An argumentation-based approach for decision making

Jann Müller
SAP Research
Belfast BT3 9DT
United Kingdom
jann.mueller@sap.com

Anthony Hunter
University College London
London WC1E 6BT
United Kingdom
a.hunter@cs.ucl.ac.uk

***Abstract*—The formalisation of design decisions serves two purposes: To support the decision maker in choosing which decision to take (decision analysis), and to document the reasons behind decisions for future reference (decision documentation). Approaches which solve the latter task involve a semi-formal pattern of documenting the reasons for and against each of the options, but they generally do not allow an automation of the decision making process. Approaches which solve the former task use a mathematical model of the problem, in which each option is evaluated numerically with respect to some relevant criteria, but they do not support documentation. We investigate the use of argumentation to both analyse and document decisions, solving both tasks with the same method. Additionally, the system we present is able to generate decisions for analysis, instead of relying on a predefined input of options. We collaborated with an aerospace manufacturer to identify common problems in the industry and to create realistic examples from the engineering domain.**

We show that our system subsumes a certain class of multi criteria decision making problems and that it improves upon previous argumentation-based decision making systems by adding the capability to generate decisions and by clearly defining the semantics used to choose accepted arguments.

***Keywords*-argumentation; decision making; decision documentation; decision analysis; generating decisions**

I. Introduction

Engineering design processes, such as those practised in the aerospace industry, are often complex and long-running. They involve a multitude of decisions, many of which affect subsequent steps in the process. Automated methods are commonly used to handle the complexity and interrelatedness of the decisions. The formal foundation of these methods has been studied widely in the literature [1]. However, while the quantitative analysis of decisions helps to manage the complexity of individual decisions, it does not address requirements that result from the long duration of the overall process. These requirements relate to decision documentation and include justifiability of decisions and traceability of the impact decisions have on one another. Information of this kind is recorded in a less rigorous manner, for example as text documents. The two tasks of analysing and documenting decisions are solved using two different methods. Therefore, a single human reasoning process (making a particular design decision) results in the production not only of two different artifacts, but of two entirely different models of that decision. We claim that an argumentation can serve as a common foundation for both the analysis and the documentation of decisions.

We present an argumentation-based method of decision making. Computational models of argument are being developed to formalise human reasoning patterns, especially when handling conflicting information [2], [3]. Previous work by Amgoud and Prade [4] uses argumentation to reason about the consequences of decisions. We extend their abstract framework in two ways: By adding the ability to generate decisions, and by clearly specifying a semantics for selecting accepted arguments.

Our system is a method for decision analysis. Unlike in earlier proposals, decisions in our framework are modeled as sets of literals, rather than as single literals. This means that they can partially overlap, resulting in a more finely tuned set of decisions. The need for this kind of analysis was identified during our collaboration with an aerospace manufacturer.

Our system also provides output that can be used to create decision documentation. It uses formal logic to reason with arguments and counterarguments. Because these arguments are generated from structured knowledge (in the form of rules), they can with little additional effort be transformed into an ontology-based format. This is because ontology standards are based upon a formal foundation of description logics [5].

The rest of the paper is organised as follows: We give an overview of a class of multi criteria decision analysis, and outline some possible methods of documenting decisions (section II). In section III, we discuss some of the problems of the current approach in more detail. In particular, we present a list of common problems that companies face in the decision analysis and documentation process. In section IV, we present our solution and our main results. Section V contains a formal discussion of the relation between our system and multi criteria decision analysis. In the following section (VI), we evaluate our system with respect to the requirements for a method of analysing and documenting decisions. The paper concludes with a discussion of related work in section VII.

II. Current Approach

This section summarises the current approach to the two tasks of modeling and documenting decisions.

A. Multi criteria decision making

A variety of formal definitions exists for multi criteria decision making. We will focus our analysis on the class of problems characterised by the definition below. The criteria C are represented by functions D that map decisions onto numerical values, representing the quality of a decision w.r.t a criterion. The values for each criterion are then aggregated to obtain an overall result (or a ranking) that determines the most favourable decision.

Definition 1 (Multi criteria decision problem)**.** *A multi criteria decision problem $P = (D, C, agg)$ consists of*

1) *A sequence of decisions $D = (d_1, \ldots, d_n)$*
2) *A sequence of criteria $C = (c_1, \ldots, c_k)$. Each $c_i \in C$ is a function $c_i : D \to \mathbb{R}$*
3) *An aggregation function $agg : \mathbb{R}^{|D|*|C|} \to \mathbb{R}^{|D|}$*

The set of all multi criteria decision problems is called MCD. We denote with V_P the two-dimensional vector of the criteria values for each decision:

$$V_P = \begin{bmatrix} c_1(d_1) & \ldots & c_k(d_1) \\ \vdots & \ddots & \\ c_1(d_n) & & c_k(d_n) \end{bmatrix}$$

The actual rating of a decision for a particular criterion is carried out by the decision maker who creates the table. In order to achieve consistency and accountability in the decision making process, additional documentation is required to justify decisions for a later verification. The numerical model alone does not explain *why* the criteria were assigned their values.

