HAL archives-ouvertes.fr

Argumentation for conflicting viewpoint relational database querying

Nouredine Tamani, Madalina Croitoru

To cite this version:

Nouredine Tamani, Madalina Croitoru. Argumentation for conflicting viewpoint relational database querying. RR-13037, 2013. <lirmm-00915967>

HAL Id: lirmm-00915967

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00915967

Submitted on 9 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Argumentation for conflicting viewpoint relational database querying: **Technical Report**

Nouredine Tamani, Madalina Croitoru
EPI GrapIK, INRIA, University of Montpellier 2, France

Abstract

Within the framework of the European project EcoBioCap, we model a real world use case aiming at conceiving the next generation of food packagings. The objective is to select packaging materials according to possibly conflicting requirements expressed by the involved parties (food and packaging industries, health authorities, consumers, waste management authority, etc.). The requirements and user preferences are modeled by several ontological rules provided by the stakeholders expressing their viewpoints and expertise. Since several aspects need to be considered (CO2 and O2 permeance, interaction with the product, sanitary, cost, end of life, etc.) in order to select objects, an argumentation process can be used to express/reason about different aspects or criteria describing the packagings. We define then in this report an argumentation approach which combines a description logic (DLR-Lite) within ASPIC framework for relational database querying. The argumentation step is finally used to express and/or enrich a bipolar query employed for packaging selection.

1 Introduction

Within the framework of the European project EcoBioCap (www.ecobiocap.eu) about the design of next generation packagings using advanced composite structures based on constituents derived from the food industry, we aim at developing a Decision Support System (DSS) for packaging material selection. The DSS will consist of two steps: (1) aggregating possibly conflicting needs expressed by several parties involved in the considered field and (2) querying a database of packagings with the resulting aggregation obtained at point (1). For example, in order to pack cheese, researchers focus on the permeance properties in their following argument: *"a wheat gluten based material packaging is suitable for cheese because it offers a good atmosphere control"*, and cheese producers can retort with the following counter-argument *"a gluten layer cannot put in contact with cheese since the bacteria in the crust would degrade the packaging"*, considering here the interaction with the packed food aspect. In this real case, packagings have to be selected according several aspects or criteria (permeance, interaction with the packed food, end of life, etc.), highlighted by the expressed stakeholders' arguments. The problem at hand does not simply consist in addressing a multi-criteria optimization problem [6]: the domain experts would need to be able to justify why a certain packaging (or set of possible packagings) are chosen. Argumentation theory in general [13, 4, 20] is actively pursued in the literature, some approaches even combining argumentation and multi criteria decision making [2].

These arguments are based on syllogism reasoning. For instance, given that all packagings providing a well controlled atmosphere are suitable, and since wheat gluten based materiel packaging is a well controlled atmosphere packaging, then it is suitable too. To be able to model this kind of reasoning, we need a structured format for arguments in which it is possible to express the involved concepts and rules modeling the semantics behind the text. Therefore, we rely in this work on a logical structured argumentation system [1, 19, 17] since it (i) allows the expression of logical arguments as a combination of facts and inference rules, (ii) defines attacks and defeat relations between arguments based on a logical conflict notion.

Furthermore, the reasoning process underlying arguments is related to a domain which can be easily set-based interpreted. It is then possible to consider certain subsets of description logics to define concepts and rules forming arguments, for their well known trade off between a good level of expressivity and computational decidability. Here we make a choice to use the DLR-Lite description logic [11, 7] since it also allows a direct connection to a database schema and relational tables needed in the second step of querying. Concepts are defined by m-ary relations connected to sets of tuples from a database and rules can be expressed by natural subsumption between concepts. In this way, the interpretation of the arguments is closely related to the domain of definition of a database, and the result of the argumentation process can be directly harnessed in the querying phase.

Stakeholder's set of arguments i is then modeled as concepts, facts and rules to build a partial knowledge bases $\mathcal{K}_{\mathcal{I}_i}$. The union of every stakeholder knowledge base $\mathcal{K} = \bigcup_{i=1,\dots,n} \mathcal{K}_{\mathcal{I}_i}$ will be used to instantiate the ASPIC[1] argumentation system. Indeed, ASPIC is based on a logical language equipped with a contrariness function and certain assumptions on the form of inference rules. The attack and defeat relations are then inferred based on this logical language, and rationality postulates (completeness under closure of strict rules, and consistency) are used to ensure the quality of the output. ASPIC+ [19, 17] extends ASPIC in a number of ways but the extensions are not needed for the simple logic language we employ in this report. The language is quite simple (in terms of expressivity) thus the ASPIC defined argument and attack definitions are enough for instantiation.

The solution developed in this report is to instantiate for each criterion, called viewpoint or aspect, an argumentation system to reason about arguments solely expressed on it. This will then be used to generate the query on the packaging database. The main contribution of this report is to demonstrate the use of argumentation in a real world industrial scenario within the EcoBioCap project. To this aim we show how to instantiate ASPIC with the DLR-Lite logic modeling expert ontologies in this real world scenario.

In Section 2 we introduce parts of the designed DSS for the Ecobiocap project. In Section 3, we detail our approach defining a DLR-Lite based argumentation theory relying on ASPIC. In section 4, we detail our modeling choice of the rules expressed within the text arguments and we point out its limitation. Then, we propose in Section 5 a solution based on viewpoints to overcome the modeling limitation. Section 6 is dedicated to the implementation and the test of the approach. Section 7 sums up some related works, and finally, in Section 8 we recall our contributions and introduce some perspectives.

