An Argumentation-Based Approach for Automatic Evaluation of Design Debates

Pietro Baroni[1], Marco Romano[1], Francesca Toni[2], Marco Aurisicchio[2], and Giorgio Bertanza[1]

[1] Università degli Studi di Brescia, Italy
{pietro.baroni,giorgio.bertanza}@ing.unibs.it, marcojulioromano@gmail.com
http://www.unibs.it
[2] Imperial College London, UK
{m.aurisicchio,f.toni}@imperial.ac.uk
http://www.imperial.ac.uk

Abstract. This paper presents a novel argumentation framework to support design debates in an IBIS-based style, by providing an automatic evaluation of the positions put forwards in the debates. It also describes the integration of the proposed approach within the design-VUE software tool along with two case studies in engineering design and their initial evaluation by domain experts.

Keywords: argumentation, design rationale, IBIS.

Engineering design is often described as an information-processing activity based on problem solving within the constraints of bounded rationality [23,22]. It consists of decomposing an initial problem into a range of sub-problems, proposing and assessing partial solutions, and integrating them in a way that they satisfy the overall problem. This process is collaborative and often involves communication between non co-located engineers. The development and communication of design solutions require engineers to form and share their rationale, i.e. the argumentation in favour or against proposed designs.

These aspects of the engineering design process have led to the development of the *Issue Based Information System* (IBIS) method [16], a graph-based formalisation of the decisions made during a design process along with the reasons why they were made. The IBIS method envisions a decision-making process where problems (or issues) are given solutions (or answers) after a thorough debate involving technical, economical, and ethical considerations. It also provides a means to actively develop, communicate and record the argumentation and reasoning behind the design process.

Initially, IBIS has been conceived purely as conceptual information system and its first implementations were paper-based and totally operated by hand. However, over time several software tools have been developed, which provide a means to edit and visualise IBIS graphs [6,3]. One such tool is designVUE [1], an open-source software developed by the Design Engineering Group of the Mechanical Engineering Department at Imperial College London. These tools, including

J. Leite et al. (Eds.): CLIMA XIV, LNAI 8143, pp. 340–356, 2013.

designVUE, still leave to the users the burden of actually deriving any conclusion from the argumentative process and, eventually, making a decision. This is a task that, depending on the structure of the graph, may not be trivial.

This paper describes the outcome of a collaborative project, involving experts of *engineering design* and *argumentation theory*, undertaken to overcome the limitation of standard design tools in general, and designVUE in particular. The ultimate goal of this project is to support engineers by providing them with an automated evaluation of alternative design solutions, and quickly identifying the most promising answer to a design issue, given the underlying graph structure developed during the design process.

We have singled out argumentation theory as a promising companion to engineering design towards achieving this goal since one of the main features thereof is evaluating arguments' acceptability (e.g. as in [10,9]) or strength (e.g. as in [8,18,17,12]) within debates and dialogues. For this application area, conventional notions of "binary" acceptability (e.g. the notions in [10]), sanctioning arguments as acceptable or not, are better replaced with notions of numerical strength, as the latter are more fine-grained and allow to distinguish different degrees of acceptability.

This paper presents theoretical and practical results from this project. On the theoretical side, we propose a formal method to assign a numerical score to the nodes of an IBIS graph, starting from a base score provided by users. On the practical side, we describe the implementation of this method within designVUE and its preliminary evaluation in the context of two case studies.

The paper is organised as follows. Section 1 gives the basic notions concerning IBIS and the necessary background on argumentation theory. Section 2 introduces a form of argumentation frameworks abstracting away IBIS graphs and Section 3 defines our approach for evalutating quantitatively arguments in these frameworks. Section 4 describes an implementation of our approach as an extension of designVUE and Section 5 illustrates its application in two engineering domains. Section 6 discusses related work and Section 7 concludes.

1 Background

1.1 Issue Based Information System (IBIS)

IBIS [16] is about proposing answers to issues, and assessing them through arguments. At the simplest level, the IBIS method consists of a structure that can be represented as a directed acyclic graph with four types of node: an *issue* node represents a problem being discussed, namely a question in need of an answer; an *answer* node represents a candidate solution to an issue; a *pro-argument* node represents an approval to a given answer or to another argument; a *con-argument* node represents an objection to a given answer or to another argument. An answer node is always linked to an issue node, whereas pro-argument and con-argument nodes are normally linked to answer nodes or to another argument. Each link is directed, pointing towards the dependent node.

Figure 1 shows an example of IBIS graph, with a concrete illustration of the content of the nodes (labelled A1, A2, P1, C1 and C2) in the design domain of Internal Combustion Engines (ICE). This example graph has three layers: the first layer consists of the issue node, the second layer of the two alternative answers, and the third layer of the arguments.

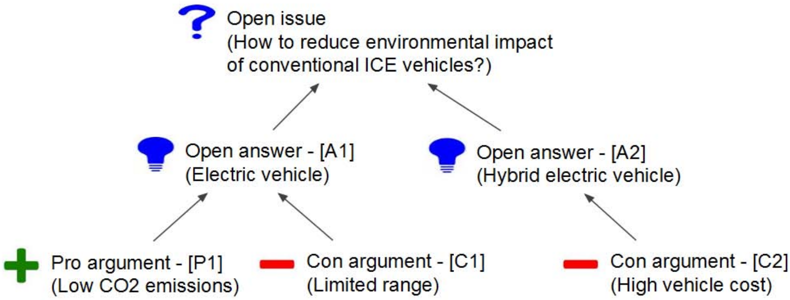

Fig. 1. A simple IBIS graph

An IBIS graph is constructed according to the following rules: (1) an issue is captured; (2) answers are laid out and linked to the issue; (3) arguments are laid out and linked to either the answers or other arguments; (4) further issues may emerge during the process and be linked to either the answers or the arguments.

Conceptually, each addition of an answer or an argument corresponds to a move in the exploration of the design space. In the design domain, IBIS graphs have specific features. First, each IBIS graph concerns a single issue. Second, answers correspond to alternative solutions and compete among them as just one answer can be accepted for an issue.

In some implementations of the IBIS method, the four nodes can have alternative statuses to help users visualise aspects of the decision making process. The precise meaning of these statuses depends on the node type, and is manually assigned by the users. For example, a designer may change the status of an answer from "open" to "accepted", "likely" or "unlikely". In this paper we define a method for automatically, rather than manually, evaluating nodes in (restricted kinds of) IBIS graphs, based on a form of argumentation theory, reviewed next.

