0,medium,average)}^{\rightarrow} = (none, 0, medium, average)^T$$

$$as_{(none,0,medium,fast)}^{\rightarrow} = (none, 0, medium, fast)^T$$

$$as_{(none,0,high,slow)}^{\rightarrow} = (none, 0, high, slow)^T$$

$$as_{(none,0,high,average)}^{\rightarrow} = (none, 0, high, average)^T$$

$$as_{(none,0,high,fast)}^{\rightarrow} = (none, 0, high, fast)^T$$

$$as_{(Twofish,128,low,slow)}^{\rightarrow} = (Twofish, 128, low, slow)^T$$

$$as_{(Twofish,128,low,average)}^{\rightarrow} = (Twofish, 128, low, average)^T$$

$$as_{(Twofish,128,low,fast)}^{\rightarrow} = (Twofish, 128, low, fast)^T$$

$$as_{(Twofish,128,medium,slow)}^{\rightarrow}$$

$$= (Twofish, 128, medium, slow)^T$$

$$as_{(Twofish,128,medium,average)}^{\rightarrow}$$

$$= (Twofish, 128, medium, average)^T$$

$$as_{(Twofish,128,medium,fast)}^{\rightarrow}$$

$$= (Twofish, 128, medium, fast)^T$$

$$as_{(Twofish,128,high,slow)}^{\rightarrow} = (Twofish, 128, high, slow)^T$$

$$as_{(Twofish,128,high,average)}^{\rightarrow}$$

$$= (Twofish, 128, high, average)^T$$

$$as_{(Twofish,128,high,fast)}^{\rightarrow} = (Twofish, 128, high, fast)^T$$

$$as_{(Twofish,196,low,slow)}^{\rightarrow} = (Twofish, 196, low, slow)^T$$

$$as_{(Twofish,196,low,average)}^{\rightarrow}$$

$$= (Twofish, 196, low, average)^T$$

$$as_{(Twofish,196,low,fast)}^{\rightarrow} = (Twofish, 196, low, fast)^T$$

$$as_{(Twofish,196,medium,slow)}^{\rightarrow} = (Twofish, 196, medium, slow)^T$$

$$as_{(Twofish,196,medium,average)}^{\rightarrow}$$

$$= (Twofish, 196, medium, average)^T$$

$$as_{(Twofish,196,medium,fast)}^{\rightarrow} = (Twofish, 196, medium, fast)^T$$

$$as_{(Twofish,196,high,slow)}^{\rightarrow} = (Twofish, 196, high, slow)^T$$

$$as_{(Twofish,196,high,average)}^{\rightarrow}$$

$$= (Twofish, 196, high, average)^T$$

$$\begin{aligned}
\vec{as}_{(Twofish,196,high,fast)} &= (Twofish, 196, high, fast)^T \\
\vec{as}_{(Twofish,256,low,slow)} &= (Twofish, 256, low, slow)^T \\
\vec{as}_{(Twofish,256,low,average)} &= (Twofish, 256, low, average)^T \\
\vec{as}_{(Twofish,256,low,fast)} &= (Twofish, 256, low, fast)^T \\
\vec{as}_{(Twofish,256,medium,slow)} &= (Twofish, 256, medium, slow)^T \\
\vec{as}_{(Twofish,256,medium,average)} &= (Twofish, 256, medium, average)^T \\
\vec{as}_{(Twofish,256,medium,fast)} &= (Twofish, 256, medium, fast)^T \\
\vec{as}_{(Twofish,256,high,slow)} &= (Twofish, 256, high, slow)^T \\
\vec{as}_{(Twofish,256,high,average)} &= (Twofish, 256, high, average)^T \\
\vec{as}_{(Twofish,256,high,fast)} &= (Twofish, 256, high, fast)^T \\
\vec{as}_{(AES,128,low,slow)} &= (AES, 128, low, slow)^T \\
\vec{as}_{(AES,128,low,average)} &= (AES, 128, low, average)^T \\
\vec{as}_{(AES,128,low,fast)} &= (AES, 128, low, fast)^T \\
\vec{as}_{(AES,128,medium,slow)} &= (AES, 128, medium, slow)^T \\
\vec{as}_{(AES,128,medium,average)} &= (AES, 128, medium, average)^T \\
\vec{as}_{(AES,128,medium,fast)} &= (AES, 128, medium, fast)^T \\
\vec{as}_{(AES,128,high,slow)} &= (AES, 128, high, slow)^T \\
\vec{as}_{(AES,128,high,average)} &= (AES, 128, high, average)^T \\
\vec{as}_{(AES,128,high,fast)} &= (AES, 128, high, fast)^T \\
\vec{as}_{(AES,196,low,slow)} &= (AES, 196, low, slow)^T \\
\vec{as}_{(AES,196,low,average)} &= (AES, 196, low, average)^T \\
\vec{as}_{(AES,196,low,fast)} &= (AES, 196, low, fast)^T \\
\vec{as}_{(AES,196,medium,slow)} &= (AES, 196, medium, slow)^T \\
\vec{as}_{(AES,196,medium,average)} &= (AES, 196, medium, average)^T \\
\vec{as}_{(AES,196,medium,fast)} &= (AES, 196, medium, fast)^T \\
\vec{as}_{(AES,196,high,slow)} &= (AES, 196, high, slow)^T \\
\vec{as}_{(AES,196,high,average)} &= (AES, 196, high, average)^T \\
\vec{as}_{(AES,196,high,fast)} &= (AES, 196, high, fast)^T
\end{aligned}$$

$$\begin{aligned}
\vec{as}_{(AES,256,low,slow)} &= (AES, 256, low, slow)^T \\
\vec{as}_{(AES,256,low,average)} &= (AES, 256, low, average)^T \\
\vec{as}_{(AES,256,low,fast)} &= (AES, 256, low, fast)^T \\
\vec{as}_{(AES,256,medium,slow)} &= (AES, 256, medium, slow)^T \\
\vec{as}_{(AES,256,medium,average)} &= (AES, 256, medium, average)^T \\
\vec{as}_{(AES,256,medium,fast)} &= (AES, 256, medium, fast)^T \\
\vec{as}_{(AES,256,high,slow)} &= (AES, 256, high, slow)^T \\
\vec{as}_{(AES,256,high,average)} &= (AES, 256, high, average)^T \\
\vec{as}_{(AES,256,high,fast)} &= (AES, 256, high, fast)^T
\end{aligned}$$

11.2. STAGE 2: Calculation of Values (Φ)

During the second stage of our decision-making process, we have to calculate the level of promotion of values for all possible actions that can be performed initial state of the device.

The set Φ has three functions $\Phi = \{\Phi_{security}, \Phi_{quality}, \Phi_{speed}\}$. All three functions can have two kinds of arguments: the initial state of affairs (s_{ins}) and all the elements of the set AS_{ins} .