2 A DSS for the EcobioCap project

The Ecobiocap workflow is divided into 7 work packages, denoted WP in figure 1. As part of the WP1, dealing with industrial and environmental concerns, we are involved in the design of a decision support system (DSS) allowing the users to select packaging materials, based on such aspects. The DSS is made of two parts: (i) a flexible querying process which is based on

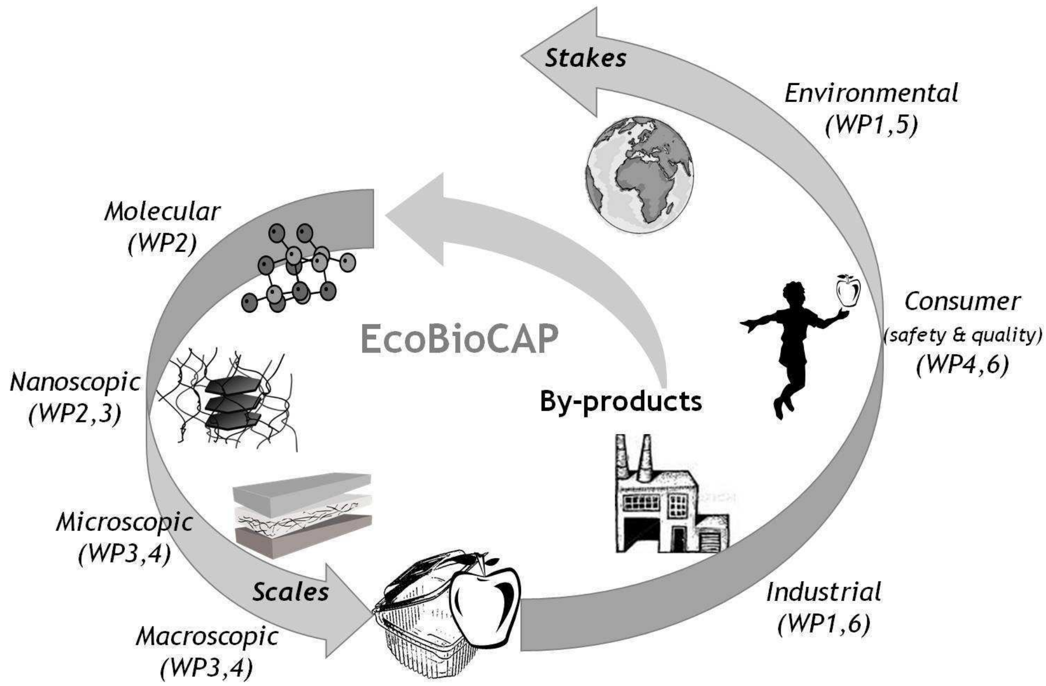

Figure 1: The EcoBioCap workflow.

a bipolar querying approach [12] dealing with imprecise data corresponding to the characteristics related to the food product to pack like the optimal permeance, the dimension of the packaging, its shape, etc., and (ii) an argumentation process which aims at aggregating several stakeholders requirements expressed as simple text arguments, to enrich the querying process by stakeholders' justified preferences. The former implements a database containing the respiration parameters of the packed food, and a second database storing the characteristics of packaging materials (O_2 and CO_2 permeance, biodegradability, transparency, etc.). The user can also specify some preferences such as the preferred storage temperature, dimensions of the packaging, etc, which could be mandatory or optional. This part of the software combines these inputs to compute the optimal permeance which guaranties the best shelf life for the packed food. Then, this optimal permeance are mixed with other user preference to form a bipolar query addressed to the packaging database. The returned list of packaging is ranked from the most to the least relevant one with regard to the expressed preferences.

To collect the user preferences, different surveys have been carried out among the stakeholders involved in the design of packaging: researches, packaging industries, food producers, consumers, etc. The asked questions are about cost, end of life, biodegradability, the use of nano-particles in the packaging, etc. For each criterion expressed through surveys, the stakeholders identified its importance, indicated the preferred values, and the reasons that justify their choice. The importance permits to give a priority to criteria over others, used in the bipolar approach considering mandatory and optional preferences. For instance, sanitary criteria ensuring a good preservation of the packed food are naturally more important than the color or the transparency of the packaging. The values, in the other hand, can be easily used as predicate in the bipolar query. But, in the current version of the DSS, the justifications associated to criteria have not been used in this querying process. These justifications are then seen in this report as arguments pros or cons some choices or values. They can justify why a packaging is better than another and can be then used to enrich the bipolar querying system. So, we detail in the next sections how arguments are logically modeled within a structured argumentation system and how the delivered justified conclusions can be used in the querying process.

3 Ontology-based argumentation theory

We describe in the following subsections the elements of our ASPIC argumentation system AS based on the DLR-Lite ontology extended to a negation.

3.1 Language specifications

In this report, we only consider the crisp interpretation of fuzzy DLR-Lite [11, 8] specification based on a database.

3.1.1 A reminder of the DLR-Lite

A DLR-Lite is a restriction of the fuzzy DLR-Lite [11, 8] knowledge base to crisp facts, rules and operators. A DLR-Lite is made of facts ($\mathcal{F}$) extracted from a database, abstraction rules ($\mathcal{A}bs$) and an ontology component ($\mathcal{O}$).

The *facts component* $\mathcal{F}$ contains tuples from relational tables of the form $R(c_1, ..., c_m)$, where R is an m-ary relation.

The *ontology component* $\mathcal{O}$ defines the domain concepts by means of projection, intersection and inclusion axioms.

Inclusion axioms have the form $(R_1 \sqcap R_2 \sqcap ... \sqcap R_l \sqsubseteq R_{l'})$, where $l \geq 1$, $R_{i,i=1,...,l}$ and $R_{i'}$ are relations of the same arity. This inclusion expresses a subsumption relation between the concept built by the intersection of the m-ary relations $R_{i,i=1,...,l}$ and the the right-hand side relation $R_{i'}$.

Projection axioms are used to define new concepts by extracting columns from relational tables. *Simple* and *restricted projections* are defined. The former is of the form $\exists[i_1, ..., i_k]R$, and expresses the projection of relation R on columns $i_1, ..., i_k$, and the latter is of the form $\exists[i_1, ..., i_k]R.$
$(Cond_1 \sqcap ... \sqcap Cond_h)$, and restricts the projection of R according to conditions $Cond_{i,i=1...h}$ of the form $([i]\theta v)$, such that $\theta \in \{<, \leq, =, \neq, >, \geq\}$ and v is a value.