1.2 Abstract Argumentation and Argument Valuations

In this work we will make use of Abstract Argumentation [10] and some extensions thereof. We review these briefly here (see the original papers for more details).

Definition 1. *A (finite)* abstract argumentation framework *(*AF*) is a pair* $\langle \mathcal{X}, \mathcal{D} \rangle$, *where* $\mathcal{X}$ *is a finite set of* arguments *and* $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{X}$ *is the* attack *(or* defeat*) relation. A pair* $\langle x, y \rangle \in \mathcal{D}$ *is referred to as 'x is an* attacker *(or* defeater*) of y'.*

An AF can be described as a directed graph whose nodes represent arguments and whose edges represent attacks. The nature and underlying structure of the arguments are completely abstracted away and the focus of the theory is essentially on the management of the conflicts represented by the attack relation. In this context an *(argumentation) semantics* is a criterion to identify the *extensions* of an AF, namely those sets of arguments which can "survive the conflict together". In turn, the *justification status* of an argument, according to a given semantics, can be defined in terms of its membership to the extensions prescribed by the semantics. A variety of semantics have been considered in the literature, whose review is beyond the scope of this paper (see [4] for a survey). The only point we remark is that these semantics evaluate arguments based on a binary notion of membership and thus give rise to a discrete set of justification statuses, which may be appropriate when arguments, e.g., are interpreted as logical sentences in a reasoning process, but may be unsuitable in other contexts.

While AFs are focused on conflicts between arguments, other forms of arguments interaction can be considered, in particular a relation of support, which can be incorporated into AFs to give rise to bipolar AFs [9]:

Definition 2. *A (finite)* bipolar AF *(*BAF*) is a triple $\langle \mathcal{X}, \mathcal{D}, \mathcal{S} \rangle$, where $\langle \mathcal{X}, \mathcal{D} \rangle$ is a (finite)* AF *and $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{X}$ is the* support relation*. A pair $\langle x, y \rangle \in \mathcal{S}$ is referred to as 'x is a* supporter *of y'.*

The discrete argument evaluation for AFs can be extended to BAFs (see [9]).

Another direction of enhancement of AFs amounts to assigning a numerical evaluation to arguments on a continuous scale. We recall here two proposals in this direction. The first gives a notion of local gradual valuation of a BAF, that can be summarised as follows (see [8] for details):

Definition 3. *Let L be a completely ordered set, L^* be the set of all the finite sequences of elements of L (including the empty sequence), and H_{def} and H_{sup} be two ordered sets. Let $\langle \mathcal{X}, \mathcal{D}, \mathcal{S} \rangle$ be a* BAF*. Then, a* local gradual valuation *on $\langle \mathcal{X}, \mathcal{D}, \mathcal{S} \rangle$ is a function $v : \mathcal{X} \to L$ such that, for a generic argument $a \in \mathcal{X}$, given $\mathcal{D}^-(a) = \{d_1, \ldots, d_n\}$ the set of attackers of a and $\mathcal{S}^-(a) = \{s_1, \ldots, s_p\}$ the set of supporters of a (for $n, p \geq 0$):*

$$v(a) = g(h_{sup}(v(s_1), \ldots, v(s_p)), h_{def}(v(d_1), \ldots, v(d_n)))$$

where $g : H_{sup} \times H_{def} \to L$ is a function with $g(x, y)$ increasing on x and decreasing on y, and $h_{def} : L^* \to H_{def}/h_{sup} : L^* \to H_{sup}$ are functions (valuing the quality of the defeat/support, respectively) satisfying for any $x_1, \ldots, x_n, x_{n+1}$ (here $h = h_{def}$ or h_{sup}): (i) if $x_i \geq x_{i'}$ then $h(x_1, \ldots, x_i, \ldots, x_n) \geq h(x_1, \ldots, x_{i'}, \ldots, x_n)$; (ii) $h(x_1, \ldots, x_n) \leq h(x_1, \ldots, x_n, x_{n+1})$; (iii) $h() \leq h(x_1, \ldots, x_n)$; (iv) $h(x_1, \ldots, x_n)$ is bounded by a limit value β.

Note that the local gradual valuation (LGV in the remainder) of an argument is defined recursively in terms of the valuations of its attackers and supporters.

The second proposal we consider is the Extended Social Abstract Argumentation approach of [12], taking into account, in addition to attackers and supporters, also positive or negative votes on arguments. In a nutshell, the idea is that in a social context (like an Internet-based social network or debate) opinions (arguments) are evaluated by a community of users through a voting process.

Definition 4. *An* Extended Social Abstract Argumentation Framework *(*ESAAF*) is a 4-tuple* $\langle \mathcal{X}, \mathcal{D}, \mathcal{S}, \mathcal{V} \rangle$ *where* $\langle \mathcal{X}, \mathcal{D}, \mathcal{S} \rangle$ *is a (finite)* BAF *and* $\mathcal{V} : \mathcal{X} \rightarrow \mathbb{N} \times \mathbb{N}$ *is a function mapping arguments to the number of their positive and negative votes.*

Given an (acyclic) ESAAF, argument evaluation is based on votes and on the attack/support relations. It involves a set of operators (called Semantic Framework) extending the operators of [17], where only attackers were considered:

Definition 5. *A* semantic framework *is a 7-tuple* $\langle L, \tau, \wedge, \vee, \neg, \odot, \uplus \rangle$ *where* L *is a completely ordered set,* $\tau : \mathbb{N} \times \mathbb{N} \rightarrow L$, $\wedge : L \times L \rightarrow L$, $\vee : L \times L \rightarrow L$, $\neg : L \rightarrow L$, $\odot : L \times L \rightarrow L$, $\uplus : L \times L \rightarrow L$. *Given an* ESAAF $\langle \mathcal{X}, \mathcal{D}, \mathcal{S}, \mathcal{V} \rangle$ *and a semantic framework* $\langle L, \tau, \wedge, \vee, \neg, \odot, \uplus \rangle$*, the* valuation of *argument* $a \in \mathcal{X}$ *is:*

$$M^{+}(a) = (\tau(a) \wedge \neg \vee \{M^{+}a_i : (a_i, a) \in \mathcal{D}\}) \uplus (\tau(a) \odot \vee \{M^{+}a_i : (a_i, a) \in \mathcal{S}\})$$

Omitting details, informally, the operator τ evaluates the social support for each argument a, based on its accumulated positive and negative votes (given by $\mathcal{V}$), and so assigns an *initial score*, $\tau(a)$, to a. This initial score has no counterpart in LGV seen earlier. Then, as in the case of LGV, the valuation of a is defined recursively in terms of the valuations of its attackers and supporters. The individual valuations of the attackers and of the supporters of a are first aggregated using the $\vee$ operator. Then the aggregated valuations of the attackers and supporters are combined with $\tau(a)$ using the $\wedge$ and $\neg$ operators and the $\odot$ operator respectively. This results in a pair of values which roughly corresponds to the pair $h_{sup}(v(s_1), \ldots, v(s_p)), h_{def}(v(d_1), \ldots, v(d_n))$ in LGV, the main difference being the fact that $\tau(a)$ can be regarded as an additional parameter of these functions. Finally, the $\uplus$ operator maps the above pair of values in a single final evaluation (and so clearly corresponds to the function g in LGV).

