Modelling Negotiation Protocols in a Dialectical Framework

Alejandro G. Stankevicius* Alejandro J. García

Grupo de Investigación en Inteligencia Artificial (GIIA)
Departamento de Ciencias de la Computación
Universidad Nacional del Sur
Bahía Blanca - Buenos Aires - ARGENTINA
e-mail: {astank,ccgarcia}@criba.edu.ar

Abstract

It has been shown elsewhere that in order to overcome the limitations identified in monotonic reasoning we require a kind of *defeasible reasoning*. Moreover, *defeasible argumentation* is one of several approaches that address the problems faced when we attempt to reason defeasibly. Its main goal is to model the kind of reasoning that a *single* intelligent agent may need.

Despite of the fact that defeasible argumentation has been conceived for the single agent scenario, it is possible to consider certain argumentation process (the dialectical analysis) as a dispute or debate between *two parties*.

The two parties view can lead us to an attractive type of negotiation protocol which is based on a formal theory. In this paper, we exploit this approach by defining a dialectical framework capable of modelling a set of negotiation protocols of this kind. We also discuss how these protocols are affected by the design decisions taken inside the framework.

Keywords: DEFEASIBLE ARGUMENTATION, NEGOTIATION PROTOCOLS.

1 Motivations

It has been shown elsewhere [McC80, Rei80] that in order to overcome the limitations identified in monotonic reasoning we require a kind of *defeasible reasoning* (a form of non-monotonic reasoning). Moreover, *defeasible argumentation* [Pol87] is one of several approaches that address the problems faced when we attempt to reason defeasibly. In our group, we have conducted some research in this particular area: a sound and well grounded theory has been defined [SL92, SCG94], also a knowledge representation language based on that theory [SG95a], and a few implementations [Gar97, Sta99].

*Supported by a fellowship of the Secretaria General de Ciencia y Tecnología, UNS.

In these systems, the main goal is to model the kind of reasoning that a single intelligent agent may need. The knowledge of an agent is split in two disjoint sets $\mathcal{S}$ and $\mathcal{D}$, $\mathcal{S}$ containing strict knowledge (undisputed beyond any doubt) and $\mathcal{D}$ containing defeasible knowledge. The set $\mathcal{S}$ is required to be non contradictory. However, the set $\mathcal{D}$ is allowed to be contradictory. Briefly stated, the epistemic state of an agent is the set of justified assertions based on the knowledge base $\mathcal{S} \cup \mathcal{D}$.

In the single agent view, the process of justifying an assertion entails the construction of an argument for that assertion which should not be defeated. An argument is defeated when it has defeaters which in turn are not defeated. Hence, to decide the state of an argument a dialectical analysis is required. This analysis is structured by means of a tree of arguments (called dialectical tree) where the children of a certain node are all of its defeaters. Once the construction of the dialectical tree is finished, a simple marking of its nodes can decide the status of the root (the initial argument under consideration).

Inside the single agent approach, it is possible to consider the dialectical analysis as a dispute or debate between a *proponent* who try to justify the assertion being considered, and an *opponent* who try to defeat any supporting argument given by the proponent. Even though this seems to be an interesting idea (having in mind a potential negotiation protocol), this situation is the most constrained because both parties should have complete knowledge about each other, as they share the same $\mathcal{S}$ and the same $\mathcal{D}$. Of course, these constraints arise from the fact that both the proponent and the opponent are the same agent.

In a more general negotiation context, it is more likely to have several parties involved with different knowledge backgrounds. Naturally, the way to get rid of the aforementioned constraints is to apply the ideas from defeasible argumentation to more than one agent (recall that the goal of the original system was to model the reasoning of a *single* intelligent agent).

For these reasons, the two parties view can lead us to an attractive type of negotiation protocol which is based on a formal theory. In this paper, we exploit this approach by defining a dialectical framework capable of modelling a set of negotiation protocols of this kind. We also discuss how the protocols are affected by the design decisions taken inside the framework. The proposed framework was devised after considering the decisions implicitly made inside the protocol reported in [SG95b].