The *abstraction component* $\mathcal{A}bs$ defines statements which express relationships between concepts defined in the ontology component and tuples defined in the facts component. Simple and complex abstractions are defined. *Simple abstraction statement* has the form $R_1 \longmapsto R_2(c_1[t_1], ..., c_k[t_k])$, which states that k-ary relation $R_1 \in \mathcal{O}$ is mapped into the projection on columns c_i of type $t_i, i = 1, ..., k$ of the m-ary table R_2, such that $k \leq m$. *Complex abstraction statement* has the form $R_1 \longmapsto (t_1, ..., t_k).sql$, where R_1 is a k-ary relation in $\mathcal{O}$ and *sql* is an SQL query which returns k-ary tuples $\langle t_1, ..., t_k \rangle$.

3.1.2 DLR-Lite language: the syntax

Definition 1 *(m-ary relations R). An m-ary relation $\rho(a_1, ..., a_m)$ is a complex concept made of m attributes $a_1, ..., a_m$ from a database schema. Each attribute a_i defines a data property which associates each instance from ρ a value from the domain of definition of a_i. The cartesian product of domains of $a_1, ..., a_m$ forms the domain of definition of ρ. The set of m-ary relations is denoted $\Re$.*

Example 1 *We consider a database about packagings, we can consider the m-ary relation* Packaging (id, color, shelflife, machine, labellingType, material, cost) *of arity* 7.

The language defines the operators of projection (π), restriction (σ), abstraction (α), and intersection ($\sqcap$) as defined in the DLR-Lite. It is also extended to a negation operator.

3.1.3 DLR-Lite language: the semantics

In ontologies, the semantics associated to concepts and roles is set-based, and each concept describes a set of objects sharing the same characteristics. The conjunction and subsumption between concepts are interpreted respectively by the intersection and the inclusion between their corresponding sets of objects. An interpretation is a pair $(\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set called the domain of interpretation, and $.^{\mathcal{I}}$ is an interpretation function mapping the concept constructors into subsets of $\Delta^{\mathcal{I}}$ where:

- The semantics attached to operators of projection, restriction, and intersection is the same as in the relational algebra and in the SQL language,
- An m-ary relation is defined by its relational table.

3.1.4 Extension of the language to a restricted negation operator

The set-based interpretation of the negation operator in crisp DL is $(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} - A^{\mathcal{I}}$. This interpretation is not adequate for an ontology-based database querying process because it does not satisfy the inclusion set property: if $A \subseteq B$ then $\tilde{\neg} B \subseteq \tilde{\neg} A$, which is important w.r.t. the inference rules and the rationality postulate concerning the closure under strict rules (studied below) in the argumentation system.

So, we propose a restricted negation, denoted $\tilde{\neg}$, having the following interpretation:

$$(\tilde{\neg} A)^{\mathcal{I}} = G_A^{\mathcal{I}} - A^{\mathcal{I}} = G_A^{\mathcal{I}} \cap \bar{A}^{\mathcal{I}} \tag{1}$$

where $G_A^{\mathcal{I}}$ is the concept generator of A (defined below), and $\bar{A}^{\mathcal{I}}$ is the complement of $A^{\mathcal{I}}$.

Definition 2 *(Concept generator G_A). Let A be a concept in an ontology hierarchy. Its concept generator G_A is the abstract concept $Thing$ if $Thing$ is the direct ancestor of A. Otherwise, G_A is the most generic ancestor of the same domain of A having $Thing$ as direct ancestor.*

It is worth noticing that a concept could have more than one concept generator. This case happens when a concept A is subsumed in the ontology by several super-concepts, which are in their turn derived from the most abstract concept of the ontology ($Thing$ in domain ontologies). The definition of G_A is then the union of the super-concepts the concept A is derived from.

Definition 3 *(Generalized concept generator). Let A and $S_A = \{A_1, ..., A_n\}$ be respectively a concept from an ontology hierarchy its ancestors directly derived from Thing. The concept generator of A is then $G_A = \bigcup_{i=1,\dots,n}(A_i)$.*

The interpretation of the restricted negation in this case becomes then:

$$(\tilde{\neg}A)^{\mathcal{I}} = (\bigcup_{i=1,\dots,n}(A_i^{\mathcal{I}})) - A^{\mathcal{I}} = (\bigcup_{i=1,\dots,n}(A_i^{\mathcal{I}})) \cap \bar{A}^{\mathcal{I}} \tag{2}$$

Remark 1 *If $S_A = \{A_1\}$ then formula (2) is equivalent to formula (1).*

3.2 DLR-Lite based argumentation framework

3.2.1 Dung argumentation principles

Dung argumentation framework (AF) [13] is a tuple $(\mathcal{A}, \mathcal{C})$, where $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary attack relation on the set of arguments $\mathcal{A}$. For each argument $X \in \mathcal{A}$, X is *acceptable* w.r.t. a set of arguments $S \subseteq \mathcal{A}$ iff any argument attacking X, is attacked by an argument of S. A set of arguments $S \subseteq \mathcal{A}$ is *conflict free* iff $\forall X, Y \in S, (X, Y) \notin \mathcal{C}$. For any conflict free set of arguments S, S is an *admissible extension* iff $X \in S$ implies X is acceptable w.r.t. S; S is a *complete extension* iff $X \in S$ whenever X is acceptable w.r.t. S; S is a *preferred extension* iff it is a set inclusion maximal complete extension; S is the *grounded extension* iff it is the set inclusion minimal complete extension; S is a *stable extension* iff it is preferred and $\forall Y \notin S, \exists X \in S$ such that $(X, Y) \in \mathcal{C}$.

For $T \in$ {complete, preferred, grounded, stable}, X is skeptically (resp. credulously) justified under the T semantics if X belongs to all (resp. at least one) T extension. Output of an extension E is $Concs(E) = \{Conc(A), A \in E\}$, where $Conc(A)$ is the conclusion of argument A. The skeptical output of AF is $Output(AF) = \bigcap_{i=1,\dots,n} Concs(E_i)$ such that E_i are its T extensions.