2 Quantitative Argumentation Debate Frameworks

In section 1.1 we have seen that design scenarios require IBIS graphs with specific features, and in particular with a single specific (design) issue and answers (linking to that issue) corresponding to different alternative solutions. Whereas IBIS graphs (in general and in design contexts) allow new issues to be brought up during the argumentation, in this paper for simplicity we will disallow this possibility, and focus on design debates that can be represented by IBIS graphs where arguments can only be pointed to by other arguments, although argument nodes may have other argument nodes as children, recursively.

We will define, in Section 3, a method for evaluating arguments and answers in IBIS graphs, and accompanying or replacing the manual evaluation available in some IBIS implementations (see Section 1.1). Examining some design scenarios with the relevant experts (see also Section 5) it emerged that, in their valuations, they typically ascribe different importance to pro- and con-arguments, which entails that a *base score* is required as a starting point for the evaluation. In order to fulfil these requirements, we propose a formal framework as follows:

Definition 6. *A* QuAD (Quantitative Argumentation Debate) framework *is a 5-tuple* $\langle \mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{R}, \mathcal{BS} \rangle$ *such that (for* scale $\mathbb{I}=[0,1]$*):*

$\mathcal{A}$ *is a finite set of* answer arguments*;*
$\mathcal{C}$ *is a finite set of* con-arguments*;*
$\mathcal{P}$ *is a finite set of* pro-arguments*;*
the sets $\mathcal{A}$*,* $\mathcal{C}$*, and* $\mathcal{P}$ *are pairwise disjoint;*
$\mathcal{R} \subseteq (\mathcal{C} \cup \mathcal{P}) \times (\mathcal{A} \cup \mathcal{C} \cup \mathcal{P})$ *is an acyclic binary relation;*
$\mathcal{BS} : (\mathcal{A} \cup \mathcal{C} \cup \mathcal{P}) \to \mathbb{I}$ *is a total function;* $\mathcal{BS}(a)$ *is the* base score *of* a*.*

The framework is referred to as "quantitative" due to the presence of the base score. Ignoring this base score, clearly QuAD graphs are abstractions of (restricted forms of) IBIS graphs, with the issue node omitted since QuAD frameworks are focused on the evaluation of answer nodes for a specific issue. For example, the QuAD graph representation of the IBIS graph in Figure 1 has $\mathcal{A} = \{A1, A2\}$, $\mathcal{C} = \{C1, C2\}$, $\mathcal{P} = \{P1\}$ and $\mathcal{R} = \{(P1, A1), (C1, A1), (C2, A2)\}$.

It is easy to see that a QuAD framework can also be interpreted as a BAF (again ignoring the base score), as notions of attack and support are embedded in the disjoint sets $\mathcal{C}$ and $\mathcal{P}$. This is made explicit by the following definition.

Definition 7. *Let* $\mathcal{F} = \langle \mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{R}, \mathcal{BS} \rangle$ *be a QuAD framework and let* $a \in (\mathcal{A} \cup \mathcal{C} \cup \mathcal{P})$*. The set of* direct attackers of *a is defined as* $\mathcal{R}^-(a) = \{b \in \mathcal{C} : (b, a) \in \mathcal{R}\}$*. The set of* direct supporters of *a is defined as* $\mathcal{R}^+(a) = \{b \in \mathcal{P} : (b, a) \in \mathcal{R}\}$*. Then, the* BAF corresponding to $\mathcal{F}$ *is* $\langle \mathcal{X}, \mathcal{D}, \mathcal{S} \rangle$ *such that:*
$\mathcal{X} = \mathcal{A} \cup \mathcal{C} \cup \mathcal{P}$, $\mathcal{D} = \{(b, a) | b \in \mathcal{R}^-(a), a \in \mathcal{X}\}$, $\mathcal{S} = \{(b, a) | b \in \mathcal{R}^+(a), a \in \mathcal{X}\}$.

Note that an ESAAF equipped with a semantic framework can give rise to a QuAD framework, with the base score in the QuAD framework given by the initial score τ in the semantic framework for the ESAAF. The semantic framework includes however a recipe for calculating the initial score of arguments, based on votes in the ESAAF, whereas our QuAD framework assumes that the base score is given. Indeed, differently from the application contexts envisaged for ESAAF, design debates do not involve large community of users so the notion of a base score based on votes is not appropriate, rather the base score can be represented as a numerical value directly assessed by experts.

3 Automatic Evaluation in QuAD Frameworks

Given a QuAD framework, in order to support the decision making process by design engineers we need a method to assign a quantitative evaluation, called *final score*, to answer nodes. To this purpose we investigate the definition of a *score function* $\mathcal{SF}$ for arguments of a QuAD framework. The basic idea is that the final score of an argument depends on its base score and on the final scores of its attackers and supporters, so $\mathcal{SF}$ is defined recursively using a *score operator* able to combine these three elements. For a generic argument a, let $(a_1, \ldots, a_n)$ be an arbitrary permutation of the ($n \geq 0$) attackers in $\mathcal{R}^-(a)$. We denote as $\mathcal{SC}(\mathcal{R}^-(a)) = (\mathcal{SF}(a_1), \ldots, \mathcal{SF}(a_n))$ the corresponding sequence of final scores. Similarly, letting $(b_1, \ldots, b_m)$ be an arbitrary permutation of the $(m \geq 0)$ supporters in $\mathcal{R}^+(a)$, we denote as $\mathcal{SC}(\mathcal{R}^+(a)) = (\mathcal{SF}(b_1), \ldots, \mathcal{SF}(b_m))$ the corresponding sequence of final scores. Then, using the hypothesis (implicitly adopted both in [8] and [12]) of separability of the evaluations concerning attackers and supporters,[1] a generic score function for an argument a can be defined as:

$$\mathcal{SF}(a) = g(\mathcal{BS}(a), \mathcal{F}_{att}(\mathcal{BS}(a), \mathcal{SC}(\mathcal{R}^-(a))), \mathcal{F}_{supp}(\mathcal{BS}(a), \mathcal{SC}(\mathcal{R}^+(a)))) \quad (1)$$