Example 1. *Table 1 illustrates the problem of choosing a material for a wing component of an airplane. There are four possible decisions, aluminium, plastics, steel, and composite materials. The two criteria are weight and cost. The example demonstrates how the choice of agg influences the results: If one considers the sum of the criteria, aluminium is the first choice, but if one is instead interested in maximising the best criterion, then composites and steel are tied for first place, and aluminium is last. Aluminium has better results for both criteria than plastics, therefore plastics is dominated by aluminum.*

A *preferred decision* is a decision that as good as or better than all other decisions.

Definition 2 (Preferred Decision)**.** *Given a decision system $P = (D, C, agg)$, a decision $d_i \in D$ is* preferred *iff for all $d_j \in D$*

$$agg(V_P)_j \leq agg(V_P)_i$$

Table I
A Multi Criteria Decision Making Example

	Criteria[a]		*Aggregations*	
	Weight	*Cost*	Σ^b	$\max^c$
Aluminium	0.4	0.7	1.1	0
Plastics	0.3	0.6	0.9	0
Steel	0.2	0.8	1.0	1
Composites	0.7	0.2	0.9	1

[a] The higher the value, the better this criterion is met, e.g. low weight will result in a high value for $weight$.
[b] Σ is the sum of all criteria for one decision.
[c] $\max(d)$ is the number of criteria in which d has the highest value.

Example 2. *In the example given in table 1, aluminium is preferred if we choose Σ as the aggregation. Otherwise, steel or composites would be preferred.*

B. Informal Documentation of Decisions

As outlined in section I, the primary reasons for documenting design decisions are consistency and accountability. The design decision process needs to be consistent throughout organisations and, with regards to regulations, the entire industry. Consistency means that given a specific problem, any decision maker would ideally come to the same conclusion. Records need to be kept in order to verify that the decision making process is consistent.

Design decisions in the aerospace industry are often complex. They are also part of an iterative design process, which means that decisions may need to be revised to account for previously unconsidered factors or for changed requirements.

In our collaboration with one major aerospace manufacturer, we found that discussions about design questions were primarily carried out in emails and personal meetings. Only when a decision was made, it was documented in a central repository used to manage the design process. To create this documentation, some of the information contained in the emails had to be duplicated, while the rest remained only on the email server and was thus not readily available to a search of the structured repository. Having to duplicate information carries the risk of introducing errors, sometimes simply by using a slightly different wording.

This process also entails that alternative decisions which had been discussed would only be documented informally. Later on in the process, a changing requirement might lead to a re-evaluation of previously made decisions. In this case, the alternatives have to be retrieved from the unstructured documentation. This is a labour intensive process. It is also error prone, especially after several iterations of the design. The retrieval of decision rationale could therefore benfit greatly from a formal, structured documentation.

Contract and claims management are another use case for decision documentation. At the beginning of each phase in the project life cycle, it needs to be shown that all of the requirements of the previous stage have been fulfilled. Design documentation is used to show how each requirement is addressed. Here, the same issues that were described in the previous paragraph arise.

III. Problems With Current Approach

The two problems outlined at the beginning of this paper are the modeling and the documentation of decisions. Having presented common solutions to each of those problems, we are now going to highlight their shortcomings.

1, Opaque resoning process: For each of the potential decisions, a set of criteria has to be evaluated. Assigning values to decisions accounts for a large part of the effort involved in making decisions with an MCDM approach. It usually represents the outcome of some reasoning process, which itself does not appear in the final model and needs to be documented separately.

2, Local optimum: An MCDM model can only identify the best of a predefined set of options. It is possible that there exists a better decision that was not part of the model. Therefore, MCDM models have the inherent risk of producing only a local optimum.

3, Proprietary documentation formats: There is no standard method of documenting decisions. This prevents the development of standard tools to support decision documentation.

4, Manual analysis: If decisions are documented informally, there is no underlying model on which an automated analysis of decisions could be performed. With an automated approach, one could, for example, immediately spot conflicting assumptions made by two different engineers.

5, Costly retrieval of documentation: Most of the documentation relating to the process of decision making is documented in an unstructured way. The effort required to find a particular piece of information in an unstructured repository is much higher than that of finding it in a structured repository. If, at the end of a project, there are claims that some requirements have not been met, the entire documentation related to that subcomponent has to be read by an engineer in order to verify that either the requirement has actually been met or that the requirement was defined differently in the contract.

Both of the problems associated with multi criteria decision making (1-2) relate to the fact that creating the model is the actual challenge in MCDM. Once a model has been created, the actual evaluation is simply applying a set of predefined mathematical operations. The informal documentation of decisions, on the other hand, is limited in terms of automated processing of information (3-5).

IV. Argumentation Decision Framework (ADS)

We present a framework for decision making using argumentation. More precisely, given a set of possible decisions and a set of goals, we are going to use argumentation to construct arguments in favour of and against each decision, with the aim of identifying the best decision. The best decision is the one that satisfies the most goals. However, this should not be the only criterion for selecting a decision: Any formal method of decision making depends on a model, a formalised representation of the problem. The quality of the model affects the quality of the decision. There are two factors which determine the quality of the model: The quality of the information that the model was built on, and the quality (correctness) of the model itself. Because of these two inherent risk factors, we argue that any formal method of decision making should not only identify the best decision, but it should also present a justification for the outcome, so that the influence on the decision making process of potential flaws in the model can be traced.