3.2.2 ASPIC knowledge base and arguments

In this report we consider a simpler ASPIC argument structure than the ones presented in [19] . The subset of ASPIC+ we use is compatible to the one presented in [1]. In the following an ASPIC argumentation system is denoted $\mathcal{AS} = (\mathcal{L}, cf, \mathcal{R}, \geq)$, where:

- $\mathcal{L}$ is a logical language,

- cf is a contrariness function which associates to each formula of $\mathcal{L}$ a set of its incompatible formulas (in $2^{\mathcal{L}}$),

- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is the set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules of the form $\varphi_1, ..., \varphi_m \rightarrow \varphi$ and $\varphi_1, ..., \varphi_m \Rightarrow \varphi$ respectively, where φ_i, φ are well-formed formulas in $\mathcal{L}$, and $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$,

- $\geq$ is a preference ordering over defeasible rules.

A knowledge base in an $AS = (\mathcal{L}, cf, \mathcal{R}, \geq)$ is $\mathcal{K} \subseteq \mathcal{L}$ such that $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_p$ and $\mathcal{K}_a \cap \mathcal{K}_p = \emptyset$, $\mathcal{K}_a$ contains axioms and $\mathcal{K}_p$ contains ordinary premises.

3.2.3 The contrariness function on $\mathcal{L}$

The contrariness function $cf : \mathcal{L}' \longrightarrow 2^{\mathcal{L}'}$, where $\mathcal{L}' = \mathcal{L} \backslash \{\pi, \sigma, \alpha\}$, has to specify to each formula of $\mathcal{L}'$ at least a contradictory formula. We discard from $\mathcal{L}$ operators $\{\pi, \sigma, \alpha\}$ since they refer to relational operators which define facts from a database and are not directly involved in logical formulas. In our case, $cf = \tilde{\neg}$ with: $cf(A) = \{\tilde{\neg} A\}$, $cf(\tilde{\neg} A) = \{A\}$, and $cf(A_1 \sqcap A_2) = \{\tilde{\neg}(A_1 \sqcap A_2)\}$.

We notice that the union operator is not defined in the DLR-Lite. We can obtain it indirectly by combining the negation and the intersection operators.

3.2.4 Inference rules $\mathcal{R}$

We define the following rules:

- Strict/defeasible subsumption: a strict subsumption, denoted $\sqsubset$, expresses natural inclusion in the domain, as *"GlutenPackaging is a Packaging"*. A defeasible subsumption, denoted $\Subset$, expresses an inclusion involving a user choice, as *"GlutenPackaging is a suited Packaging"*. So, $R_1 \sqsubset / \Subset R_2$ with R_1, R_2 are two relations of the same arity, means that $\forall x \in R_1$ then $x \in R_2$,

- Intersection: let $R_1, ..., R_n$ and $R_{n'}$ be m-ary relations (of the same arity). The rule $R_1 \sqcap R_2 \sqcap ... \sqcap R_n \sqsubset / \Subset R_{n'}$ means that if $x \in R_{k,k=1,...,n}$, then $x \in R_{n'}$. This rule is denoted simply by $R_1, ..., R_n \sqsubset / \Subset R_{n'}$,

- Transitivity: let R_1, R_2, R_3 be three relations of the same arity. If $(R_1 \sqsubset / \Subset R_2) \wedge (R_2 \sqsubset / \Subset R_3)$, then $R_1 \sqsubset / \Subset R_3$,

- Contraposition: if $(r_1 \sqsubseteq / \Subset r_2)$ then $\tilde{\neg} r_2 \sqsubseteq / \Subset \tilde{\neg} r_1$.

3.2.5 DLR-Lite based ASPIC argumentation system

In our application case:

- $\mathcal{K}_a$ contains the concepts of the considered domain, and $\mathcal{K}_p$ contains user choices,

- $\mathcal{L}$ is the language defined in subsection 3.1,

- cf is the contrariness function corresponding to $\tilde{\neg}$,

- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$, where $\mathcal{R}_s$ / $\mathcal{R}_d$ is the set of strict / defeasible subsumptions as defined in Subsection 3.2.4,

- $\geq$ is a preorder on defeasible rules, not defined.

A DLR-Lite ASPIC argument A can be of the following forms:

1. $\emptyset \sqsubseteq / \Subset c$ with $c \in \mathcal{K}_a/\mathcal{K}_p$ such that $Prem(A) = \emptyset$, $Conc(A) = c$, $Sub(A) = \{A\}$, $Rules(A) = \emptyset$,
$TopRule(A) = undefined$ with $Prem$ returns premisses of A, $Conc$ returns its conclusion, Sub returns its sub-arguments, $Rules$ returns rules applied on A, $TopRule$ returns the last rule applied,

2. $A_1, ..., A_m \sqsubseteq / \Subset c$, such that there exists a strict / defeasible rule in $\mathcal{R}_s/\mathcal{R}_d$ of the form
$Conc(A_1) \sqcap ... \sqcap Conc(A_m) \sqsubseteq / \Subset c$, and $Prem(A) = Prem(A_1) \cup ... \cup Prem(A_m)$, $Conc(A) = c$, $Sub(A) = Sub(A_1) \cup ... \cup Sub(A_m) \cup \{A\}$, $Rules(A) = Rules(A_1) \cup ... \cup Rules(A_m) \cup \{Conc(A_1), ..., Conc(A_m) \sqsubseteq / \Subset c\}$, $TopRule\ (A) = Conc(A_1), ..., Conc(A_m) \sqsubseteq / \Subset c$.

We make the assumption that the set of arguments constructed from the argumentation system is finite. An argument is said strict iff it does not involve any defeasible rules. Otherwise, it is called defeasible.

The set of strict rules $\mathcal{R}_s$ is consistent iff $\nexists A, B$ such that A, B are strict arguments and $Conc(A) = \tilde{\neg} Conc(B)$.

Remark 2 *(Notation). An ASPIC argument has a nested form. A subargument is also an argument. To improve the readability, by abuse of notation, we associate to each argument a label made of a capital letter followed by a subscript number. The labels are then used in an argument to refer to its subarguments. In this notation, a label followed by colon is not a part of the argument.*

Example 2 *Let AS be an ASPIC argumentation system defining the rules* $\mathcal{R}_s = \{a \sqcap b \sqsubset c\}$ *and the ordinary premises* $\mathcal{K}_p = \{a, b\}$*. The following arguments can be built:*

- $A_1 : \emptyset \Subset a$
- $A_2 : \emptyset \Subset b$
- $A_3 : A_1, A_2 \sqsubset c.$

3.2.6 DLR-Lite based ASPIC attacks and defeat

We considered rebutting and undercutting attacks. Argument A *rebuts* argument B iff $\exists A' \in Sub(A) : Conc(A') = \varphi$ and $B' \in Sub(B)$ such that B' is of form $B'_1, ..., B'_m \Subset \tilde{\neg}\varphi$. It is defined as a restricted rebut attack [9] to satisfy rationality postulates as shown in [9].