Referring to the example of Figure 1, suppose that $\mathcal{BS}(A1) = \mathcal{BS}(A2) = 0.5$, $\mathcal{BS}(C1) = 0.7$, $\mathcal{BS}(C2) = 0.4$, $\mathcal{BS}(P1) = 0.9$. Then, denoting the empty sequence as $()$, we obtain

$$\mathcal{SF}(A1) = g(0.5, \mathcal{F}_{att}(0.5, \mathcal{SC}((C1))), \mathcal{F}_{supp}(0.5, \mathcal{SC}((P1))));$$
$$\mathcal{SF}(A2) = g(0.5, \mathcal{F}_{att}(0.5, \mathcal{SC}((C2))), \mathcal{F}_{supp}(0.5, ()));$$
$$\mathcal{SF}(C1) = g(0.7, (), ()); \quad \mathcal{SF}(C2) = g(0.4, (), ()); \quad \mathcal{SF}(P1) = g(0.9, (), ()).$$

We identify some basic requirements for the score function. First, if there are neither attackers nor supporters for an argument then its final evaluation must coincide with the base score (in our running example this applies to arguments $C1$, $C2$, and $P1$). For any $v_0 \in \mathbb{I}$, this requirement can be expressed as

$$g(v_0, (), ()) = v_0. \quad (2)$$

Moreover, each attacker (supporter) should have a negative or null (positive or null, respectively) effect on the final scores. Given a generic sequence $S = (s_1, \ldots, s_k) \in \mathbb{I}^k$ and $v \in \mathbb{I}$, let us denote as $S \cup (v)$ the sequence $(s_1, \ldots, s_k, v) \in \mathbb{I}^{k+1}$. The above requirements can then be expressed, for sequences S_1, S_2, as

$$g(v_0, \mathcal{F}_{att}(S_1), \mathcal{F}_{supp}(S_2)) \geq g(v_0, \mathcal{F}_{att}(S_1 \cup (v)), \mathcal{F}_{supp}(S_2)) \quad (3)$$
$$g(v_0, \mathcal{F}_{att}(S_1), \mathcal{F}_{supp}(S_2)) \leq g(v_0, \mathcal{F}_{att}(S_1), \mathcal{F}_{supp}(S_2 \cup (v))) \quad (4)$$

We define $\mathcal{F}_{att}$ (and dually $\mathcal{F}_{supp}$) so that the contribution of an attacker (supporter) to the score of an argument decreases (increases) the argument score by an amount proportional both to (i) the score of the attacker (supporter), i.e. a strong attacker (supporter) has more effect than a weaker one, and (ii) to the previous score of the argument itself, i.e. an already strong argument benefits quantitatively less from a support than a weak one and an already weak argument suffers quantitatively less from an attack than a stronger one. Focusing on the case of a single attacker (supporter) with score v this leads to the following base expressions:[2]

$$f_{att}(v_0, v) = v_0 - v_0 \cdot v = v_0 \cdot (1 - v) \quad (5)$$
$$f_{supp}(v_0, v) = v_0 + (1 - v_0) \cdot v = v_0 + v - v_0 \cdot v \quad (6)$$

1 Here, separability amounts to absence of interaction between attackers and supporters.

2 The expression of f_{supp} corresponds to the T-conorm operator also referred to as *probabilistic sum* in the literature [15].

The definitions of $\mathcal{F}_{att}$ and $\mathcal{F}_{supp}$ have then the same recursive form. Let $*$ stand for either *att* or *supp*. Then:

$$\mathcal{F}_*(v_0, ()) = v_0 \tag{7}$$

$$\mathcal{F}_*(v_0, (v)) = f_*(v_0, v) \tag{8}$$

$$\mathcal{F}_*(v_0, (v_1, \ldots, v_n)) = f_*(\mathcal{F}_*(v_0, (v_1, \ldots, v_{n-1})), v_n) \tag{9}$$

Note that this definition directly entails that $\mathcal{F}_{att}(v_0, S) \geq \mathcal{F}_{att}(v_0, S \cup (v))$ and $\mathcal{F}_{supp}(v_0, S) \leq \mathcal{F}_{supp}(v_0, S \cup (v))$. In our running example, we get

$\mathcal{F}_{att}(0.5, \mathcal{SC}((C1))) = \mathcal{F}_{att}(0.5, (0.7)) = f_{att}(0.5, 0.7) = 0.15$,
$\mathcal{F}_{supp}(0.5, \mathcal{SC}((P1))) = \mathcal{F}_{supp}(0.5, (0.9)) = f_{supp}(0.5, 0.9) = 0.95$,
$\mathcal{F}_{att}(0.5, \mathcal{SC}((C2))) = \mathcal{F}_{att}(0.5, (0.4)) = f_{att}(0.5, 0.4) = 0.3$, and
$\mathcal{F}_{supp}(0.5, ()) = 0.5$.

We now establish some basic properties of $\mathcal{F}_{att}$ and $\mathcal{F}_{supp}$. First, they return values in $\mathbb{I} = [0, 1]$, as required:

Proposition 1. *For any $v_0 \in \mathbb{I}$ and for any sequence $(v_1, \ldots, v_k) \in \mathbb{I}^k$, $k \geq 0$, $\mathcal{F}_{att}(v_0, (v_1, \ldots, v_k)) \in \mathbb{I}$ and $\mathcal{F}_{supp}(v_0, (v_1, \ldots, v_k)) \in \mathbb{I}$.*

Proof. By induction on k. For the base case, trivially the statement holds for $k = 0$ (empty sequence) and $k = 1$ given the definitions of f_{att} and f_{supp}. Assume that the statement holds for a generic sequence of length $k - 1$, i.e. $\mathcal{F}_{att}(v_0, (v_1, \ldots, v_{k-1})) = v_x \in \mathbb{I}$ then, from (9), $\mathcal{F}_{att}(v_0, (v_1, \ldots, v_k)) = f_{att}(v_x, v_k)$. Similarly, letting $\mathcal{F}_{supp}(v_0, (v_1, \ldots, v_{k-1})) = v_y \in \mathbb{I}$ we get $\mathcal{F}_{supp}(v_0, (v_1, \ldots, v_k)) = f_{supp}(v_y, v_k)$. Then, again the statement holds by definition of f_{att} and f_{supp}.