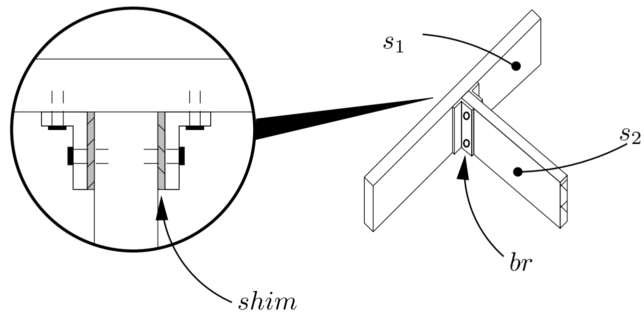

Figure 1. Joining Two Structures

Formal methods of argumentation are suitable for this purpose. They produce not only a claim, but also a description of how the available information was used to arrive at that claim, and how any counterarguments were addressed. Our decision making framework uses argumentation to reason about the possible outcomes of decisions. It is based on previous work by Amgoud and Prade [4], but it has been enhanced in some key points which we will discuss below in section VII.

A. Argumentation Graphs

We reproduce the relevant definitions from Dung's framework [6].

Definition 3 (Argument Graph)**.** *An* argument graph *is a pair* $\langle \mathcal{A}, defeat \rangle$. $\mathcal{A}$ *is a set of arguments and* $attacks \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary relation of defeat. An argument* $A \in \mathcal{A}$ defeats *an argument* $B \in \mathcal{A}$ *if* $(A, B) \in defeat$*.*

Example 3. *Imagine we are faced with the problem of designing a structure (*t*) that has two components,* s_1 *and* s_2 *as illustrated in Figure 1.* s_1 *and* s_2 *are joined with a bracket fixed by bolts, bb. Our task is to decide which bolts to use, whether or not to use a shim, a thin sheet of metal, between*

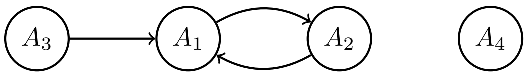

Figure 2. Argument graph for examples 3 and 6

the bracket and the components, and how many layers of varnish to apply. Some requirements are to maintain the structural integrity of t and to achieve a high resistance to corrosion. The first decision we consider, D_1, is to use steel/titanium bolts, not to use a shim, and to apply two layers of varnish to the product. The following arguments might be put forward:

- A_1 *Not using a shim means that the structure remains balanced. Therefore, it will not be damaged.*
- A_2 *Steel/titanium bolts cause microscopic fractures in s_1 and s_2, resulting in damage to the structure.*
- A_3 *Steel/titanium bolts are too strong for the material that s_1 is made of, so there will be microscopic fractures.*
- A_4 *The structure has a high corrosion resistance, because two layers of varnish are used.*

The argument graph for this example is shown in Figure 2.

Definition 4 (Conflict-free, defence)**.** *Let $\mathcal{B} \subseteq \mathcal{A}$.*

- *$\mathcal{B}$ is* conflict-free *iff there exist no A_i, A_j in $\mathcal{B}$ such that A_i defeats A_j.*
- *$\mathcal{B}$* defends *an argument A_i iff for each argument A_j $\mathcal{A}$: If A_j defeats A_i, then there exists an argument A_k in $\mathcal{B}$ such that A_k defeats A_j.*

Definition 5 (Acceptability semantics)**.** *Let $\mathcal{B}$ be a conflict-free set of arguments, and let $\mathcal{F} : 2^{\mathcal{A}} \mapsto 2^{\mathcal{A}}$ be a function such that $\mathcal{F}(\mathcal{B}) = \{A \mid \mathcal{B}\ defends\ A\}$. $\mathcal{B}$ is a* grounded extension *iff it is the least (w.r.t set-inclusion) fixpoint of $\mathcal{F}$.*

Every argumentation framework has a single grounded extension.

Example 4. *The grounded extension of example 3 is $\{A_2, A_3, A_4\}$.*

B. Argumentation Formalism

The logic we use in our argumentation system is a simplified version of ASPIC+ [7]. In our system, we only have defeasible rules, and we only consider two kinds of attack.
An argument consists of a conclusion, a set of assumptions, and a method of inferring the conclusion from the assumptions. We use a defeasible logic to represent arguments. Let $\mathcal{L}$ be a logical language. An *argumentation system* is a tuple $(\mathcal{A}, \mathcal{K})$ such that $\mathcal{A} \subseteq \mathcal{L}$ and $\mathcal{K}$ is a set of rules of the form $\alpha_1, \ldots, \alpha_n \Rightarrow \alpha_n$ where each $alpha_i$ is in $\mathcal{L}$. Negation is denoted with $\neg$. The only inference rule we use is modus ponens. $\vdash_{\mathcal{K}}$ denotes inference over a set of rules $\mathcal{K}$. Axioms are written as rules with an empty body.

Definition 6 (Argument)**.** *Let $(\mathcal{A}, \mathcal{K})$ be an argumentation system. Let $\Phi \subseteq \mathcal{A}$. The tuple $\langle \Phi, \alpha \rangle$ is an* argument *iff $\Phi \vdash_{\mathcal{K}} \alpha$.*

Φ is the set of premises and α is the conclusion. From now on, we will omit the subscript and simply write $\vdash$ if there is no danger of ambiguity. Arguments can be attacked in two ways, on their premises and on their conclusion.