The undermining attack [19] is a particular case of the rebutting attack (A undermines B on $B' = \varphi$ iff $Conc(A) \in \bar{\varphi}$ and $\varphi \in Prem_p(B)$). An ordinary premise is a conclusion of an argument of form $\emptyset \Subset C$, $C \in \mathcal{K}_p$.

Argument A *undercuts* argument B iff $\exists B' \in Sub(B)$ of form $B'_1, ..., B'_m \sqsubseteq \varphi$ and $\exists A' \in Sub(A) : Conc(A') = \neg \lceil Conc(B''_1), ..., Conc(B''_n) \Subset \varphi \rceil$, s.t. operator $\lceil . \rceil$ converts a defeasible rule into a literal.

Then, A *defeat* B if A rebuts or undercuts B.

3.2.7 Rationality postulates

In [9], it has been shown that the closure under transposition of the set of strict rules (denoted Cl_{tp}) ensures the closure of the output, and $\mathcal{R}_s$ must be consistent to ensure consistency of its output, which corresponds to the c-consistence introduced in [17].

Since the implication operator is the logical counterpart of set inclusion operator, it is easy to show that the proposed AS satisfies the rationality postulates (closure, direct and indirect consistency). The proofs are similar to the ones in [19, 9]. We remind the postulates below.

Let AS be an argumentation system, and $E_1, ..., E_n$ is its extensions under one Dung's semantics, and its $Output = \bigcap_{i=1,...,n} Concs(E_i)$ (following [9], because credulous attitude can lead to inconsistencies).

• AS is closed under sub-arguments iff

$$\forall A \in E_i,\ Sub(A) \in E_i \text{ [19]},$$

• AS is closed under strict rules iff

(i) $\forall E_{i,i=1,\ldots,n}, Concs(E_i) = Cl_{\mathcal{R}_s}(Concs(E_i))$ and

(ii) $Output = Cl_{\mathcal{R}_s}(Output)$.

• As in [9], AS is directly consistent iff

$$\forall E_{i,i=1,\ldots,n}, Concs(E_i) \text{ and } Output \text{ are consistent,}$$

• AS is indirectly consistent iff $\forall E_{i,i=1,\ldots,n}$, $Cl_{\mathcal{R}_s}(Concs(E_i))$ and $Cl_{\mathcal{R}_s}(Output)$ are consistent.

Property 1 *DLR-Lite based ASPIC argumentation system is closed under sub-arguments, closed under strict rules, direct and indirect consistent.*

4 Modeling rules in the DLR-Lite based argumentation system

In this section, we investigate the way to model the rules involved in the arguments. We recall that rules can be either strict or defeasible. It is intuitive to consider ontological rules such as *a wheat gluten based packaging is a packaging* or *a wheat gluten based packaging is a packaging which provides a good atmosphere control* as strict rule since they represent real and measured facts, and rules such as *the user prefers wheat gluten based packaging* or *packaging providing a good atmosphere control is suitable packaging* as defeasible, since they express context-based and questionable user choices. However, we can show through the following example a limitation of this modeling approach from the decision making standpoint, when the arguments are expressed on the same objects but according to different aspects.

Example 3 *We consider a database about packagings and two stakeholders: researchers and cheese producers arguing about cheese packagings. Researchers say:* "wheat gluten based packagings are suitable because a wheat gluten layer provides a good atmosphere control"*, and cheese producers say:* "gluten cannot be put in contact with cheese because the bacteria in the crust would eat the gluten, degrading then the packaging layer"*.*

Researchers ontology defines the following elements:

- *AtmControlPack, denoted ACP: is a concept referring to packagings that ensure a good atmosphere control,*

- *SuitedPack, denoted SP: a concept that refers to the set of acceptable packagings,*

- $ACP \Subset SP$*: a defeasible rule expressing that any well controlled atmosphere packaging is a suited packaging,*

- *GlutenPack, denoted GP: a concept defined by the restriction of the relation* Packaging *based on the condition* matPack = 'WGluten'. *As it represents a user choice then* $GP \in \mathcal{K}_p$,

- $GP \sqsubset ACP$*: a strict rule expressing the measurable knowledge relating the wheat gluten based material and the packaging permeance property.*

Cheese producers ontology defines the following elements:

- *SuitedPack and GlutenPack defined in the same way as in the researchers ontology,*

- *DestructivePack, denoted DP: this concept refers to destructive packagings,*

- $DP \Subset \tilde{\neg} SP$*: a defeasible rule expressing that any destructive based material packaging is not suitable,*

- $GP \sqsubset DP$*: this strict rule expresses the interaction with the packaging material and the packed food.*

These ontologies are merged to define the knowledge base on which we instantiate the DLR-Lite based argumentation system $AS = (\mathcal{L}, \mathcal{R}, \tilde{\neg}, \geq)$ *such that:*

- *Strict rules* $\mathcal{R}_s = Cl_{tp}\{GP \sqsubset ACP, GP \sqsubset DP\} =$
$$\{GP \sqsubset ACP, \tilde{\neg} ACP \sqsubset \tilde{\neg} GP, GP \sqsubset DP, \tilde{\neg} DP \sqsubset \tilde{\neg} GP\}$$

- *Defeasible rules* $\mathcal{R}_d = \{ACP \Subset SP, DP \Subset \tilde{\neg} SP\}$

- $\mathcal{K}_a = \{Packaging\}$ *and* $\mathcal{K}_p = \{GP\}$.