Then, it is of course required that $\mathcal{F}_{att}$ and $\mathcal{F}_{supp}$ produce the same result for any permutation of the same sequence.

Proposition 2. *For any $v_0 \in \mathbb{I}$ and $(v_1, \ldots, v_k) \in \mathbb{I}^k$, $k \geq 0$, let $(v_{1_i}, \ldots, v_{k_i})$ be an arbitrary permutation of $(v_1, \ldots, v_k)$. It holds that $\mathcal{F}_{att}(v_0, (v_1, \ldots, v_k)) = \mathcal{F}_{att}(v_0, (v_{1_i}, \ldots, v_{k_i}))$ and $\mathcal{F}_{supp}(v_0, (v_1, \ldots, v_k)) = \mathcal{F}_{supp}(v_0, (v_{1_i}, \ldots, v_{k_i}))$.*

Proof. $\mathcal{F}_{att}(v_0, (v_1, \ldots, v_k)) = f_{att}(f_{att}(\ldots f_{att}(v_0, v_1) \ldots), v_{k-1}), v_k) = (((v_0 \cdot (1 - v_1)) \cdot (1 - v_2)) \ldots \cdot (1 - v_k)) = v_0 \cdot \prod_{i=1}^{k}(1 - v_i)$. Thus the statement follows directly from commutativity and associativity of the product of the $(1 - v_i)$ factors. As to $\mathcal{F}_{supp}$, $\mathcal{F}_{supp}(v_0, (v_1, \ldots, v_k)) = f_{supp}(f_{supp}(\ldots f_{supp}(v_0, v_1) \ldots), v_{k-1}), v_k)$, the statement follows from the well-known properties of commutativity and associativity of any T-conorm.

Another desirable property of $\mathcal{F}_{att}$ and $\mathcal{F}_{supp}$ is a sort of monotonic behavior with respect to the increasing score of attackers and supporters respectively.

Proposition 3. *For any $v_0 \in \mathbb{I}$ and for any $S = (v_1, \ldots, v_h, \ldots, v_k) \in \mathbb{I}^k$, $k \geq 1$, $1 \leq h \leq k$, let S^+ be a sequence obtained from S by replacing v_h with some $v_l > v_h$. Then $\mathcal{F}_{att}(v_0, S) \geq \mathcal{F}_{att}(v_0, S^+)$ and $\mathcal{F}_{supp}(v_0, S) \leq \mathcal{F}_{supp}(v_0, S^+)$.*

Proof. As to $\mathcal{F}_{att}$ given that for a generic sequence $\mathcal{F}_{att}(v_0, (v_1, \ldots, v_k)) = v_0 \cdot \prod_{i=1}^{k}(1 - v_i)$, we observe that $\mathcal{F}_{att}(v_0, S^+) = \mathcal{F}_{att}(v_0, S) \cdot \frac{1-v_l}{1-v_h}$ and the statement follows from $0 \leq 1 - v_l < 1 - v_h$. As to $\mathcal{F}_{supp}$, from commutativity and associativity of f_{supp}, letting $S^* = (v_1, \ldots, v_{h-1}, v_{h+1}, \ldots, v_k) \in \mathbb{I}^{k-1}$, we get $\mathcal{F}_{supp}(v_0, S) = f_{supp}(\mathcal{F}_{supp}(v_0, S^*), v_h)$ and $\mathcal{F}_{supp}(v_0, S^+) = f_{supp}(\mathcal{F}_{supp}(v_0, S^*), v_l)$ and the statement follows from the well-known monotonicity of T-conorms.

In order to finalise the definition of score function we need to define g. For this we adopted the idea that when the effect of attackers is null (i.e. the base score is left unchanged as far as attackers are concerned) the final score must coincide with the one established on the basis of supporters, and dually when the effect of supporters is null. Clearly, when both are null the final score must coincide with the base score. When both attackers and supporters have an effect, the final score is obtained averaging the two contributions. Formally:

Definition 8. *The operator $g : \mathbb{I} \times \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ is defined as follows:*

$$g(v_0, v_a, v_s) = v_a \textit{ if } v_s = v_0 \quad (10)$$

$$g(v_0, v_a, v_s) = v_s \textit{ if } v_a = v_0 \quad (11)$$

$$g(v_0, v_a, v_s) = \frac{(v_a + v_s)}{2} \textit{ otherwise} \quad (12)$$

Then, the following result directly follows from Propositions 1–3:

Proposition 4. *The score function $\mathcal{SF}(a)$ defined by equations (1), (7), (8) and (9) and by Definition 8 satifies properties (2), (3), and (4).*

For our running example, we get $\mathcal{SF}(A1) = g(0.5, 0.15, 0.95) = 0.55$ and $\mathcal{SF}(A2) = g(0.5, 0.3, 0.5) = 0.3$.

Note that, by definition of our operator g, the addition of an attack (support) for an argument previously not attacked (supported, respectively) gives rise to a discontinuity. This in a sense reflects a discontinuity in the underlying debate. Whether this behaviour is suitable in all contexts is an open question, and the definition of different forms of $\mathcal{SF}$ without this discontinuity is an importnat direction for future work.

On the computational side, given that in a QuAD framework the relation $\mathcal{R}$ is acyclic, evaluating $\mathcal{SF}$ for answers nodes (in fact, for any node) is quite easy: given an argument a to be evaluated the score function is invoked recursively on its attackers and supporters to obtain $\mathcal{SC}(\mathcal{R}^-(a))$ and $\mathcal{SC}(\mathcal{R}^+(a))$ which are finally fed to the $\mathcal{SF}$ operator along with the base score $\mathcal{BS}(a)$. The recursion is well-founded given the acyclicity of $\mathcal{R}$, the base being provided by nodes with neither attackers nor supporters whose final score coincides with their base score.

4 Implementation in designVUE

The proposed approach has been implemented within a pre-existing IBIS application known as *design Visual Understanding Environment* (designVUE) [1].

designVUE has been chosen as a platform for the implementation of the proposed approach for various reasons: it is open-source; it has been developed by the Design Engineering Group at Imperial College London; it is receiving increasing interest from academia and industry and as a result has a growing user community. In the following paragraphs we describe in more detail designVUE and its extension with the QuAD framework.

designVUE is an application developed using Java to attain cross-platform portability. Its GUI consists primarily of a main window, which contains the menu bar, the toolbar and the graph canvas.