Definition 7 (Argument Attack)**.** *Let $A_1 = \langle \Phi_1, \alpha_1 \rangle$ and $A_2 = \langle \Phi_2, \alpha_2 \rangle$ be arguments.*

1) A_1 rebuts A_2 *iff $\alpha_1 = \neg\alpha_2$*
2) A_1 undercuts A_2 *iff there exists an argument $\langle \Phi_2', \alpha_2' \rangle$ with $\Phi_2' \subseteq \Phi_2$ and $\alpha_1 = \neg\alpha_2'$*

An undercut is a rebuttal of a subargument. Because $\neg$ is symmetrical, rebuttals are always mutual.

Example 5. *Based on our previous example, we can write down the rules as follows:*

r_1 $material(bb, s-t) \Rightarrow fractured(s_1)$
r_2 $fractured(s_1) \Rightarrow damaged(t)$
r_3 $\neg shim(t) \Rightarrow balanced(t)$
r_4 $\neg shim(t) \Rightarrow \neg fractured(s_1)$
r_5 $\neg fractured(s_1), balanced(t) \Rightarrow \neg damaged(t)$
r_6 $varnish(t, 2) \Rightarrow co\_res(t, high)$

r_1 shows that using steel/titanium bolts ($s-t$) will cause microscopic fractures in s_1. r_2 says that if s_1 is fractured, then the whole structure T is damaged. Not using a shim results in a good overall balance of t, as well as in the absence of fractures (r_3, r_4). If the structure is balanced and free of fractures, then it is not damaged (r_5). Finally, applying two layers of varnish results in high resistance to corrosion (r_6).

C. Argumentation Decision Framework ADF

The argumentation formalism presented so far is quite general, but is is sufficient to develop the main contribution of this paper. The next step is to define a framework for modelling decisions.
A decision D is a set of literals. Each of the D_i, together with the knowledge base, forms an argumentation system. This is used to derive arguments about the goals achieved by taking this decision. Goals are represented as literals. To show that a decision has good consequenes, we will construct arguments that have a goal as their claim. We then check which of those arguments are part of an grounded extension.

Definition 8 (ADS)**.** *A tuple $(\mathcal{D}, \mathcal{K}, \mathcal{G})$ is an* argumentation decision system *(ADS) iff*

1) *$\mathcal{D} = \{D_1, \ldots, D_k\}$ and each $D_i \in D$ is a subset of $\mathcal{L}$*
2) *For each $D \in \mathcal{D}$: $(D, \mathcal{K})$ is an argumentation systen*

Table II
AN ARGUMENTATION DECISION MAKING EXAMPLE

Name	*Definition*
D_1	$\{mat(bb, s-t), \neg shim(t), varnish(T, 2)\}$
D_2	$\{mat(bb, s-t), shim(t), varnish(T, 2)\}$
$\mathcal{K}$	$\{r_1, \ldots, r_6\}$
$\mathcal{G}$	$\{\neg damaged(t), co\_res(T, high)\}$

Every ADS defines several argumentation systems, namely one for each of its decisions. The function $Arg(D)$ returns the set of all arguments in the argumentation system $(\mathcal{A}, \mathcal{K})$.

Example 6. *Continuing with the previous example, we can define an ADS $P_M = (\mathcal{D}, \mathcal{K}, \mathcal{G})$ with the decisions explained in Table I.*
For decision D_1, we get the argumentation system $(D_1, \mathcal{K})$. The following are some of the arguments we can form:
$A_1 = \langle \{\neg shim(t)\}, \neg damaged(t)\rangle$.
$A_2 = \langle \{mat(bb, s-t)\}, damaged(t)\rangle$
$A_3 = \langle \{mat(bb, s-t)\}, fractured(s_1)\rangle$.
$A_4 = \langle \{varnish(t, 2)\}, co\_res(t, high)\rangle$.
Argument A_1 rebuts A_2 and vice versa. A_3 undercuts A_1. The argumentation graph for decision D_1 is shown in Figure 2.

A common notion in multi criteria decision making is that of *dominated decisions*. Informally, a decision d is dominated if there is another decision that satisfies the same criteria at least as well as d. It is not reasonable to choose a dominated decision, because there is a better decision that can be chosen without any disadvantages. In order to identify the goals that a decision satisfies, we define a *satisfaction function* s. It returns the goals that can be shown by arguments generated with this decision.

Definition 9 (Satisfaction Function)**.**
Let $S = (\mathcal{D}, \mathcal{K}, \mathcal{G})$ be an ADS. $sat_S : \mathcal{D} \rightarrow \mathcal{P}(\mathcal{G})$ is defined as $sat_S(D) = \{\alpha \in \mathcal{G} \mid \exists\langle \mathcal{A}, \alpha\rangle \in Arg(D)$ such that $\langle \mathcal{A}, \alpha\rangle$ is in the grounded extension of $(\mathcal{A}, \mathcal{K})\}$.