If by $\vec{x} = (x_1, x_2, x_3, x_4)$ we denote our decision vector, a and b will be coefficients, then function $\Phi_{security}$ will be as follows:

Definition 19. [Definition of $\Phi_{security}$] For the special case in which Dev_1 stays in the initial state of affairs ($\Phi_{security}(s_{ins})$), the level of security equals $\Phi_{security}(s_{ins}) = 0$ (since the image has not been backed up, the security of the data equals zero). For the other cases, if

- $x_1 = none$, then $a = 0$;
- $x_1 = Twofish$, then $a = 0.95$;
- $x_1 = AES$, then $a = 1$.

Then

$$\Phi_{security}(as_{\vec{x}}) = x_2 * a / (\max(x_2) + 1 / \max(x_2) - \min(x_2)).$$

For example,

$$\begin{aligned}
v_{security}(as_{(Twofish,128,low,average),ins}) \\
&= \Phi_{security}(as_{(Twofish,128,low),ins}) \\
&= (128 * a) / (256 + 0.003 + 0) \\
&= (128 * 0.95) / 256.003 = 0.406
\end{aligned}$$

The levels of promotion of the value $v_{security}$:

$$\begin{aligned}
v_{security}(s_{ins}) &= \Phi_{security}(s_{ins}) = 0 \\
v_{security}(as_{(none,0,low,slow),ins}) &= 0.0 \\
v_{security}(as_{(none,0,low,average),ins}) &= 0.0
\end{aligned}$$

$$\begin{aligned}
v_{security}(as_{(none,0,low,fast),ins}) &= 0.0 \\
v_{security}(as_{(none,0,medium,slow),ins}) &= 0.0 \\
v_{security}(as_{(none,0,medium,average),ins}) &= 0.0 \\
v_{security}(as_{(none,0,medium,fast),ins}) &= 0.0 \\
v_{security}(as_{(none,0,high,slow),ins}) &= 0.0 \\
v_{security}(as_{(none,0,high,average),ins}) &= 0.0 \\
v_{security}(as_{(none,0,high,fast),ins}) &= 0.0 \\
v_{security}(as_{(Twofish,128,low,slow),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,low,average),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,low,fast),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,medium,slow),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,medium,average),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,medium,fast),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,high,slow),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,high,average),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,high,fast),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,low,slow),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,low,average),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,low,fast),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,medium,slow),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,medium,average),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,medium,fast),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,high,slow),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,high,average),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,128,high,fast),ins}) &= 0.474 \\
v_{security}(as_{(Twofish,256,low,slow),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,low,average),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,low,fast),ins}) &= 0.949 \\
a \ v_{security}(as_{(Twofish,256,medium,slow),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,medium,average),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,medium,fast),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,high,slow),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,high,average),ins}) &= 0.949 \\
v_{security}(as_{(Twofish,256,high,fast),ins}) &= 0.949 \\
s \ v_{security}(as_{(AES,128,low,slow),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,low,average),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,low,fast),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,medium,slow),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,medium,average),ins}) &= 0.499
\end{aligned}$$

$$\begin{aligned}
v_{security}(as_{(AES,128,medium,fast),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,high,slow),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,high,average),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,high,fast),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,low,slow),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,low,average),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,low,fast),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,medium,slow),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,medium,average),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,medium,fast),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,high,slow),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,high,average),ins}) &= 0.499 \\
v_{security}(as_{(AES,128,high,fast),ins}) &= 0.499 \\
v_{security}(as_{(AES,256,low,slow),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,low,average),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,low,fast),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,medium,slow),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,medium,average),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,medium,fast),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,high,slow),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,high,average),ins}) &= 0.999 \\
v_{security}(as_{(AES,256,high,fast),ins}) &= 0.999
\end{aligned}$$

Definition 20. [$\Phi_{quality}$] Let us assume that:

- For the initial state of affairs $\Phi_{quality}(s_{ins}) = 0.8$ which means that the quality of the not sent picture is high (due to low compression). In the other cases:
 $\forall_{as_{(x_1,x_2,x_3,x_4),ins} \in AS_{ins}} \text{ s.t.:}$
 - $(x_3 = low) \rightarrow \Phi_{quality} = 0.8$
 - $(x_3 = medium) \rightarrow \Phi_{quality} = 0.6$
 - $(x_3 = high) \rightarrow \Phi_{quality} = 0.2$

Note that quality is inversely proportional (lower compression = higher quality).

On the basis of the function $\Phi_{quality}$ we compute the levels of promotion of $v_{quality}$:

$$\begin{aligned}
v_{quality}(s_{ins}) &= 0.8 \\
v_{quality}(as_{(none,0,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(none,0,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(none,0,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(none,0,medium,slow),ins}) &= 0.6
\end{aligned}$$

$$\begin{aligned}
v_{quality}(as_{(none,0,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(none,0,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(none,0,high,slow),ins}) &= 0.2 \\
v_{quality}(as_{(none,0,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(none,0,high,fast),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,128,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,128,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,128,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,128,medium,slow),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,128,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,128,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,128,high,slow),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,128,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,128,high,fast),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,128,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,128,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,128,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,128,medium,slow),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,128,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,128,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,128,high,slow),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,128,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,128,high,fast),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,256,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,256,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,256,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(Twofish,256,medium,slow),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,256,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,256,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(Twofish,256,high,slow),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,256,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(Twofish,256,high,fast),ins}) &= 0.2 \\
v_{quality}(as_{(AES,128,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(AES,128,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(AES,128,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(AES,128,medium,slow),ins}) &= 0.6 \\
v_{quality}(as_{(AES,128,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(AES,128,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(AES,128,high,slow),ins}) &= 0.2
\end{aligned}$$

$$\begin{aligned}
v_{quality}(as_{(AES,128,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(AES,128,high,fast),ins}) &= 0.2 \\
v_{quality}(as_{(AES,128,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(AES,128,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(AES,128,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(AES,128,medium,slow),ins}) &= 0.6 \\
v_{quality}(as_{(AES,128,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(AES,128,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(AES,128,high,slow),ins}) &= 0.2 \\
v_{quality}(as_{(AES,128,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(AES,128,high,fast),ins}) &= 0.2 \\
v_{quality}(as_{(AES,256,low,slow),ins}) &= 0.8 \\
v_{quality}(as_{(AES,256,low,average),ins}) &= 0.8 \\
v_{quality}(as_{(AES,256,low,fast),ins}) &= 0.8 \\
v_{quality}(as_{(AES,256,medium,slow),ins}) &= 0.6 \\
v_{quality}(as_{(AES,256,medium,average),ins}) &= 0.6 \\
v_{quality}(as_{(AES,256,medium,fast),ins}) &= 0.6 \\
v_{quality}(as_{(AES,256,high,slow),ins}) &= 0.2 \\
v_{quality}(as_{(AES,256,high,average),ins}) &= 0.2 \\
v_{quality}(as_{(AES,256,high,fast),ins}) &= 0.2
\end{aligned}$$