The main purpose of designVUE is to draw graphs (also referred to as diagrams and maps) mostly consisting of nodes (depicted as boxes) and links (depicted as arrows) among them. The programme does not impose any restriction on the way a graph can be drawn. It is up to the user to confer any meaning to a graph. Among the large variety of graphs that can be drawn, designVUE supports IBIS graphs. These have no special treatment in designVUE and, in particular, there is no support to the evaluation of the argumentative process. In addition to the main window, there are floating windows that can be opened from the *Windows* menu. One of these, called Info Window, presents information about the currently selected node.

The QuAD framework has been implemented in Java and integrated into a customised version of designVUE, forking its existing codebase. The additions and modifications brought to designVUE fit broadly in two categories: those related to the GUI; and those concerning the implementation of the score assignment method. As for the GUI:

- a new pane called *BaseScore Pane* has been added to the *Info Window*: it displays the base score of the currently selected IBIS node and allows the user to edit it (base scores are created with a default value of 0.5);
- a new pane called *Score Pane* has been added to *Info Window*: it displays the final score of the currently selected IBIS node;
- a new menu item labeled *Compute Argumentation on IBIS node* has been added to the *Content* menu: it can be invoked only after selecting an IBIS answer node and triggers the score computation for the selected node (and for all the nodes on which it depends).

As to the algorithm to compute final scores, it has been implemented in a Java class, which basically carries out a depth-first post-order traversal, which acts directly onto the IBIS nodes displayed in the canvas. To enhance performances in complex graphs where some pro and/or con arguments affect many other arguments, the algorithm implements a so-called closed list in order to reuse the scores already computed in previous phases of the graph traversal.

5 Case Studies

The enhanced version of designVUE was evaluated through two case studies. The first, in the domain of civil engineering, concerns the choice of foundations for a multi-storey building to be developed on a brownfield. The second, in the domain of water engineering, focuses on the choice of a reuse technology for sludge produced by wastewater treatment plants.

The first case study was developed in collaboration with a civil engineer with more than ten years of experience in the industry, who was already familiar with the IBIS concept having used it through the Compendium software [6]. Differently, the second case study was developed together with an expert at the University of Brescia, who had neither previous knowledge of the IBIS concept, nor of any tool implementing it.

5.1 Foundations

This case study is based on a design task, which was selected to satisfy the following criteria: the design problem had to be well known to the industry; and the problem solving process had to rely on the application of known and established solution principles. On this basis the task presented in this case study can be considered to be at the boundary between adaptive and variant design [19]. The reason for choosing this type of design task is to adopt a *walk before you run* approach to evaluation.

The case is based on real project experience of the collaborating engineer. However, it was not developed during the actual design process but rather reconstructed retrospectively. Prior to the development of the case, the engineer was introduced to the enhanced version of designVUE and instructed to use it including inputting values for the base scores.

As mentioned earlier, the design problem focuses on the selection of the most appropriate type of foundation for a multi-storey building in a brownfield area. This is the part of urban planning concerning the re-use of abandoned or underused industrial and commercial facilities. When considering the choice of building foundations in brownfield sites, multiple alternatives are common and multiple considerations have to be made starting from the different kinds of ground and their load bearing capabilities, which are usually different than in greenfield sites.

The starting point of the IBIS graph developed by the engineer is the issue to choose a suitable foundation given the requirements discussed earlier (see Figure 2). Three types of foundation solutions are considered, namely Pad, Raft and Piles, and these are subsequently evaluated using several pro- and con-arguments. After the development of the IBIS graph the engineer executed the score computation on the three solutions under two situations: 1) using default values for the base scores; and 2) using modified values for the base scores. The modified values for the base scores emerged through a three step process involving extraction of the criteria behind each argument (see text in bracket at the bottom of each argument in Figure 2), analysis of the relative importance of the criteria in the context of the selected design task, and assignment of a numerical value between 0 and 1 to each criteria.

The results for the situation with unchanged values indicate that Pad (0.51) is the preferred solution over Raft (0.49) and Piles (0.44). Differently, the results for the situation in which the values were changed suggest that Piles (0.56) is sligthly preferable to Raft (0.55) and considerably preferable to Pad (0.41). As it can be seen, the three alternatives are ranked exactly in the reverse order. Only the results based on the modified values for the base scores were judged by the expert consistent with his conclusions.

On one hand this confirms the importance of weighting pro- and con-arguments with expert-provided base scores in order to get meaningful results. On the other hand, it shows that a purely graphical representation of the pros and cons is typically insufficient to give an account of the reasons underlying the final choice by the experts. In this sense, representing and managing explicitly quantitative valuations enhances transparency and accountability of the decision process.

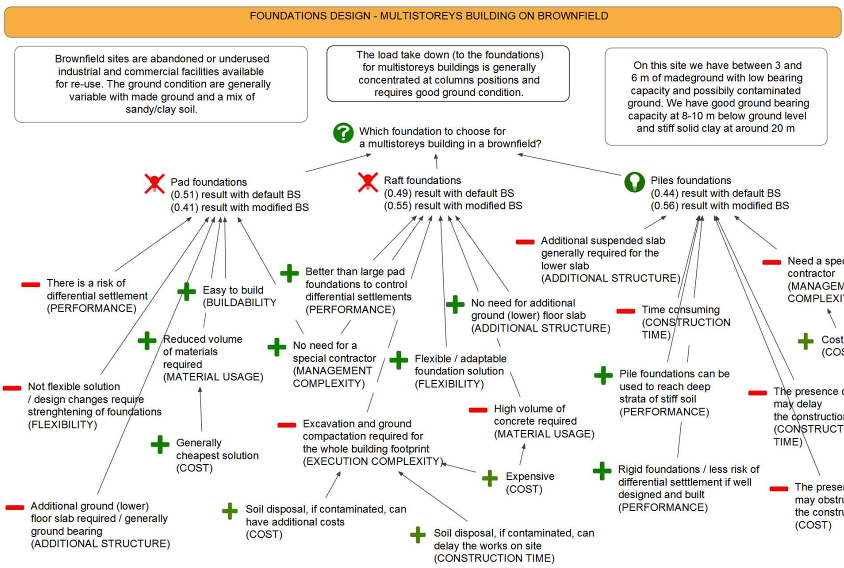

Fig. 2. designVUE graph of the foundation project debate. Note that in designVUE the answer node has multiple statuses. In agreement with the automatic evaluation, the status for the Pad and Raft foundation answers has been manually changed to 'rejected' (red crossed out light bulb icon), while that for the Piles foundation answer to 'accepted' (green light bulb icon).