2 The framework

In this section we describe the proposed framework. It models a family of dispute protocols suitable for pairs of agents. In the first place, the architecture of the agents will be discussed. Next then we will try to isolate every aspect of the framework in which a decision should be taken, considering that each set of answers to those decisions defines a different protocol. To this purpose, the outcome of the alternative choices for each decision will also be analyzed.

2.1 Agent architecture

Every agent must use the *same* representation for its knowledge. In our framework we adopt the knowledge representation language defined in [SG95a]. As stated

before, the knowledge of an agent is split in two disjoint sets containing strict and defeasible knowledge, and this knowledge is represented using strict and defeasible rules. The strict rules are used to capture certain information (*e.g.*, "if X is a penguin, then X is a bird."). On the other hand, the defeasible rules are used to capture tentative information (*e.g.*, "if X is a bird, then typically X flies.")

Definition 2.1. We call *knowledge base* to the set $KB = \mathcal{S} \cup \mathcal{D}$, where $\mathcal{S}$ is a finite set of strict rules and $\mathcal{D}$ is a finite set of defeasible rules. ■

The knowledge base of each agent will be used for building *arguments*. In a debate, the arguments will be used by each agent for supporting its assertions or rebutting the assertions of its adversaries.

Definition 2.2. Let $KB = \mathcal{S} \cup \mathcal{D}$ be a knowledge base. An *argument* $\mathcal{A}$ for an assertion h is an instantiated subset of $\mathcal{D}$, such that:

- there exists a defeasible derivation for h from $\mathcal{S} \cup \mathcal{A}$,
- the set $\mathcal{S} \cup \mathcal{A}$ is non-contradictory, and
- $\mathcal{A}$ is minimal with respect to set inclusion (*i.e.*, there is no $\mathcal{A}' \subset \mathcal{A}$ such that $\mathcal{A}'$ satisfies the two previous conditions.) ■

Definition 2.3. The *architecture of an agent involved in the debate* consist of the pair $\langle KB, \mathcal{C} \rangle$, where KB is its knowledge base and $\mathcal{C}$ is a finite set of argument-comparison criteria[1] ■

Usually, when implementing these concepts the comparison criterion is fixed. For instance, *specificity* [Poo85] was used on many of them. For our purpose, this is unacceptable: the criteria diversity is another desired degree of freedom in our framework.

Definition 2.4. Consider the agents Ag_1 and Ag_2. We say that the argument $\mathcal{A}$ of Ag_1 *rebuts* the argument $\mathcal{B}$ of Ag_2 whenever $\mathcal{A}$ is either a *proper defeater* or a *blocking defeater* for $\mathcal{B}$ (we refer the interested reader to [SCG94] where the definition of defeater is given.) ■

2.2 Framework alternatives

We have described the common basis of the new framework. In order to complete its definition several decisions should be taken. Consider that each set of answers will define a valid variation of this framework which in turn models a specific negotiation protocol. In what follows, we will cover the framework's aspects of:

1. comparison criteria,
2. knowledge background, and
3. mutual trust.

[1] Any partial order defined over the set of arguments could be used as a valid comparison criterion.

2.2.1 Comparison criteria

Every agent can have a different set of comparison criteria. In a debate, the agents should agree somehow on which comparison criterion they are going to use. However, it is possible for those sets of comparison criteria to be disjoint (*i.e.*, there is no common criterion between them).

The agreement issue can be solved by imposing a default non-empty set of comparison criteria which should be understood by all the agents. The problematic case of having disjoint sets of comparison criteria is avoided because the agents always have at least one criterion in common.