Definition 21. $[\Phi_{speed}]$ Let us assume that:

- For the initial state of affairs $\Phi_{speed}(s_{ins}) = 0$ which means that the network speed of the not yet sent picture is at the slowest (none). In the other cases:

$\forall_{as_{(x_1,x_2,x_3,x_4),ins} \in AS_{ins}}$ s.t.: If:

- $x_1 = none$, then $a = 1$;
- $x_1 = Twofish$, then $a = 0.95$;
- $x_1 = AES$, then $a = 0.9$;

and if:

- $x_3 = low$, then $b = 0.4$;
- $x_3 = medium$, then $b = 0.75$;
- $x_3 = high$, then $b = 1$;

and if:

- $x_4 = slow$, then $c = 0.2$;
- $x_4 = average$, then $c = 0.7$;
- $x_4 = fast$, then $c = 0.99$;

then:

$$\Phi_{speed}(as_{\rightarrow x}) = a * b * c.$$

On the basis of the function Φ_{speed} we compute the levels of promotion of v_{speed} :

$$v_{speed}(s_{ins}) = 0$$

$$v_{speed}(as_{(none,0,low,slow),ins}) = 0.08$$

$$v_{speed}(as_{(none,0,low,average),ins}) = 0.279$$

$$v_{speed}(as_{(none,0,low,fast),ins}) = 0.396$$

$$v_{speed}(as_{(none,0,medium,slow),ins}) = 0.15$$

$$v_{speed}(as_{(none,0,medium,average),ins}) = 0.524$$

$$v_{speed}(as_{(none,0,medium,fast),ins}) = 0.742$$

$$v_{speed}(as_{(none,0,high,slow),ins}) = 0.2$$

$$v_{speed}(as_{(none,0,high,average),ins}) = 0.7$$

$$v_{speed}(as_{(none,0,high,fast),ins}) = 0.99$$

$$v_{speed}(as_{(Twofish,128,low,slow),ins}) = 0.076$$

$$v_{speed}(as_{(Twofish,128,low,average),ins}) = 0.265$$

$$v_{speed}(as_{(Twofish,128,low,fast),ins}) = 0.376$$

$$v_{speed}(as_{(Twofish,128,medium,slow),ins}) = 0.142$$

$$v_{speed}(as_{(Twofish,128,medium,average),ins}) = 0.498$$

$$v_{speed}(as_{(Twofish,128,medium,fast),ins}) = 0.705$$

$$v_{speed}(as_{(Twofish,128,high,slow),ins}) = 0.19$$

$$v_{speed}(as_{(Twofish,128,high,average),ins}) = 0.664$$

$$v_{speed}(as_{(Twofish,128,high,fast),ins}) = 0.94$$

$$v_{speed}(as_{(Twofish,128,low,slow),ins}) = 0.076$$

$$v_{speed}(as_{(Twofish,128,low,average),ins}) = 0.265$$

$$v_{speed}(as_{(Twofish,128,low,fast),ins}) = 0.376$$

$$v_{speed}(as_{(Twofish,128,medium,slow),ins}) = 0.142$$

$$v_{speed}(as_{(Twofish,128,medium,average),ins}) = 0.498$$

$$v_{speed}(as_{(Twofish,128,medium,fast),ins}) = 0.705$$

$$v_{speed}(as_{(Twofish,128,high,slow),ins}) = 0.19$$

$$v_{speed}(as_{(Twofish,128,high,average),ins}) = 0.664$$

$$v_{speed}(as_{(Twofish,128,high,fast),ins}) = 0.94$$

$$v_{speed}(as_{(Twofish,256,low,slow),ins}) = 0.076$$

$$v_{speed}(as_{(Twofish,256,low,average),ins}) = 0.265$$

$$v_{speed}(as_{(Twofish,256,low,fast),ins}) = 0.376$$

$$v_{speed}(as_{(Twofish,256,medium,slow),ins}) = 0.142$$

$$v_{speed}(as_{(Twofish,256,medium,average),ins}) = 0.498$$

$$v_{speed}(as_{(Twofish,256,medium,fast),ins}) = 0.705$$

$$v_{speed}(as_{(Twofish,256,high,slow),ins}) = 0.19$$

$$v_{speed}(as_{(Twofish,256,high,average),ins}) = 0.664$$

$$v_{speed}(as_{(Twofish,256,high,fast),ins}) = 0.94$$

$$v_{speed}(as_{(AES,128,low,slow),ins}) = 0.072$$

$$v_{speed}(as_{(AES,128,low,average),ins}) = 0.252$$

$$v_{speed}(as_{(AES,128,low,fast),ins}) = 0.356$$

$$v_{speed}(as_{(AES,128,medium,slow),ins}) = 0.135$$

$$v_{speed}(as_{(AES,128,medium,average),ins}) = 0.472$$

$$v_{speed}(as_{(AES,128,medium,fast),ins}) = 0.668$$

$$v_{speed}(as_{(AES,128,high,slow),ins}) = 0.18$$

$$v_{speed}(as_{(AES,128,high,average),ins}) = 0.63$$

$$v_{speed}(as_{(AES,128,high,fast),ins}) = 0.891$$

$$v_{speed}(as_{(AES,128,low,slow),ins}) = 0.072$$

$$v_{speed}(as_{(AES,128,low,average),ins}) = 0.252$$

$$v_{speed}(as_{(AES,128,low,fast),ins}) = 0.356$$

$$v_{speed}(as_{(AES,128,medium,slow),ins}) = 0.135$$

$$v_{speed}(as_{(AES,128,medium,average),ins}) = 0.472$$

$$v_{speed}(as_{(AES,128,medium,fast),ins}) = 0.668$$

$$v_{speed}(as_{(AES,128,high,slow),ins}) = 0.18$$

$$v_{speed}(as_{(AES,128,high,average),ins}) = 0.63$$

$$v_{speed}(as_{(AES,128,high,fast),ins}) = 0.891$$

$$v_{speed}(as_{(AES,256,low,slow),ins}) = 0.072$$

$$v_{speed}(as_{(AES,256,low,average),ins}) = 0.252$$

$$v_{speed}(as_{(AES,256,low,fast),ins}) = 0.356$$

$$v_{speed}(as_{(AES,256,medium,slow),ins}) = 0.135$$

$$v_{speed}(as_{(AES,256,medium,average),ins}) = 0.472$$

$$v_{speed}(as_{(AES,256,medium,fast),ins}) = 0.668$$

$$v_{speed}(as_{(AES,256,high,slow),ins}) = 0.18$$

$$v_{speed}(as_{(AES,256,high,average),ins}) = 0.63$$

$$v_{speed}(as_{(AES,256,high,fast),ins}) = 0.891$$

Since all values have different importance, we assume three functions of the weight: $\Omega_{security}$, $\Omega_{quality}$ and Ω_{speed} . In our case, we assume that all of them will be linear ones:

$$vo_{security}(\vec{x}) = \Omega(v_{security}(\vec{x}) = 0.5 * (v_{security}(\vec{x})));$$

$$vo_{quality}(\vec{x}) = \Omega(v_{quality}(\vec{x}) = 0.3 * (v_{quality}(\vec{x})));$$

$$vo_{speed}(\vec{x}) = \Omega(v_{speed}(\vec{x}) = 0.2 * (v_{speed}(\vec{x}))).$$