5.2 Sludge Reuse

Sewage sludge is produced from the treatment of wastewater. Its traditional reuse option (alternative to landfill disposal) had been land application (due to its content of organic carbon and nutrients). Actually, reuse in agriculture is subject to restrictions (since the sludge also contains pollutants), so that other disposal routes, such as wet oxidation, reuse in the cement industry or energy recovery by combustion are considered as viable alternatives. The choice of the best alternative depends on technical (feasibility, applicability, reliability), economic, environmental and social factors whose importance varies from site to site. In this context, the use of the enhanced version of designVUE was proposed to an environmental engineering expert, who had no previous experience with any IBIS support tool.

As a first step, the expert provided a qualitative valuation scheme in tabular form that has been translated into a designVUE graph. Then the expert was asked to assign weights to the pro and con arguments associated with the different options and to compare the system's evaluation of the alternatives with his own one. As for the first request, the expert was able to assign weights to the pro and con arguments associated with each technology without particular problems. As to the second request, he observed that in this context technical experts are not in charge of the final decision since environment related projects are subjected to the approval of public officers or committees, who, taking into account context-specific aspects (e.g. social issues), may ascribe different importance to the technical considerations formulated by the expert. To properly represent this two-phase decision process within designVUE the expert suggested the use of a graph with a characteristic 2-tier structure (see figure 3), where:

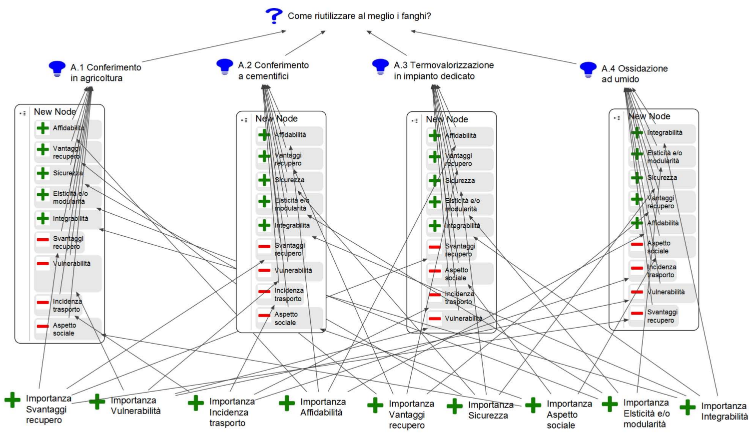

Fig. 3. designVUE graph for the second phase of the sludge reuse project debate. Note that the four answers are in the 'open' status (blue light bulb icon) as a decision has not been made yet.

– the first tier takes into account the technical strengths and weaknesses of every single alternative. These are the pro and con arguments directly linked with the answers, whose base scores are provided by domain experts.
– The second tier involves the final decision-makers and consists of pro arguments attached to the pros and cons expressed by the expert. By assigning the base scores to the arguments of the second tier, the decision-makers

modulate the actual influence of first tier arguments according to context specific considerations. The structure of the 2-tier graph is defined so as to ensure that the same factor gets the same weight in the assessment of all alternatives.

Following this line, designVUE can be used to support a multistep methodology taking explicitly into account different classes of stakeholders. While the study of this methodology is left to future work, the expert expressed a positive judgment about the tool, with particular appreciation for the intuitive visual representation and the traceability of the reasons underlying the final decisions.

6 Related Work

In engineering design, various methods are used to support the evaluation of design alternatives, e.g. decision-matrix [20] and analytic hierarchy process [21]. Among these, the decision-matrix, also known as Pugh method, is the simplest and most commonly adopted. It consists of ranking alternatives by identifying a set of evaluation criteria, weighting their importance, scoring the alternatives against each criteria, multiplying the scores by the weight, and computing the total score for each alternative. Our work differs from the Pugh method in that it aims to extract a quantitative evaluation of alternatives from rich and explicitly captured argumentation rather than systematically assigned and justified scores. Hence, it seems to have the potential to lead to more logically reasoned decisions.

Turning to argumentation literature, the idea of providing a quantitative evaluation of a given position on the basis of arguments in favor and against has been considered in several works.

In [5], in the context of a logic-based approach to argumentation, an argument structure for a logical formula α is (omitting some details) a collection of reasons supporting $(\neg)\alpha$. Each reason is represented as an argument tree, whose root is an argument for $(\neg)\alpha$ and where the children of an argument node are attackers of the node itself. Each argument tree is quantitatively evaluated using a *categoriser*. The results of the evaluation of argument trees for $(\neg)\alpha$ are aggregated separately using an *accumulator* function and then combined. Though this work shows several similarities with our approach at a generic level, we point out some important differences. In [5] the evaluation concerns logical formulas rather than arguments, arguments can only attack (not support) each other, while the notion of support for a formula coincides with the (defeasible) derivation of the formula. Then, differently from our approach, the recursive procedure corresponding to the categoriser concerns attacks only and the notion of support plays a role only in the accumulator. Also, in [5] there is no notion of base score.

The gradual valuation of BAFs [8] (see Section 1.2) is closer to our proposal. In fact, the generic valuation function v of BAFs (see Definition 3) has a similar structure to our $\mathcal{SF}$, with h_{sup}, h_{def} corresponding to our $\mathcal{SC}(\mathcal{R}^+(a))$, $\mathcal{SC}(\mathcal{R}^-(a))$ respectively and satisfying analogous properties. A basic difference concerns the base score, absent in [8] and crucial in our application domain.

The ESAAF approach of [12] (see Section 1.2) has more similarities, as it encompasses an initial score for arguments (obtained from votes) and a recursive evaluation mechanism similar to ours. In fact, the treatment we propose for attackers coincides with the one proposed in [12], while our proposal differs in the treatment of supporters: in [12] supporters are treated as a sort of "negative attacks", while in our approach supporters contribute to increase the base score specularly to the way attackers contribute to decrease it. As a consequence, in ESAAF the operator $\odot$ for the combination of the initial score with the aggregation of supporters' valuations includes the min operator to prevent that the combination exceeds the limit value of 1. This means that the contribution of supporters is subject to a saturation which may be undesirable in some cases.