Example 7. *The grounded extension for decision D_1 is $\{A_2, A_3, A_4\}$. Since only the conclusion of A_4 is a goal, we get $sat_S(D_1) = \{co\_res(T, high)\}$.*

It is important to note that we restrict the satisfaction function to arguments in the grounded extension. If, for example, admissible or stable semantics had been used instead, we would face the problem of having to choose one out of a set of extensions. Each of the extensions could contain arguments in favour of the decision, but they cannot all be accepted at the same time. Further information would be required to resolve this conflict.

Definition 10 (Dominated Decision)**.** *Let $S = (\mathcal{D}, \mathcal{K}, \mathcal{G})$ be an ADS. A decision $D \in \mathcal{D}$ is* dominated *iff there is a decision $D' \in \mathcal{D}$ with $D \neq d'$ such that $sat_S(D) \subseteq sat_S(D')$. D is* strictly dominated *if $sat_S(D') \nsubseteq sat_S(D)$ also holds.*

In the above example, D_1 is strictly dominated by D_2.

D. Generating Decisions

ADF prefers decisions which satisfy the most goals, as shown by the number of accepted arguments in favour of each decision, which are in turn built from a fixed knowledge base. This means that it is possible to identify options (i.e. sets of assumptions) that satisfy a maximum number of goals, by using a backward reasoning method. In this section, we formally define such an approach. The method consists of two steps. In the first step, we examine each goal separately and define sets of assumptions that can be used to construct grounded arguments pro. In the second step, we take the possible combinations of assumptions from the first step and choose the one with the best result.

Definition 11 (Backward Argument)**.** *Let $(\mathcal{A}, \mathcal{K})$ be an argumentation system. A* backward argument *for a conclusion $\alpha \in \mathcal{L}$ is a sequence of two sets, $(C_1, B_1), \ldots, (C_m, B_m)$ where $(C_1, B_1) = (\{\alpha\}, \emptyset)$, $B_m \subseteq A$ such that $B_m \vdash \alpha$ and $C_m = \emptyset$. For each (C_i, B_i) with $1 \leq i < m$, the following condition holds:*

1) *There exists a rule $\gamma_1, \ldots, \gamma_n \Rightarrow \beta \in \mathcal{K}$ such that $\beta \in C_i$ and for all γ_i, if $\gamma_i \in \mathcal{A}$ then $\gamma_i \in B_{i+1}$, else $\gamma_i \in C_{i+1}$*

The C_i contain literals that have yet to be proven, and the B_i contain assumptions. If $A = (C_1, B_1), \ldots, (C_m, B_m)$ is a backward argument for α, then $claim(A) = \alpha$ and $support(A) = B_m$.

Example 8. *To make our previous example more detailed, we want to explain why steel/titanium bolts might cause microscopic fractures. The reason is that s_1 is made of composites (comp), which are a relatively brittle material: $brittle(comp)$, $mat(s_1, comp) \Rightarrow brittle(s_1)$; $joined\_with(t, bb)$, $mat(bb, s-t)$, $part\_of(s_1, t)$, $brittle(s_1) \rightarrow dmg(t)$. A backward argument for $dmg(t)$ might consist of the following (C_i, B_i):*

$$
\begin{aligned}
(C_1, B_1) &= (\{dmg(t)\},\ \emptyset)\\
(C_2, B_2) &= (\{brittle(s_1)\},\ \{mat(bb, s-t),\\
&\quad joined\_with(t, bb),\ part\_of(s_1, t)\}\\
(C_3, B_3) &= (\emptyset,\ \{mat(bb, s-t),\ mat(s_1, comp),\\
&\quad joined\_with(t, bb),\ part\_of(s_1, t),\\
&\quad brittle(comp)\})
\end{aligned}
$$

Step by step, the remaining literals in C are removed and replaced according to rules in $\mathcal{K}$. The last set B_3 supports the argument $\langle B_3, dmg(t)\rangle$.

In the context of decision making problems, we are presented with a knowledge base $\mathcal{K}$ and a set of goals $\mathcal{G}$, and it is our aim to generate a set of assumptions. We first define the set of assumptions $\mathcal{A}_{\mathcal{K}}$ that contains all literals which are in the body of at least one rule in $\mathcal{K}$, but not the head of any rule.

Definition 12 (Recommended Decision)**.** *Let $\mathcal{K}$ be a set of rules, $\mathcal{G} \subseteq \mathcal{L}$ a set of literals, and $\mathcal{A}_{\mathcal{K}}$ the set of assumptions as above. A set $D \subseteq \mathcal{A}_{\mathcal{K}}$ is a* recommended decision *iff for all $D' \subseteq \mathcal{A}_{\mathcal{K}}$:*

1) $|sat_S(D')| \leq |sat_S(D)|$ *with* $S = (\{D, D'\}, \mathcal{K}, \mathcal{G})$ *or*
2) $|sat_S(D')| = |sat_S(D)|$ *with* $S = (\{D, D'\}, \mathcal{K}, \mathcal{G})$ *and* $D \subset D'$

Condition 2 assures that every assumption in D is used in at least one argument that is part of a grounded extension of the argumentation system $(\mathcal{A}, \mathcal{K})$. To compute a recommended decision, one could consider all subsets of $\mathcal{A}_{\mathcal{K}}$. However, the effort required for this approach grows exponentially with the size of the input set. We propose to work backwards from the set of goals, generating only the relevant sets of assumptions. Our algorithm works as follows: First, we generate $\mathcal{A}_{\mathcal{K}}$. Then, for each goal $g \in \mathcal{G}$, we generate arguments that support g. Finally, combinations of the generated arguments are evaluated to maximise the set of satisfied goals.