11.3. STAGE 3: Construction of Arguments

In the previous stages we declared an abstract unreachable goal *gua*. Since the abstract unreachable goal is also an ordinary abstract goal, in order to determine the practical goal, the system can use two argumentation schemes: AS3 and AS4. However, due to the fact that AS4 based arguments win over AS3 (because of OARG order), in this case the argument construction process considers only AS4 type arguments.

Since we assumed that:

$$v_{security}min(gua) = 0.85,$$

$$v_{quality}min(gua) = 0.5, \text{ and}$$

$$v_{speed}min(gua) = 0.45$$

we can notice that the initial state of affairs s_{ins} does not satisfy the goal ($v_{security}(s_{ins}) < v_{security}min(gua)$), but since gua is an unreachable abstract goal, two situations satisfy the conditions of the ARG4 inference rule: $as_{ins,(AES,256,medium,average)}^T, as_{ins,(AES,256,medium,fast)}^T$.

This is because both situations satisfy the requirements ($as_{ins,(AES,256,medium,average)}^T$,

$as_{ins,(AES,256,medium,fast)}^T$ promote $v_{quality}$, v_{speed} and $v_{security}$ to a greater extent than the minimal requirements, and promote the key value ($v_{security}$) to a greater level than the other situations ($v_{security}(as_{ins,(AES,256,medium,average)}^T) = v_{security}(as_{ins,(AES,256,medium,fast)}^T) = 0.999$). On the basis of that, we can construct two arguments:

- $arg1$ with conclusion:
 $gp = as_{ins,(AES,256,medium,average)}^T$;
- $arg2$ with conclusion:
 $gp = as_{ins,(AES,256,medium,fast)}^T$.

To sum up, we have two arguments setting two different practical goal declarations. Since we can only have one practical goal, we will need to resolve the conflict in the next stage.

11.4. STAGE 4: Conflict Resolution

We have two arguments:

- $arg1$ with conclusion $gp = as_{ins,(AES,256,medium,average)}^T$;
- $arg2$ with conclusion $gp = as_{ins,(AES,256,medium,fast)}^T$.

with two different conclusions. Arguments $arg1$ and $arg2$ attack each other. The conflicting arguments $arg1$ and $arg2$ conclude that $gp = as_{ins,(AES,256,medium,average)}^T$ and $gp = as_{ins,(AES,256,medium,fast)}^T$. Both of them are built on the basis of the same inference rule (ARG4). In the case of such a conflict, the argumentation framework regulates that if all arguments ($arg1$ and $arg2$) are built on the basis of inference rule AS4, they attack each other, the conclusion of $arg1$ is $gp = gm_l$, where $gm_l = x_1$, the conclusion of $arg2$ is $gp = gm_k$, where $gm_k = x_2$, and $VO^{x_2}(x_2) \blacktriangleright VO^{x_1}(x_1)$, then $arg1 > arg2$.

In our case, argument $arg2$ attacks argument $arg1$.

In the first step, we calculate the relative level of promotion of both values:

$$vo_{security}(as_{ins,(AES,256,medium,average)}^T) = \Omega(v_{security}(as_{ins,(AES,256,medium,average)}^T)) = 0.499$$

$$\begin{aligned} vo_{security}(as_{ins,(AES,256,medium,fast)}^T) &= \\ \Omega(v_{security}(as_{ins,(AES,256,medium,fast)}^T)) &= 0.499 \\ vo_{quality}(as_{ins,(AES,256,medium,average)}^T) &= \\ \Omega(v_{quality}(as_{ins,(AES,256,medium,average)}^T)) &= 0.18 \\ vo_{quality}(as_{ins,(AES,256,medium,fast)}^T) &= \\ \Omega(v_{quality}(as_{ins,(AES,256,medium,fast)}^T)) &= 0.18 \\ vo_{speed}(as_{ins,(AES,256,medium,average)}^T) &= \\ \Omega(v_{speed}(as_{ins,(AES,256,medium,average)}^T)) &= 0.094 \\ vo_{speed}(as_{ins,(AES,256,medium,fast)}^T) &= \\ \Omega(v_{speed}(as_{ins,(AES,256,medium,fast)}^T)) &= 0.133 \end{aligned}$$

In the second step, we calculate the cumulative evaluation of the two conflicting arguments:

$$\begin{aligned} o_1 &= vo_{security}(as_{ins,(AES,256,medium,average)}^T) + \\ &vo_{quality}(as_{ins,(AES,256,medium,average)}^T) - \\ &vo_{security}(as_{ins,(AES,256,medium,average)}^T) * \\ &vo_{quality}(as_{ins,(AES,256,medium,average)}^T) = \\ &0.499 + 0.18 - 0.499 * 0.18 = 0.589 \\ \Theta(VO(as_{ins,(AES,256,medium,average)}^T)) &= \\ &o_1 + vo_{speed}(as_{ins,(AES,256,medium,average)}^T) - \\ &o_1 * vo_{speed}(as_{ins,(AES,256,medium,average)}^T) = \\ &0.589 + 0.094 - 0.589 * 0.094 = 0.628 \\ o_2 &= vo_{security}(as_{ins,(AES,256,medium,fast)}^T) + \\ &vo_{quality}(as_{ins,(AES,256,medium,fast)}^T) - \\ &vo_{security}(as_{ins,(AES,256,medium,fast)}^T) * \\ &vo_{quality}(as_{ins,(AES,256,medium,fast)}^T) = \\ &0.499 + 0.18 - 0.499 * 0.18 = 0.589 \\ \Theta(VO(as_{ins,(AES,256,medium,fast)}^T)) &= \\ &o_2 + vo_{speed}(as_{ins,(AES,256,medium,fast)}^T) - \\ &o_2 * vo_{speed}(as_{ins,(AES,256,medium,fast)}^T) = \\ &0.589 + 0.133 - 0.589 * 0.133 = 0.644 \end{aligned}$$

In consequence, we obtain that:

$$\begin{aligned} \Theta(VO(as_{ins,(AES,256,medium,fast)}^T)) &\geq \Theta(VO(as_{ins,(AES,256,medium,average)}^T)) \text{ and} \\ (VO(as_{ins,(AES,256,medium,average)}^T)) &\not\geq (VO(as_{ins,(AES,256,medium,fast)}^T)) \text{ and} \\ (VO(as_{ins,(AES,256,medium,fast)}^T)) &\blacktriangleright (VO(as_{ins,(AES,256,medium,average)}^T)) \end{aligned}$$

which means that $arg2$ defeats $arg1$ ($arg2 > arg1$).