The approach of [13] also features significant similarities with our proposal. In fact the notion of *real equational network* introduced in [13] uses an evaluation function $f(a)$ from the set of arguments to $[0, 1]$ which is defined recursively, for an argument a, as $f(a) = h_a(f(a_1), \ldots, f(a_k))$ where $a_1, \ldots, a_k$ are the attackers of a. [13] explores several alternatives for the function f with unrestricted graph topology (in the presence of cycles the solution is a fixed point of f) but no notion of base argument score is considered. Note that, assuming a fixed initial score of 1 for any argument, our $\mathcal{F}_{att}$ coincides with the function called $Eq_{inverse}$ in [13]. [13] considers also the presence of a support relation, but treated as a potential "vehicle" for attacks, in the sense that if an argument a supports another argument b, an attacker of a is also considered as an (indirect) attacker of b and contributes to decreasing its score. On the other hand a supporting argument cannot increase the score of the supported argument. This view is coherent with the absence of a base score and is clearly alternative to ours.

Other approaches to quantitative valuation have been proposed in the context of Dung's abstract argumentation where only the attack relation is encompassed. For example, [18] proposes a game-theoretic approach to evaluate argument strength in abstract argumentation frameworks. In a nutshell, the strength of an argument x is the value of a game of argumentation strategy played by the proponent of x. The approach does not encompass a support relations nor base scores: extending this game-theoretic perspective with these notions appears to be a significant direction of future investigation. Also, in weighted argumentation frameworks [11], real valued weights are assigned to attacks (rather than to arguments). These weights are not meant to be a basis for scoring arguments, rather they represent the "amount of inconsistency" carried by an attack. This use of weights is clearly different from ours and, in a sense, complementary. Investigating a combination of these two kinds of valuations (possibly considering also weights for support links) is a further interesting direction of future work.

Our system extends an existing IBIS-based tool, designVUE, already used in the engineering domain and in particular familiar to some of the experts responsible for our case studies. Other IBIS-based system exist in the literature. For example, Cohere and Compendium [7,6] adopt an IBIS methodology to support design rationale in collaborative settings. However, these systems do not incorporate means to automatically evaluate debates. Other examples are the Carneades [14] and the PARMENIDES [2] systems. These adopt a more articulate model of debate as they use argument schemes and critical questions as basic building blocks of the argumentation process. However, they do not incorporate a numerical evaluation of positions in debates. The extension of these other systems to take advantage of our scoring methodology is a possible direction of future work.

7 Conclusions

We presented a novel argumentation-based formal framework for quantitative assessment of design alternatives, its implementation in the designVUE software tool, and its preliminary experimentation in two case studies. Several directions of future work can be considered. On the theoretical side, a more extensive analysis of the properties of the proposed score function is under way, along with the study of alternative score functions exhibiting a different behavior (e.g. concerning the effect of attacks and supports and their balance) while satisfying the same basic requirements. On the implementation side, we plan to integrate the QuAD framework in the web-based debate system `www.quaestio-it.com` so to gain experience on its acceptability by users in other domains. On the experimentation side, the development of further engineering design case studies (more complex and in other domains) is under way and we intend to carry out a detailed on field comparison with more traditional approaches to the evaluation of design alternatives.

Acknowledgments. The authors thank V. Evripidou and E. Marfisi for their support and cooperation. Aurisicchio and Toni thank the support of a Faculty of Engineering EPSRC Internal Project on 'Engineering design knowledge capture and feedback'. The authors also thank the anonymous reviewers for their detailed and helpful comments.

References

1. designVUE (February 2013), `http://www3.imperial.ac.uk/designengineering/tools/designvue`
2. Atkinson, K., Bench-Capon, T.J.M., McBurney, P.: PARMENIDES: Facilitating deliberation in democracies. Artificial Intelligence and Law 14(4), 261–275 (2006)
3. Aurisicchio, M., Bracewell, R.H.: Capturing an integrated design information space with a diagram based approach. Journal of Engineering Design 24, 397–428 (2013)
4. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. Knowledge Eng. Review 26(4), 365–410 (2011)
5. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence 128(1-2), 203–235 (2001)
6. Buckingham Shum, S.J., Selvin, A.M., Sierhuis, M., Conklin, J., Haley, C.B., Nuseibeh, B.: Hypermedia support for argumentation-based rationale: 15 years on from gIBIS and QOC. In: Dutoit, A.H., McCall, R., Mistrik, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 111–132. Springer (2006)

7. Buckingham Shum, S.J.: Cohere: Towards web 2.0 argumentation. In: Besnard, P., Doutre, S., Hunter, A. (eds.) Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28-30. Frontiers in Artificial Intelligence and Applications, vol. 172, pp. 97–108. IOS Press (2008)
8. Cayrol, C., Lagasquie-Schiex, M.C.: Gradual valuation for bipolar argumentation frameworks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 366–377. Springer, Heidelberg (2005)
9. Cayrol, C., Lagasquie-Schiex, M.C.: On the acceptability of arguments in bipolar argumentation frameworks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 378–389. Springer, Heidelberg (2005)
10. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–357 (1995)
11. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results. Artificial Intelligence 175(2), 457–486 (2011)
12. Evripidou, V., Toni, F.: Argumentation and voting for an intelligent user empowering business directory on the web. In: Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR 2012), pp. 209–212 (2012)
13. Gabbay, D.M.: Equational approach to argumentation networks. Argument & Computation 3(2-3), 87–142 (2012)
14. Gordon, T.F., Walton, D.: The Carneades argumentation framework - using presumptions and exceptions to model critical questions. In: Dunne, P.E., Bench-Capon, T.J.M. (eds.) Computational Models of Argument: Proceedings of COMMA 2006, Liverpool, UK, September 11-12 (2006)
15. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer (2000)
16. Kunz, W., Rittel, H.: Issues as elements of information systems. Working Paper 131, Institute of Urban and Regional Development. University of California, Berkeley, California (1970)
17. Leite, J., Martins, J.: Social abstract argumentation. In: Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011), pp. 2287–2292 (2011)
18. Matt, P.-A., Toni, F.: A game-theoretic measure of argument strength for abstract argumentation. In: Hölldobler, S., Lutz, C., Wansing, H. (eds.) JELIA 2008. LNCS (LNAI), vol. 5293, pp. 285–297. Springer, Heidelberg (2008)
19. Pahl, G., Beitz, W.: Engineering design: a systematic approach. Tech. rep., Design Council, London, UK (1984)
20. Pugh, S.: Total Design: Integrated Methods for Successful Product Engineering. Addison-Wesley (1991)
21. Saaty, T.L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill (1980)
22. Simon, H.A.: The Sciences of the Artificial, 3rd edn. The MIT Press (1996)
23. Simon, H.A., Newell, A.: Human problem solving: The state of the theory in 1970. American Psychologist 26(2), 145–159 (1971)