Another solution could be to restrict the domain of application of the dispute protocol (see section 2.3) to pairs of agents $Ag_1 = (KB_1, \mathcal{C}_1)$ and $Ag_2 = (KB_2, \mathcal{C}_2)$ where $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$ (avoiding the tricky situation stated above). Further, this solution could also be improved to cope with the problematic situation: when no common criterion is found, a third agent acting as a *judge* (see section 2.2.3) performs the actual argument comparison. It goes without saying that the agents must rely on this judge, considering that it will clearly affect the outcome of the debate.

Finally, another possibility is to define a kind of setup protocol which should find the best comparison criterion available to both agents. This is the least explicit alternative: the proper course of action when there is no common criterion remains unspecified. Nevertheless, this situation is assumed to be solved by this setup protocol.

The alternatives discussed above can be summarized as:

a) A non-empty set of default comparison criteria is given.

b) A non-empty intersection is required on the agents' sets of comparison criteria.

c) A judge is designated to mediate in the dispute.

d) A kind of setup protocol is settled.

2.2.2 Knowledge background

Although no restriction was imposed on the KB of each agent there is a particular situation to consider. If we impose no restrictions on the KB, it will be possible to build a valid argument for some agent Ag_1 (with respect to Definition 2.2) which is invalid for another agent Ag_2.

This pitfall can be averted in several ways: for instance, every agent can share the same $\mathcal{S}$. In this setting, the depicted uncertain situation is impossible. Unfortunately, as we stated before, this highly constrained situation is undesired. A somewhat weaker requirement that still avoids that problem is to restrict the domain of application of the dispute protocol to agents whose strict knowledge is non-contradictory (*i.e.*, $\mathcal{S}_1 \cup \mathcal{S}_2 \nvdash \bot$.)

Furthermore, a more general solution is to impose no restriction on the set $\mathcal{S}$ itself, but each agent propose a subset $\mathcal{S}'$, $\mathcal{S}' \subseteq \mathcal{S}$, which satisfies the previous condition (*i.e.*, $\mathcal{S}'_1 \cup \mathcal{S}'_2 \nvdash \bot$, for some $\mathcal{S}'_1 \subseteq \mathcal{S}_1$ and $\mathcal{S}'_2 \subseteq \mathcal{S}_2$)

To conclude, the alternatives identified in order to take care of this pitfall can be summarized as:

a) All the agents share the same strict knowledge.

b) The union of the strict knowledge of the agents involved in the dispute is non-contradictory.

c) Only conflict-free subsets of the agents' strict knowledge is considered throughout the negotiation.

2.2.3 Mutual trust

Almost every negotiation protocol defined in the literature make a strong assumption: they take for granted that the agents *behave well* in the sense that they do not lie, they do not threaten each other, etc. In our framework, this assumption underpins the confidence that an agent has on the validity of the arguments introduced by its adversary. Bear in mind that non-minimal or inconsistent arguments are disallowed, as stated in Definition 2.2.

Naturally, someone will eventually be in charge of the argument validation. Considering this, we can obtain several flavors in the confidence between parties just by modifying who will be charge of this validation.

In the naive approach (*total trust*) each agent is in charge of checking the validity of its own arguments and of avoiding fallacious reasoning (see Definition 2.7). Of course, in the total trust approach it is impossible to drop that assumption (the whole approach relies on it). However, there are domains where that assumption is harmless and the naive approach can be successfully applied.

Another alternative, which requires only a weaker form of that assumption, is to assign to each agent the responsibility of checking the arguments of its opponent and also verifying that fallacious reasoning is avoided. In this approach, note that the strict knowledge of each agent should be made available to its adversary.

If the sharing of knowledge between adversaries is not an option, we can designate a third agent as the *judge* of the dispute. This judge performs the validity check and the fallacy control of every argument introduced throughout the debate. To accomplish these tasks, the judge should have access to the strict knowledge of both agents. Clearly, trust among the conflicting parties is not required and for this reason the assumption is not needed.

These alternatives ensures the validity of arguments appearing in a debate. There is another way of dropping the assumption without resorting to a third agent: each agent can check the acceptability of the arguments from both agents against its own knowledge. However, the validity of arguments is no longer ensured in this new setting because it is possible for an agent to build an invalid argument that could be accepted by its adversary. After all, this approach is resembling real world discussions.