Researchers and Cheese producer's arguments are then:

- $A_0 : \emptyset \Subset GP$
- $A_1 : A_0 \sqsubset ACP$
- $A_2 : A_1 \Subset SP$
- $A_3 : A_0 \sqsubset DP$
- $A_4 : A_3 \Subset \tilde{\neg} SP$

We have a mutual attack between A_2 and A_4. We get then, for instance, two preferred extensions:

- $E_1 = \{A_0, A_1, A_2, A_3\}$ *and*
$$Output(E_1) = \{GP, ACP, DP, SP\}.$$
- $E_2 = \{A_0, A_1, A_3, A_4\}$ *and*
$$Output(E_2) = \{GP, ACP, DP, \tilde{\neg} SP\}.$$

The skeptical output (this follows the line of work of [9], because credulous attitude can lead to inconsistencies) of the argumentation system is then $Output = \{GP, ACP, DP\}$.

In this case, the argumentation system delivers no decision about the rejection and the acceptance of an option. When we consider only delivered extensions, we notice that the system recommends the acceptance and the rejection of the considered option (GP) based on the same knowledge about it ($\{GP, ACP, AP, DP\}$). One can suggest to model all the rules as defeasible but we obtain the same extensions inasmuch as the rules of acceptance and rejection are defeasible.

The behavior described in Example 3 shows that the problem lies in the fact that the system is unable to catch that an object can be at the same time accepted according to a criterion and rejected according to another. To make the system able to capture this way of reasoning, we have to consider the rules from criteria point of view. In other words, when the system applies the rules defined on a criterion, it must discard all the rules (defeasible and strict rules) defined on other criteria. Then, when an agent expresses an argument over one criterion (according to its knowledge), he/she does not presume any knowledge related to the other criteria.

We define then in the next section a viewpoint argumentation system in which a viewpoint corresponds to an instance of the DLR-Lite based argumentation system limited to the rules defined on one and only one criterion (attribute) describing the objects of the considered domain. The approach defined next is similar in spirit to the intuition presented in the approach of [2]. The difference with this work is the logical language used ([2] uses propositional logic) and the practical scenario demonstrated in the report.

5 Viewpoints DLR-Lite based argumentation system

Rather than merging the knowledge expressed by the involved part in the project, here we split rules according to the considered aspects. We then instantiate on each subset of rules the DLR-Lite based argumentation system, as shown in the following example.

Example 4 *We consider the argument expressed in Example 3. From the researchers' standpoint, the criterion* atmosphere control *(denoted atm) is considered, whereas cheese makers consider the* interaction with the product *criterion (denoted int).*
Researchers' arguments are based on viewpoint $v_1 = v_{atm}$*:*

- $\mathcal{R}_{v_1} = (\mathcal{R}_{sv_1}, R_{dv_1})$, *with* $\mathcal{R}_{sv_1} = \{GP \sqsubset ACP,$ $\tilde{\neg}ACP \sqsubset \tilde{\neg}GP\}$ *and* $R_{dv_1} = \{ACP \Subset SP\}$

Cheese makers' arguments are based on viewpoint $v_2 = v_{int}$*:*

- $\mathcal{R}_{sv_2} = (\mathcal{R}_{sv_2}, \mathcal{R}_{dv_2})$ *with* $\mathcal{R}_{sv_2} = \{\ GP \sqsubset DP,$ $\tilde{\neg}DP \sqsubset \tilde{\neg}GP\}$ *and* $R_{dv_2} = \{DP \Subset \tilde{\neg}SP\}$.

Researchers' arguments under $\mathcal{R}_{sv_1}$ *and* $\mathcal{R}_{dv_1}$ *are:*

- $A_1 : \emptyset \Subset GP$
- $A_2 : A_1 \sqsubset ACP$
- $A_3 : A_2 \Subset SP$

The set of arguments $a = \{A_1, A_2, A_3\}$ *forms the only preferred extension in the defined view* v_{atm}*, and*
$Output(v_{atm}) = Concs(a) = \{GP, ACP, SP\}$.

The cheese makers' arguments are:

- $B_1 : \emptyset \Subset GP$
- $B_2 : B_1 \sqsubset DP$
- $B_3 : B_2 \Subset \tilde{\neg}SP$

The set of arguments $b = \{B_1, B_2, B_3\}$ *forms the only preferred extension in view* v_{san}*, and*
$Output(v_{san}) = Concs(b) = \{GP, DP, \tilde{\neg}SP\}$.

We obtained in Example 4 two views v_{atm} and v_{int} made of arguments $\{A_1, A_2, A_3\}$ and $\{B_1, B_2, B_3\}$ respectively, which do not attack each other (according to the definition of attacks), but their conclusions are contradictory.

We can gather views according to goals to form collections, which support/oppose them.

Example 5 *In Example 4, we can form collections* $C_{SP} = \{v_{atm}\}$ *and* $C_{\tilde{\neg}SP} = \{v_{int}\}$ *on goals* $\{SP, \tilde{\neg}SP\}$*, which indicate the (in)compatible criteria for gluten based material packagings selection.*

5.1 Rationality postulates in a collection

Each viewpoint satisfies the rationality postulates since it an instance of the DLR-Lite based argumentation system. It is also important that the delivered collections are consistent and closed under the strict rules involved in them to ensure the completeness of the reasoning process. Let $C_g = \{v_1, ..., v_{m'}\}$ be a collection of m' views on goal g. The strict rules of C_g is $\mathcal{R}_{sC_g} = \bigcup_{i=1,\dots,m'} \mathcal{R}_{sv_i}$. C_g is closed under $\mathcal{R}_{sC_g}$ iff $Output(C_g) = Cl_{\mathcal{R}_{sC}}(Output(C_g))$.

Property 2 C_g *is closed under* $\mathcal{R}_{sC_g}$.

Property 3 *If* $\mathcal{R}_{sC_g}$ *is consistent then its output is also consistent.*

Finally, a viewpoint argumentation system limits the effect of contaminating formulas [10], since ill-formed rules cannot contaminate other syntactically disjoint viewpoints, and it is possible to discard an ill-formed rule from a viewpoint as in [24] to prevent its crash.

5.2 Using the collections in database querying

The instantiation of our argumentation system is used to write a consensual query encompassing justified (by arguments) stakeholders' requirements and needs. We can then deduce automatically such queries from the collections the users formed during the argumentation process. Indeed, we can carry out an analogical reasoning by generalizing results obtained from an argumentation process applied upon instances, where an instance of the sought objects can help to better understand the involved stakeholders' needs and then to be able to express, based on arguments pros and cons, a query reflecting the way objects should be selected from a database.