Proposition 1. *Let $\mathcal{A}_{\mathcal{K}}$, $\mathcal{G}$ and $\mathcal{K}$ be defined as above and let $D \subseteq \mathcal{A}_{\mathcal{K}}$ be a recommended decision. Let $P \subseteq Args_S(D)$ with $S = (\{D\}, \mathcal{K}, \mathcal{G})$ such that P contains exactly the arguments in the grounded extension of $(D, \mathcal{K})$. Then, for every argument $\langle \Phi, \alpha \rangle \in P$, there exists a backward argument B_A such that $claim(B_A) = \alpha$ and $support(A) = \Phi$.*

This means that we can find all recommended decisions by evaluating possible combinations of backward arguments for goals and their possible counterarguments.

E. Mutually Exclusive Choices

There is one potential problem, which we will demonstrate in example 9.

Example 9. *We use the knowledge base from the previous examples and the two goals $g_1 : emc(T, high)$ and $g_2 : \neg dmg(t)$. The question we're trying to answer is: which material should the bolts be made of? We add to $\mathcal{K}$ the rules $mat(bb, c)$, $joined\_with(T, bb) \rightarrow \neg dmg(t)$ and the axioms $joined\_with(T, bb)$, $part\_of(c_1, T)$, $mat(c_1, comp)$, assuming that they represent decisions made elsewhere. Using the algorithm as described above, we first compute argument supports for g_1 and get $AM_1 = \{mat(bb, s - t)\}$. For g_2, we get $AM_2 = mat(bb, c)$. Taking $D = AM_1 \cup AM_2$, we get a decision that satisfies all of our goals. However, while both of the arguments pro D are accepted, D is not a valid solution because AM_1 and AM_2 contain two contradicting choices: the material for bb can be either c, or $s - t$, but not both. If the decisions had been modeled manually, D would not have been part of the result, because the mutual exclusion would have been represented implicitly.*

The reason for this behaviour of our system is that some domain knowledge is missing. This problem does not occur if the knowledge base contains certain rules:

Proposition 2. *Let $\mathcal{K}$, $\mathcal{G}$ and $\mathcal{A}_{\mathcal{K}}$ be defined as in definition 12 and let D be a recommended decision. For every pair of literals $a, b \in \mathcal{A}_{\mathcal{K}}$: If there are two rules $a \Rightarrow \neg b$ and $b \Rightarrow \neg a$ in $\mathcal{K}$, then $a \notin D$ or $b \notin D$.*

Proof: Proof by contradiction. Let D be a recommended decision such that a, $b \in D$ and $a \Rightarrow \neg b \in \mathcal{K}$ and $b \Rightarrow \neg a \in \mathcal{K}$. Then, the argument graph defined by the argumentation system $S = (\mathcal{D}, \mathcal{K})$ contains two arguments $A_1 : \langle \{a\}, \neg b \rangle$ and $A_2 : \langle \{b\}, \neg a \rangle$. Since A_1 attacks A_2 and A_2 attacks A_1, the grounded extension cannot contain both A_1 and A_2. Furthermore, A_1 attacks any argument whose support contains b. A_2 attacks any argument whose support contains a. The grounded extension of the argumentation system defined by D can either contain arguments with a in their support, or arguments with b in their support, but not both.

V. The relation of MCDM to ADF

In this section, we show how our system formally corresponds with the class of multi criteria decision making problems characterised by definition 1. Our proof of the correspondence consists of two results: That every ADF system can be expressed as an MCDM system, and conversely that every MCDM system can be expressed as a ADF system. The first result can be found in [4] and we repeat it here for completeness. In order to achieve the second result, we will construct a function that maps multi criteria decision problems to ADF systems. We are going to model MCDM decisions as decisions in an ADS, each containing a single element. The values of the criteria functions are also going to be represented as literals. We then express the results of the aggregation function agg as ADF goals, and finally we will create a rule for each decision that leads to the desired goals. For the second result, we construct an aggregation function based on goals in the ADF system.

Proposition 3 (Equivalent to property 7 in [4])**.** *For every argumentation decision problem $P = (\mathcal{D}, \mathcal{K}, \mathcal{G})$, there exists a multi criteria decision problem $P' = (D, C, agg)$ such that for all $D_i, D_j \in \mathcal{D}$:*

$$sat_P(D_j) \subseteq sat_P(D_i) \Leftrightarrow agg(V_{P'})_j \leq agg(V_{P'})_i$$

This shows that we can construct a mapping from ADF to MCD which preserves the preference relation over decisions. Next, we will show that a similar mapping can be constructed in the other direction, from MCD to ADF, again preserving the preference relation.