11.5. STAGE 5: Construction of the Final Argument

The last stage involves the construction of the final argument which will become the chosen action that the device Dev_1 will perform in given circumstances.

Since the system established a single practical goal, we can create the following argument $arg3$ based on AS5, which concludes that $\epsilon(as_{(AES,256,medium,fast)^T,ins})$.

The argumentation scheme indicates that action $as_{(AES,256,medium,fast)^T,ins}$ is the single practical goal of the device Dev_1 and therefore this action should be performed.

Finally, the device has to perform the action to use security AES-256, medium level of compression and chose a network that has fast upload speed. ($\epsilon(as_{(AES,256,medium,fast)^T,ins})$).

12. RELATED WORKS

The issue of using the numeric representation of levels to which values are promoted during decision-making has not been much discussed in the literature yet. The author of [24] proposes a simple model of representation of the levels of value promotion, though it is highlighted that “This is not meant to suggest that one can usually balance competing legal values through numerical computations.” Atkinson and Bench-Capon [26] (which is a developed version of an approach from [25] include a proposition of extending the AATS+V model by a simple numeric representation of the agent’s preferences (using the product of values and their weights) and the level of utility regarding a decision-making situation. Our concept of the weight’s and the level of promotion is a little bit more general: In our work the importance of value is represented by weight function, which do not have to be linear. In order to compute the relative levels of promotion of value (vo) we use both: the levels of promotion of values, represented by a numbers from $< 0; 1$) and the weight function which describes, the importance of value, while in [26] the numerical representation (expressed by weights) is constructed on the basis of binary representation of values (values are promoted or not, by a particular decisions). Our approach allows us for estimating not only the weight of value (on the basis of its promotion or demotion) but the relative level of promotion (which takes into consideration weight and the level of promotion). This is the reason why we assumed that weights are not fixed multipliers but the functions (e.g., see the battery example in Section 4.1).

The process of transition from physical value to evaluation of value itself (which in our work is represented by functions form set Φ) in [21] is realized by logic program Δ . There is a significant difference between our approach and that from [21]: our approach is based on the function returning number (the level of promotion of a given value), while in [21] Δ allows for deriving that value is demoted, promoted or neutral. We believe that numerical representation allows for more fine grained representation of the level of promotion of values.

The authors of [2] present a decision-making model for multi-agent systems with agent architecture (named BVG), making it possible to reason on the basis of values. The authors provide the possibility of numeric evaluation of the value level, but they establish constant weights and do not allow for comparing various options promoting various values. However, the system presented in [2] contains an important feature which is absent in our work—the system has a possibility of representation of failure of the action (the action does not leads to the planned state of affairs). This is an interesting property, which should be discussed in the future. Several other works (e.g., [32,33]) introduce methods of numeric representation of the agent’s preferences, though lacking any link to values or the possibility of autonomous goal-setting.

An interesting mechanism of numeric representation of the levels of value promotion and relations between the same is proposed in [34]. It is based on a premise largely similar to that described herein (decision-making options promote values to a different extent, and the levels thereof can be compared). However, despite said similarities, the two mechanisms differ in certain key aspects: Sartor introduces the concept of utility which is absent in our model as in our opinion it provides no additional benefits: in our model, the level of value promotion is determined exactly on the basis of its utility. Moreover, the weight function proposed by Sartor is linear which, as observed in Section 4.1, often fails to correspond to real-life decision-making situations. The two models (ours and Sartor’s) also differ in terms of the manner in which certain values are determined (e.g., the total level of value promotion which in Sartor’s approach is obtained through simple addition). Furthermore, unlike ourselves, Sartor does not introduce any inference mechanism and offers no formal, in-depth analysis of the model he proposes.

An interesting decision-making model for an autonomous vehicle is proposed in [35] where the authors describe a very intricate framework based on the partially observable Markov decision process. The authors’ primary goal is to provide a general solution for AV decision-making at any intersection. The model is developed in such a way so as to interact with any and unspecified number of other road users while at the same time accounting for the unpredictability of their behavior. The model is based on a value function rooted in the expected reward for a given policy. The value function allows the determination of the *total regret*, whose minimization is the primary goal of the model. The solution proposed in [35] is dedicated to a very specific type of an autonomous device and provides a very advanced decision-making model in the context of behavior at a road intersection, but does not account for a broader decision-making context based on the promotion of a variety of values and does not allow for autonomous goal-setting by a device.

13. DISCUSSION AND CONCLUSIONS

The framework included in [14] allows for the modeling of reasoning in autonomous systems. Regretfully, its practical implementation requires a declaration of a large number of orders describing relations between levels to which various situations promote various values and sets of values. With real decision-making problems,

the declaration of such a high number of orders is very challenging and can only be feasible in the case of a situation with relatively small sets of values and actions which are possible to perform. The main objective of our work is to propose modifications of the framework from [14] which would allow a facilitated implementation of the decision-making systems for autonomous agents. Our model can be seen as a kind of theoretical framework constituting a formal background for decision-making mechanisms. The basics of the framework are constructed in a flexible way, which allows for the development of the model by adding new functions to sets Φ , Ω , Λ , by utilization of various methods of obtaining these functions (including machine learning mechanisms), by adding new kinds of goals, inference rules, etc. Moreover, in particular kinds of cases (described in Section 8), the basics of the model can function as a background for optimization algorithms. Such flexibility allows for adjusting the model to various kinds of autonomous devices.

While making a decision, a person intuitively evaluates available decision options, dividing them into better and worse ones (as presented in [14] and other works), but does not attach any numeric values to them. This paper introduces a modified approach, where the level to which particular situations promote various values is represented as a number from the range $\langle 0; 1 \rangle$. Though unlike a typical human approach, we believe it is much more natural for all kinds of technical devices. The proper definition of functions from the set Φ allows for automatic evaluation of the decision options. The modification we propose allows for a substantial reduction of the number of orderings (declarations of O and OR will not be necessary) because the levels to which particular values are promoted can be easily compared; moreover, the lack of necessity to search large order sets may significantly accelerate the decision-making process. The proposed mechanism of determining the cumulative evaluation of particular situations (decision options), joined by the weight functions, makes it possible to compare complex situations promoting various values to various levels. Although this entails the necessity to declare function sets Φ and Ω , it is worth noticing that for situations in which the user or the creator of the system does not know which function to use, the system may use default functions or machine learning mechanisms which, on the basis of the previous data, can learn the course of the Φ and Ω functions.