Example 6 *In example 4 about gluten based material packagings, the system can deduce that the desired packagings are those ensuring a good atmosphere control and having no risky interaction with the packed product. The retrieved packagings from the database are similar to the one used as samples in the argumentation process.*

6 Implementation and tests

The implementation of the approach has been done in the context of the EcoBioCap DSS. A java GXT/GWT web interface was developed and a open version is accessible on *pfl.grignon.inra.fr:5880/EcoBioCapProduction.* The main difficulties encountered were the translation of text arguments into DLR-Lite formal representation. In the freely available version, stakeholder' arguments are provided as a manually built XML file specifying viewpoints and rules. The system generates then arguments and attacks and computes the extensions (stable, preferred, admissible, grounded, naive, etc. semantics) inside each view. Figure 2 shows the main interface of the application and a fragment of rules formalizing an argumentation scenario about the aspect *end of life* of packagings. Stakeholders argued about biodegradability, recyclability and compostability (the test XML file is accessible on *https://docs.google.com/file/d/0B0DPgJDRNwb LR2RjWWhwMjgwVEU/edit?usp=sharing).* The following text arguments about the *end of life* have been collected during an interview with experts in the domain of waste management:

- Packaging materials (biodegradable, compostable, recyclable) with low environmental impact are preferred,
- Life Cycle Analysis (LCA) results are not in favor of biodegradable and compostable materials,
- Consumers are in favor of biodegradable material because they help to protect the environment,
- Biodegradable materials could encourage people to throw their packaging in nature, causing visual pollution.

From the obtained extensions (see Figure 3), we know the reasons why recyclable packagings are recommended and why biodegradable and compostable packagings are not. This result can be used in the querying process by adding to the query the predicate *"recycle = true"* to retrieve the recyclable packaging materials from the database.

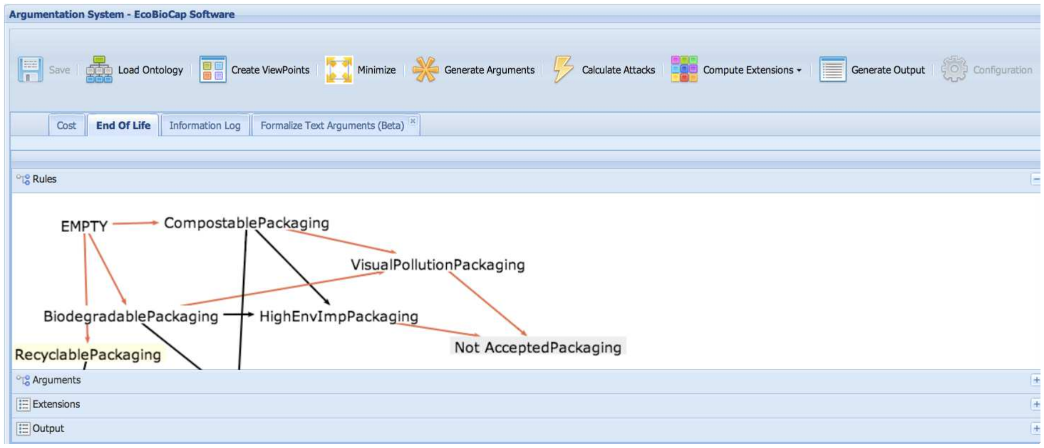

Figure 2: The user interface of the system.

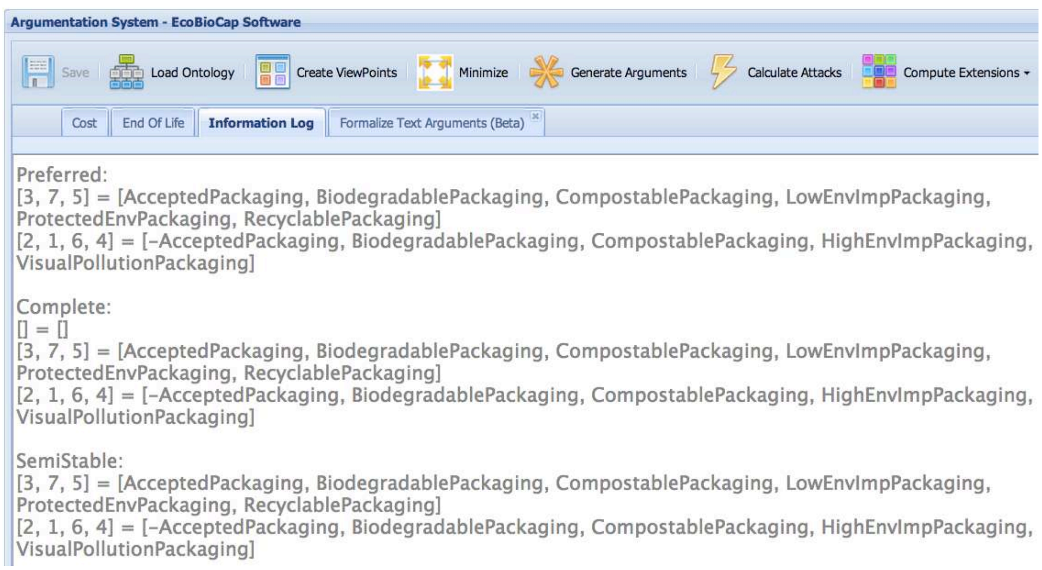

Figure 3: Extensions delivered for the viewpoint *end of life*.