Definition 13 (Mapping from MCD to ADF)**.** *Let* $P = (D, C, agg)$ *be a multicriteria decision problem. We construct an argumentation decision problem* $P' = (\mathcal{D}, \mathcal{K}, \mathcal{G})$ *as follows:*

1) $\mathcal{D} = \{\{d_i\} \mid d_i \in D\}$
2) $\mathcal{K} = R_1 \cup R_2 \cup R_3$ *and*
   a) $R_1 = \{d_i \to c_j(v_{i,j}) \mid v_{i,j} \in V_P\}$
   b) $R_2 = \{v_{i,1}, ..., v_{i,k-1}, v_{i,k} \to agg(V_P)_i \mid d_i \in D$ *and* $k = |C|\}$
   c) $R_3 = \{agg(V_P)_i \to agg(V_P)_j \mid agg(V_P)_i \geq agg(V_P)_j\}$
3) $\mathcal{G} = \{agg(V_P)_i \mid d_i \in D\}$

In P', there are no counterarguments. We will now show that every MCD problem can be represented as a ADF problem with the same results, i.e. with the same ranking of decisions. This shows that ADF is at least as expressive as MCD – and it adds benefits such as reusability, accountability and deduction of decisions which will be explained below.

Proposition 4. *Let* $P = (D, C, agg)$ *be a multi criteria decision problem and* $P' = (\mathcal{D}, \mathcal{K}, \mathcal{G})$ *the ADS as constructed according to definition 13. For every decision* $d \in \mathcal{D}$ *and every criterion* $c \in C$*, there exists an argument* $\langle\{d\}, c(d)\rangle$ *in a preferred extension of the argumentation system* $(D, \mathcal{K})$ *with* $D = \{d\}$.

If an ADS is generated from a multi criteria decision making system, then its knowledge base is very simplistic, because it does not contain any domain knowledge. This results from the fact that the domain knowledge which was applied to assign the criteria values for each decision is not represented in the model, and therefore cannot be included in the ADS.

Theorem 1. *For every multi criteria decision problem* $P = (D, C, agg)$*, there exists an ADS* $P' = (\mathcal{D}, \mathcal{K}, \mathcal{G})$ *such that for all* $d_i, d_j \in D$*:*

$$agg(V_P)_j \leq agg(V_P)_i \Leftrightarrow sat_{P'}(D_j) \subseteq sat_{P'}(D_i)$$

The proof relies on the fact that in the mapping constructed according to defintion 13, a decision $d_i \in D$ subsumes all decisions d_k that are worse than d, that is if $agg(V_P)_i \leq agg(V_P)_k$, then the corresponding decision $d \in \mathcal{D}$ satisfies all goals that are satisfied by $d' \in \mathcal{D}$, where d' corresponds to d_k.

Proof: Let $P = (D_{MCD}, C, agg)$ be a multi criteria decision problem and let $P' = (\mathcal{D}_{ARG}, \mathcal{K}, \mathcal{G})$ the ADS as constructed according to definition 13.

(⇒): Let $d_i, d_j \in D_{MCD}$ such that $agg(V_P)_j \leq agg(V_P)_i$. We are going to show that every element $v \in sat_{P'}(D_j)$ is also in $sat_{P'}(D_i)$. Let $v \in sat_{P'}(D_j)$. Then $v \in \mathcal{G}$. By condition 3 of definition 13, $v \in agg(V_P)$, and $v \leq agg(V_P)_j$ (condition 2c of definition 13). Let $v' = agg(V_P)_i$. By condition 3 of definition 13, $v' \in \mathcal{G}$, and by conditions 2a and 2b of definition 13 there is an argument $\langle\{d_i\}, v'\rangle$ in a preferred extension of $(\mathcal{L}, \mathcal{K}, D, \bar{\cdot})$ with $D = \{d_i\}$. By condition 2c of definition 13, $v' \to v \in \mathcal{K}$, so there is an argument $\langle\{d_i\}, v\rangle$ in a preferred extension of $(\mathcal{L}, \mathcal{K}, D, \bar{\cdot})$ with $D = \{d_i\}$. Therefore, $v \in sat_{P'}(D_i)$.

(⇐): Proof by contradiction. Let $d_i, d_j \in D_{MCD}$ such that $agg(V_P)_j \leq agg(V_P)_i$ and $sat_{P'}(D_j) \nsubseteq sat_{P'}(D_i)$. Then there exists an element $e \in sat_{P'}(D_j)$ sucht that $e \notin sat_{P'}(D_i)$. By condition 3 of definition 13, there exists a $k \in \mathbb{N}$ such that $e = agg(V_P)_k$. If $agg(V_P)_k \leq agg(V_P)_i$, then, by condition 2c of definition 13, there is an argument $\langle\{d_i\}, e\rangle$ in a preferred extension of $(\mathcal{L}, \mathcal{K}, D, \bar{\cdot})$ with $D = \{d_i\}$, so $e \in sat_{P'}(D_i)$, which contradicts the assumption. If $agg(V_P)_k > agg(V_P)_i$, then $agg(V_P)_j \geq agg(V_P)_k > agg(V_P)_i$, which also contradicts the assumption.

VI. Evaluation

We are now going to discuss how ADF meets the five requirements that were set in section III.

(1, opaque reasoning process) Preferred decisions in ADF are backed by arguments. Those arguments are part of grounded extensions of an argument graph generated from a knowledge base and are therefore based on a formal model of the domain. This model is part of the ADF, which means outside of that knowledge base, no additional information is needed to reproduce the reasoning process.