The problem of reasoning with values and goals has been discussed by a number of authors; however, in our opinion, there are still many issues requiring further development. One of the most important drawbacks of the existing models consists in the lack of comprehensive discussion of the definition of values, goals, and relations between them. Especially the problem of relations between values and goals has been, in our view, insufficiently analyzed and stands in need of broader discussion.

The main objective of this paper is to present and discuss a new view of the problem of the representation of values and goals in reasoning. We have pointed out some elements in which our idea differs from other concepts of utilization of values in reasoning, particularly those presented in [7] and [21]. We do not intend to prove wrong the ideas included in the aforementioned works or claim they are irrelevant for the representation of various aspects of reasoning with goals and values; it is rather an attempt to demonstrate a new and more intuitive view of the problem. Obviously, the notion

of intuitiveness is a subjective one, but we believe that establishing connection between the minimal acceptable levels of promotion of values, understood as goals and motivations of decision makers, allow us to establish a simpler and more abstract (trans-situational) way of representing the goals of agents. On the basis of such goals the decisions can be made.

The model presented in this paper provides a comprehensive tool for representation of values and goals in practical reasoning. Although there are decision-making models which allow for reasoning about values and goals (e.g., [21,36]), we are convinced, that our model of relations between goals and values makes a clear connection between those two concepts and allows us to create an autonomous system which can independently set itself a goal. We believe this is the most important novelty of the paper.

Most models presented in the literature view values as promoted (or demoted) by a change of a state of affairs. Our idea is slightly different: in the model we propose, values are promoted by a particular state of affairs or actions leading to a particular state of affairs, while the level of their promotion by a change of states can be seen as the difference between the promotion of values by the initial state and the action which leads to the target state of affairs. To our minds, such an idea is more intuitive and allows for a more adequate representation of human reasoning in which not only a change of states can be the subject of valuation, but also a group of given states of affairs. Such assumptions make our approach closer to the multi-criteria decision-making problems and we believe that our approach can be seen as a kind of bridge between value-based and multi-criteria decision systems.

On the basis of the above definitions, value-based inference patterns have been introduced with the use of argumentation schemes. The above definition of values allows for the formulation of the concept of abstract, material, and practical goals, which in consequence leads to an improved description of various kinds of teleological reasoning.

Most models presented in the literature view goals as a particular state of affairs which the agent is going to reach. In my conception, an abstract goal is the set of extents to which values should be promoted. On the basis of such a goal, the agent can infer which action brings about the desired goal. This view on the process of teleological reasoning allows for autonomous formulation of a goal by the agent.

Generally speaking, the model presented in the paper introduces a bridge between two separate but connected concepts of values and goals, providing a precise description of the relations between them as well as a possibility of reasoning about goals which, in consequence, allows us to model the motivations of the agent.

There are a few important issues calling for further discussion which I am going to elaborate in my future work. The first one is the discussion of the problem of joint actions with multiple agents. We have assumed that by an action we understand not only a single activity performed by a single agent, but also a joint action made by a group of agents. This is, however, the topic which requires further elaboration (the most comprehensive discussion of the problem is presented in [21]). The second issue is the potential possibility of the incorporation of the model into any formal framework of legal argumentation (ASPIC+ [31] or Carneades [37]). I see this

possibility as important in the context of the implementation of this model, but it still requires some elaboration which can be performed in future works. The third issue is the discussion of the problem of decision-making in a legally-regulated environment, including an analysis of various reasoning mechanisms [38,39], conflict resolution [40,41], interpretation [42,43], and other.

CONFLICTS OF INTEREST

The authors declare of no conflicts of interest.

AUTHORS' CONTRIBUTIONS

Tomasz Zurek: 80% (The idea, formal modeling, proofs, analysis, discussion), Michail Mokkas: 20% (Case analysis).