7 Related work

There are several ways to combine ontologies and argumentation systems. We mention the developed systems relied on mapping DL ontologies into DeLP program [14] as in [22, 21, 3, 15, 23]. The system knowledge base in this case contains stable knowledge (formalized within the DL part of the system) and defeasible knowledge expressed as exceptions. Such system rely on having a goal, or query to answer, based on which the arguments supporting it are computed using backwards chaining algorithms. In our case such approach is not possible as the goals are not known apriori- the argumentation process allows to identify them. In [3], conflicts between rule conclusions are resolved by a superiority relation, defined between rules. In [16], the developed argumentation system uses the Defeasible Description Logic (DDL) as logical language to analyze the Semantic Security Policy Language (SSPL). It does not define attacks notion between arguments and only conflicts can be modeled. In [23] authors principally interested in whether or not two formulas are mutually inconsistent, or whether one entails the other. The mutual inconsistency corresponds to the notion of conflict. No attacks are defined in this approach and a preference relationship between defeasible rules is defined to resolve conflicts through the notion of defeaters. In the second category, logical modeling of arguments have been introduced based on the available operators in the considered description logics. In [18], an argumentation process is used to restore the consistency of an ontology ($\mathcal{ALC}$ description logic), based on acceptability semantics. Inconsistencies in the ontology have been presented as attacks in the argumentation graph. To resolve conflicts the user has to define a preference relation called argument comparison criterion which states the priority between arguments. The attack relation is then limited to conflicting arguments having the same level of priority. This process may produce extensions having conflicting arguments. In [5], a framework for logic-based argumentation with multiple ontologies via dialogues has been presented. The agent knowledge base in this approach contains formulas expressed by a description logic. Arguments are then logically structured but they have the same level of reliability since no notion of strictness and defeasibility can be expressed in the approach. Let us highlight that both above approaches do not consider rationality postulates [9], which ensure the completeness and the coherence of the result.

8 Conclusion

We applied in this report on a real use case from the industry an argumentation approach, based on a combination of an ASPIC argumentation system with a DLR-Lite specifications allowing stakeholders to express their preferences and providing the system with stable concepts and subsumptions of a domain. We have pointed out the fact that where stakeholders express their contradictory needs as arguments on several criteria, the argumentation system can returns unsatisfactory result. Therefore, we have proposed an argumentation system in which each criterion is considered as a viewpoint in which stakeholders express their arguments in homogenous way. The set of non conflicting viewpoints are then gathered according goals, to form consistent collections which support/oppose them.

The implementation of the module of arguments semi-automatic translation from text format to concepts and rules is in progress. As future work, we need a module of derivation of query predicates from the identified collections. We also plan to extend the proposed approach to fuzzy argumentation context to make it possible to deal with vague and uncertain concepts and rules by exploiting the fuzzy interpretation of the fuzzy DLR-Lite. Another line to develop consists of studying the bipolarity in our context of argumentation, since collections can be formed to support/oppose goals. Therefore a bipolar reasoning process can be considered as a refinement of the developed reasoning process based on argumentation.

References

[1] L. Amgoud, L. Bodenstaff, M. Caminada, P. McBurney, S. Parsons, H. Prakken, J. Veenen, and G. Vreeswijk. Final review and report on formal argumentation system.deliverable d2.6 aspic. Technical report, 2006.

[2] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009.

[3] G. Antoniou and A. Bikakis. Dr-prolog: a system for defeasible reasoning with rules and ontologies on the semantic web. *IEEE Transaction on Knowledge and Data Engineering*, 19(2):233–245, 2007.

[4] P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[5] E. Black, A. Hunter, and J. Pan. An argument-based approach to using multiple ontologies. In L. Godo and A. Pugliese, editors, *Scalable Uncertainty Management*, volume 5785 of *Lecture Notes in Computer Science*, pages 68–79. Springer Berlin Heidelberg, 2009.

[6] D. Bouyssou, D. Dubois, M. Pirlot, and H. Prade. *Decision-making process – Concepts and Methods*. Wiley, 2009.

[7] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Dl-lite: Tractable description logics for ontologies. In *AAAI*, pages 602–607, 2005.

[8] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *KR*, pages 260–270, 2006.

[9] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171:286–310, 2007.

[10] M. W. A. Caminada, W. A. Carnielli, and P. E. Dunne. Semi-stable semantics. *Journal of Logic and Computation*, pages 1–45, 2011.

[11] S. Colucci, T. D. Noia, A. Ragone, M. Ruta, U. Straccia, and E. Tinelli. *Semantic Web Information Management*, chapter 19 : Informative Top-k retrieval for advanced skill management, pages 449–476. Springer-Verlag Belin Heidelberg, 2010.

[12] S. Destercke, P. Buche, and V. Guillard. A flexible bipolar querying approach with imprecise data and guaranteed results. *Fuzzy sets and Systems*, 169:51–64, 2011.

[13] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77(2):321–357, 1995.

[14] A. J. García and G. R. Simari. Defeasible logic programming: An argumentative approach. *Theory and practice of logic programming*, 4:95–138, 2004.

[15] S. A. Gómez, C. I. Chesnevar, and G. R. Simari. An argumentative approach to reasoning with inconsistent ontologies. In 90, editor, *Knowledge Representation Ontology Workshop (KROW 2008). Conferences in Research and Practice in Information Technology.*, volume Thomas Meyer and Mehmet A. Orgun, 2008.

[16] L. Hu, C. Qiu, and Y. Shi. A defeasible description logic based semantic security policy conflict detection approach. *International Journal of Security and Its Applications*, 6(2):313–320, 2012.

[17] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[18] M. O. Moguillansky, N. D. Rotstein, and M. A. Falappa. A theoretical model to handle ontologydebugging & change through argumentation. In *Second International Workshop on Ontology Dynamics (IWOD-08)*, pages 29–42, 2008.

[19] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2011.

[20] I. Rahwan and G. Simari. *Argumentation in Artificial Intelligence.* Springer, 2009.

[21] G. Wang and Z. Lin. An argumentation framework for nonmonotonic reasoning in description logic. In *7th International Conference Rough Sets and Knowledge Technology*, pages 200–2006, 2012.

[22] K. Wang, D. Billington, J. Blee, and G. Antoniou. Combining description logic and defeasible logic for the semantic web. In *RuleML*, pages 170–181, 2004.

[23] M. Williams and A. Hunter. Harnessing ontologies for argument-based decision-making in breast cancer. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - Volume 02*, ICTAI '07, pages 254–261, Washington, DC, USA, 2007. IEEE Computer Society.

[24] Y. Wu. *Between argument and conclusion. Argument-based approaches to discussion. Inference and Uncertainty.* PhD thesis, Université du Luxembourg, 2012.