(2, local optimum) With the notion of recommended decisions in ADF, a decision maker does not have to enumerate the possible decisions manually. Rather, they are the decisions are defined by the knowledge base. The problem of identifying possible decisions has thus been replaced by the problem of modeling the domain.

(3, lack of documentation standards) Ontologies, which are based on description logics, are a widely used method of recording domain knowledge. Since ADF relies on a formal language with an inference mechanism, it may easily be transformed to an ontology. Conversely, domain knowledge from an ontology can be transformed into ADF rules. Williams and Hunter [8] describe how knowledge from an ontology can be used in the argumentation process.

(4, lack of automated analysis) With the domain knowledge represented as rules used in argumentation, automated decision analsysis is possible as discussed in section IV.

(5, effort of reproducing decision rationale) For ADF systems, the problem of re-evaluating previous decisions depends mainly on the data structure chosen for the knowledge base, because it is sufficient to retrieve the relevant rules and assumptions from previous decisions. It is not necessary to perform the entire analysis again.

In section II-B, we presented decision reusability and claims management as two use cases for documentation in the aerospace industry. Both require a formal, structured documentation of decisions. If an argumentation based system is used for decision making, then the documentation could be generated by the same system with no additional cost, since the arguments used by the system to evaluate decisions are, at the same time, formal justifications of decisions. Because they are structured (as determined by the contents of the knowledge base), they can easily be converted into a structured format for documentation, for example an ontology.

VII. Related Work

Amgoud and Prade [4] propose a decision making framework based on argumentation. Their approach is similar to ours in the way that argumentation is used to generate arguments relating to decisions. There are two kinds of arguments: arguments pro and arguments con. The claims of arguments pro are goals, and the claims of arguments con are negations of goals. The approach we took in ADF differs in that ADF considers only one kind of argument (i.e. arguments *pro*). Arguments against a decision are part of the argument graph and attack arguments pro; thus, we use the argumentation formalism not for generating a list of pros and cons, but for resolving conflicts between arguments related to decisions. We further only take into account the grounded semantics of argument graphs and in ADF, the arguments in favour of each decision are in the same extension. In Amgoud and Prade's system it is possible, for example, to have a graph with multiple extensions that contain arguments in favour of only one decision. In that case, it is not specified which of the extensions will be accepted. Amgoud and Prade's system accounts for levels of uncertainty in the knowledge base and prioritised goals. It also uses a generic logic. Our system is based on a simplified version of ASPIC+ [7], which allows us to generate decisions as explained in section IV-D.

W. Ouerdane et. al. [9] describe how argumentation can be used to partially automate decision making using an interactive dialogue with a user. Their system aims to improve the documentation of decisions. This is done via a dialogue protocol that modifies the knowledge in several iterations. To determine the system's responses in this dialogue, the authors use argument schemes, an approach that captures stereotypical patterns of reasoning [10]. Investigating how argument schemes could be utilised by ADF would be a worthwhile task.

VIII. Conclusion

We presented an argumentation-based system for decision analysis. The key improvements over previous work are the use of grounded extension as acceptability criterion and adding a method of generating decisions. Furthermore, we showed an additional relation to multi criteria decision making. The second contribution of this paper is an analysis of the requirements that engineering companies have for decision analysis and decision documentation systems. We presented a list of five issues of methods that are currently in use by the industry.

A direction for future research is to further validate our approach. We already implemented a software prototype of ADS. An improved version of this software will be used to conduct a user study with mechanical engineers. From this we hope to gain further insights into the design decision making process and how argumentation can be used to assist it.

Acknowledgment

This work is supported by SAP AG and the Invest NI Collaborative Grant for R&D - RD1208002.

References

[1] G. Kou, K. Miettinen, and Y. Shin, "Multiple criteria decision making: Challenges and advancements," *Journal of Multi-Criteria Decision Analysis*, vol. 18, pp. 1–4, 2011.

[2] T. Bench-Capon and P. Dunne, "Argumentation in artificial intelligence," *Artificial Intelligence*, vol. 171, pp. 701–729, 2007.

[3] P. Besnard and A. Hunter, *Elements of Argumentation*, 1st ed. Cambridge, Massachusetts: The MIT Press, 2008.

[4] L. Amgoud and H. Prade, "Using arguments for making and explaining decisions," *Artificial Intelligence*, vol. 173, pp. 413–436, 2009.

[5] S. Staab and R. Studer, Eds., *Handbook on Ontologies*, 1st ed., ser. Internation Handbooks on Information Systems. Springer, 2004.

[6] P. Dung, "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, Sep. 1995.

[7] H. Prakken, "An abstract framework for argumentation with structured arguments," *Argument & Computation*, 2010.

[8] M. Williams and A. Hunter, "Harnessing ontologies for argument-based decision-making in breast cancer," in *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, vol. 2, oct. 2007, pp. 254–261.

[9] W. Ouerdane, Y. Dimopoulos, K. Liapis, and P. Moraitis, "Towards automating decision aiding through argumentation," *Journal of Multi-Criteria Decision Analysis*, vol. 18, pp. 289–309, 2011.

[10] D. Walton, "Justification of argument schemes," *Australasian Journal of Logic*, vol. 3, pp. 1–13, 2005.