REFERENCES

- [1] S.O. Hansson, Decision theory – a brief introduction, 1994. <https://people.kth.se/~soh/decisiontheory.pdf>
- [2] L. Antunes, H. Coelho, Decisions based upon multiple values: the BVG agent architecture, in: P. Barahona, J.J. Alferes (Eds.), *Progress in Artificial Intelligence: 9th Portuguese Conference on Artificial Intelligence, EPIA '99*, Springer, Berlin, Heidelberg, Germany, 1999, pp. 297–311.
- [3] S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, New Jersey, USA 2009.
- [4] L. Amgoud, Argumentation for decision making, in: G. Simari, I. Rahwan (Eds.), *Argumentation in Artificial Intelligence*, Springer, Boston, MA, USA, 2009, pp. 301–320.
- [5] E. Ferretti, A possibilistic defeasible logic programming approach to argumentation-based decision-making, *J. Exp. Theor. Artif. Intell.* 26 (2014), 519–550.
- [6] R. Girdle, Decision support for practical reasoning, in: C. Reed, T.J. Norman (Eds.), *Argumentation Machines: New Frontiers in Argument and Computation*, Springer, Dordrecht, Netherlands, 2004, pp. 55–83.
- [7] K. Atkinson, T. Bench-Capon, Practical reasoning as presumptive argumentation using action based alternating transition systems, *Artif. Intell.* 171 (2007), 855–874.
- [8] T. Bench-Capon, K. Atkinson, P. McBurney, Using argumentation to model agent decision making in economic experiments, *Autonom. Agents Multi-Agent Syst.* 25 (2012), 183–208.
- [9] T.J.M. Bench-Capon, P.E. Dunne, *Argumentation in artificial intelligence*, *Artif. Intell.* 171 (2007), 619–641.
- [10] I. Rahwan, G.R. Simari, *Argumentation in Artificial Intelligence*, first ed., Springer Publishing Company, Boston, MA, USA, 2009.
- [11] T.L. Weide, Practical reasoning using values, in: P. McBurney (Ed.), *Argumentation in Multi-Agent Systems*, Lecture Notes in Computer Science, vol. 6057, Springer, Berlin, Heidelberg, Germany, 2010, pp. 79–93.
- [12] A.S. Rao, M.P. Georgeff, BDI agents: from theory to practice, in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, USA 1995, pp. 312–319. <https://www.aaai.org/Papers/ICMAS/1995/ICMAS95-042.pdf>
- [13] K.V. Hindriks, Programming rational agents in GOAL, in: A.E.F. Seghrouchni (Ed.), *Multi-Agent Programming: Languages, Tools and Applications*, Springer, Boston, MA, USA, 2009, pp. 119–157.
- [14] T. Zurek, Goals, values, and reasoning, *Expert Syst. Appl.* 71 (2017), 442–456.
- [15] T. Bench-Capon, S. Modgil, Norms and value based reasoning: justifying compliance and violation, *Artif. Intell. Law.* 25 (2017), 29–64.
- [16] T. Zurek, M. Mokkas, Modeling value-based reasoning for autonomous agents, in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, ACSIS, Prague, Czech Republic 2017, pp. 259–262.
- [17] B. Ksiezopolski, T. Zurek, M. Mokkas, Quality of protection evaluation of security mechanisms, *Sci. World J.* 2014 (2014), 1–18.
- [18] B. Ksiezopolski, T. Zurek, M. Mokkas, On the modelling of context-aware security for mobile devices, *Mob. Inf. Syst.* 2016 (2016), 1–16.
- [19] B. Ksiezopolski, QoP-ML: quality of protection modelling language for cryptographic protocols, *Comput. Security.* 31 (2012), 569–596.
- [20] B. Ksiezopolski, *Multilevel Modeling of Secure Systems in QoP-ML*, CRC Press, New York, NY, USA, 2015.
- [21] K. Atkinson, T. Bench-Capon, States, goals and values: revisiting practical reasoning, in *Proceedings of 11th International Workshop on Argumentation in Multi-Agent Systems*, Paris, France 2014.
- [22] L. Amgoud, J.-F. Bonnefon, H. Prade, An argumentation-based approach to multiple criteria decision, in: L. Godo (Ed.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Springer, Berlin, Heidelberg, Germany, 2005, pp. 269–280.
- [23] M.J. Rohan, A rose by any name? The values construct, *Pers. Soc. Psychol. Rev.* 4 (2000), 255–277.
- [24] G. Sartor, Doing justice to rights and values: teleological reasoning and proportionality, *Artif. Intell. Law* 18 (2010), 175–215.
- [25] A. Chorley, T. Bench-Capon, An empirical investigation of reasoning with legal cases through theory construction and application, *Artif. Intell. Law.* 13 (2005), 323–371.
- [26] K. Atkinson, T.J.M. Bench-Capon, Value based reasoning and the actions of others, in *22nd European Conference on Artificial Intelligence (ECAI 2016)*, Including Prestigious Applications of Artificial Intelligence (PAIS 2016), The Hague, The Netherlands, 2016, pp. 680–688.
- [27] H. Prakken, On the nature of argument schemes. In C.A. Reed, C. Tindale, (Eds.), *Dialectics, Dialogue and Argumentation. An Examination of Douglas Walton's Theories of Reasoning and Argument* College Publications, London, (2010), pp. 167–185.
- [28] R.L. Keeney, H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*, Cambridge University Press, Englewood Cliffs, New Jersey, USA, 1993.
- [29] P. Lee, Learning from Tay's introduction, 2016. <https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/#sm.0000r0qlzzxm9fm4vky4imwx36th.Mar>
- [30] T.J.M. Bench-Capon, S. Modgil, When and how to violate norms, in *The Twenty-Ninth Annual Conference on Legal Knowledge and Information Systems (JURIX 2016)*, Nice, France, 2016, pp. 43–52.
- [31] S. Modgil, H. Prakken, The ASPIC+ framework for structured argumentation: a tutorial, *Argument Comput.* 5 (2014), 31–62.
- [32] P. Giorgini, Reasoning with goal models, in *Proceedings of the 21st International Conference on Conceptual Modeling (ER '02)*, London, UK, 2002, pp. 167–181. <http://dl.acm.org/citation.cfm?id=647525.725913>

- [33] J. Thangarajah, L. Padgham, Computationally effective reasoning about goal interactions, *J. Automat. Reason.* 47 (2011), 17–56.
- [34] G. Sartor, A quantitative approach to proportionality, in: G. Bongiovanni (Ed.), *Handbook of Legal Reasoning and Argumentation*, Springer, Dordrecht, Netherlands, 2018, pp. 613–636.
- [35] S. Zilberstein, K. Hollins, W. Stefan, J. Witwicki, Online decision-making for scalable autonomous systems, in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, Melbourne, Australia 2017, pp. 4768–4774.
- [36] K. Atkinson, T.J.M. Bench-Capon, States, goals and values: revisiting practical reasoning, *Argument Comput.* 7 (2016), 135–154.
- [37] T.F. Gordon, H. Prakken, D. Walton, The Carneades model of argument and burden of proof, *Artif. Intell.* 171 (2007), 875–896.
- [38] T. Zurek, Modelling of a fortiori reasoning, *Expert Syst. Appl.* 39 (2012), 10772–10779.
- [39] T. Zurek, Instrumental inference in legal expert system, in *The Twenty-Fourth Annual Conference Legal Knowledge and Information Systems (JURIX 2011)*, University of Vienna, Vienna, Austria, 2011, pp. 155–159.
- [40] T. Zurek, Modeling conflicts between legal rules, in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, *Annals of Computer Science and Information Systems*, IEEE, Gdansk, Poland 2016, vol. 8, pp. 393–402.
- [41] T. Zurek, Model of argument from social importance, in *The Twenty-Seventh Annual Conference on Legal Knowledge and Information Systems (JURIX 2014)*, Jagiellonian University, Krakow, Poland, Krakow, Poland 2014, pp. 23–28.
- [42] M. Araszkiewicz, T. Zurek, Interpreting agents, in *The Twenty-Ninth Annual Conference on Legal Knowledge and Information Systems (JURIX 2016)*, 2016, pp. 13–22.
- [43] T. Zurek, M. Araszkiewicz, Modeling teleological interpretation, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, ACM, 2013, pp. 160–168.

APPENDIX: PROOFS

A.0.1 Independence of the Ordering of Values

Proof. The proof is trivial. If set V^Z includes one or two values, then the order is of no importance, for a set consisting of three elements, let us assume that $V^Z = \{v_1, v_2, v_3\}$ and $VO^Z(x) = \{vo_1(x), vo_2(x), vo_3(x)\}$.

Then:

For any accumulation order $vo_1(x), vo_2(x), vo_3(x)$ the result is the same:

If we accumulate $vo_1(x)$ with $vo_2(x)$, and the result with $vo_3(x)$, then

$$\Theta(VO^Z(X)) = (vo_1(x) + vo_2(x) - (vo_1(x)vo_2(x))) + vo_3(x) - (vo_1(x) + vo_2(x) - (vo_1(x)vo_2(x)))vo_3(x) = vo_1(x) + vo_2(x) + vo_3(x) - vo_1(x)vo_2(x) - vo_1(x)vo_3(x) - vo_2(x)vo_3(x) + vo_1(x)vo_2(x)vo_3(x).$$

If we accumulate $vo_1(x)$ with $vo_3(x)$, and the result with $vo_2(x)$, then

$$\Theta(VO^Z(X)) = (vo_1(x) + vo_3(x) - (vo_1(x)vo_3(x))) + vo_2(x) - (vo_1(x) + vo_3(x) - (vo_1(x)vo_3(x)))vo_2(x) = vo_1(x) + vo_2(x) + vo_3(x) - vo_1(x)vo_2(x) - vo_1(x)vo_3(x) - vo_2(x)vo_3(x) + vo_1(x)vo_2(x)vo_3(x).$$

If we accumulate $vo_2(x)$ with $vo_3(x)$, and the result with $vo_1(x)$, then

$$\Theta(VO^Z(X)) = (vo_2(x) + vo_3(x) - (vo_2(x)vo_3(x))) + vo_1(x) - (vo_2(x) + vo_3(x) - (vo_2(x)vo_3(x)))vo_1(x) = vo_1(x) + vo_2(x) + vo_3(x) - vo_1(x)vo_2(x) - vo_1(x)vo_3(x) - vo_2(x)vo_3(x) + vo_1(x)vo_2(x)vo_3(x).$$

For a higher number of values in V^Z , further accumulations of results may be put under $vo_1(x), vo_2(x)$, or $vo_3(x)$ and determined with the use of the same formulae, and hence their order will be insignificant. \square

A.0.2. Proof that Θ is Monotonic

Proof. Assuming that the cumulative evaluation of the relative levels of promotion of the values from set V^Z in a situation x is $\Theta(VO^Z(x))$, then the cumulative evaluation of the relative levels of promotion of the values from set $V^Z \cup v_j$ will be: $\Theta(VO^{Z+v_j}(x)) = \Theta(VO^Z(x)) + v_j(x) - \Theta(VO^Z(x))v_j(x)$. Since $\Theta(VO^Z(x))$ and $v_j(x)$ are from the range $\langle 0; 1 \rangle$, then $\Theta(VO^Z(x))v_j(x) \leq \Theta(VO^Z(x))$ and $\Theta(VO^Z(x))v_j(x) \leq v_j(x)$. In consequence, $\Theta(VO^{Z+v_j}(x)) \geq \Theta(VO^Z(x))$. \square

A.0.3. Proof that ORO Satisfies Second Property of Relation OR .

Proof. Θ is monotonic.

If $VO^X \subseteq VO^Z$ and if $VO^X(x_1) \geq VO^Y(x_2)$, then the cumulative relative evaluation of the level to which a situation x_1 promotes value set VO^Z cannot be lower than the evaluation VO^X : $\Theta(VO^Z(x_1)) \geq \Theta(VO^X(x_1))$. \square

A.0.4. Proof that Relations between O and ORO are the Same as between O and OR .

- 1 The first relation will be fulfilled if: $v_i(x_1) > v_i(x_2) \Rightarrow \{vo_i(x_1)\} \geq \{vo_i(x_2)\}$.

Proof. Since every function in Ω is increasing and $v_i(x_1) > v_i(x_2)$, then $vo_i(x_1) \geq vo_i(x_2)$. Since function Θ is non-decreasing and $vo_i(x_1) \geq vo_i(x_2)$, then $\Theta(vo_i(x_1)) \geq \Theta(vo_i(x_2))$. Hence if $v_i(x_1) > v_i(x_2)$, then $\{vo_i(x_1)\} \geq \{vo_i(x_2)\}$. \square

- 2 The second relation will be fulfilled if:

$$(\forall_{v_z \in V^{x_2}} (v_z(x_1) \geq v_z(x_2))) \wedge \exists_{v_t \in V^{x_2}} (v_t(x_1) > v_t(x_2)) \Rightarrow VO^{x_1}(x_1) \geq VO^{x_2}(x_2)$$

Proof. Since every function in Ω is increasing and $(\forall_{v_z \in V^{x_2}} (v_z(x_1) \geq v_z(x_2)))$, then (1) $(\forall_{v_z \in V^{x_2}} (vo_z(x_1) \geq vo_z(x_2)))$ and (2) $\exists_{v_t \in V^{x_2}} (vo_t(x_1) > vo_t(x_2))$. Since function Θ is non-decreasing and monotonic and (1) and (2) are true, then $\Theta(VO^{x_1}(x_1)) > \Theta(VO^{x_2}(x_2))$, which (in consequence) gives: $VO^{x_1}(x_1) \geq VO^{x_2}(x_2)$. \square

- 3 The third relation will be introduced differently than in [14]:

$$\begin{aligned} (V^{x_2}(x_2) = \{V^{x_1 \geq x_2}(x_2) \cup V^{x_2 \geq x_1}(x_2)\}) \\ (V^{x_1}(x_1) = \{V^{x_2 \geq x_1}(x_1) \cup V^{x_1 \geq x_2}(x_1)\}) \\ \exists_{VO^s(x_1) \subseteq VO^{x_1 \geq x_2}(x_1)} (VO^s(x_1) \blacktriangleright VO^{x_2 \geq x_1}(x_2)) \Rightarrow \\ VO^{x_1}(x_1) \blacktriangleright VO^{x_2}(x_2) \end{aligned}$$

Proof. The above formula may be also presented as:

$$\begin{aligned} \exists_{VO^s(x_1) \subseteq VO^{x_1 \geq x_2}(x_1)} (\Theta(VO^s(x_1)) > \\ \Theta(VO^{x_2 \geq x_1}(x_2))) \Rightarrow VO^{x_1}(x_1) VO^{x_2}(x_2) \end{aligned}$$

The set of values promoted by a situation x_2 consists of two sets: a set of values promoted by x_2 to a lower level than x_1 ($VO^{x_1 \geq x_2}(x_2)$) and the remaining values (that is, values promoted by x_2 to a higher level than by x_1 and values not promoted by x_1 whatsoever: $VO^{x_2 \geq x_1}(x_2)$). Since function Θ is increasing and monotonic, then $\Theta(VO^{x_1 \geq x_2}(x_1)) \geq \Theta(VO^{x_1 \geq x_2}(x_2))$. If in the set of values promoted by x_1 to a higher level than by x_2 there exists a subset $V^s(x_1)$ which is preferred over the entire set $VO^{x_2 \geq x_1}(x_2)$, that is $\Theta(VO^s(x_1)) > \Theta(VO^{x_2 \geq x_1}(x_2))$, then considering the monotonic nature of the function Θ , $\Theta(VO^{x_1}(x_1)) > \Theta(VO^{x_2}(x_2))$, that is $VO^{x_1}(x_1) \blacktriangleright VO^{x_2}(x_2)$, will be true. \square