Finally, the different shades of mutual trust reviewed can be summarized as:

a) Each agent checks its own arguments (total trust).

b) Each agent checks the arguments of its adversary.

c) There is a judge which checks every argument (no trust required).

d) Each agent checks the arguments of its adversary against its own knowledge.

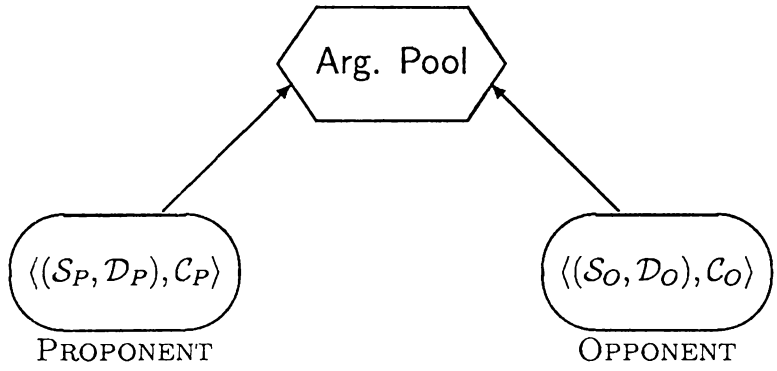

Figure 1: Scheme of the proposed framework.

2.3 Dispute protocol

We have stated before that every variant of the framework is associated with a dispute protocol. Having discussed all the alternatives, we are ready to introduce that protocol.

Definition 2.5. Let $Ag_1 = (KB_1, \mathcal{C}_1)$ and $Ag_2 = (KB_2, \mathcal{C}_2)$ be two agents, such that the restrictions of the alternatives selected are observed (see the previous discussion). A *debate* between agents Ag_1 and Ag_2 follows this scheme:

1. The proponent starts with an argument that supports the assertion being disputed. The turn goes to the opponent.
2. The opponent either rebuts an argument and the turn goes to the proponent or accepts the assertion being disputed.
3. The proponent either rebuts an argument and the turn goes to the opponent or accepts the assertion being disputed.

Accordingly, a debate can be understood as a succession of argument introduced by either the proponent (on odd turns) or the opponent (on even turns). ■

The next definition characterizes the notion of *argument pool*, the structure where the arguments introduced along the debate are stored.

Definition 2.6. An *argument pool* is a set of sequences composed by pairs $(Ag, \mathcal{A})$. Each pair contains an argument $(\mathcal{A})$ and the name of the agent which introduced that argument (Ag). ■

Note that as the dispute progress, the argumentation pool contains the (partial) argumentation lines being explored. The Figure 1 depicts an scheme of the main components of our framework.

Finally, in order to avoid the so-called fallacious reasoning [SCG94, GSC98] we need to impose some restriction on the arguments that are allowed to be introduced on a given stage of the debate.

Definition 2.7. Let $\mathcal{B}$ be an argument of the proponent (agent Ag_1) and let $\mathcal{C}$ be an argument of the opponent (agent Ag_2). The argument $\mathcal{B}$ can be used to rebut the argument $\mathcal{C}$ if and only if the sequence $[(Ag_1, \mathcal{A}_1), (Ag_2, \mathcal{A}_2), \ldots, (Ag_1, \mathcal{A}_n), (Ag_2, \mathcal{C})]$ appears as prefix of at least one sequence in the pool of arguments, and the following conditions are meet:

- the sequence $[(Ag_1, \mathcal{A}_1), \ldots, (Ag_1, \mathcal{A}_n), (Ag_2, \mathcal{C}), (Ag_1, \mathcal{B})]$ does not appear as prefix of any sequence already present in the argument pool,
- in the sequence $[(Ag_1, \mathcal{A}_1), \ldots, (Ag_1, \mathcal{A}_n), (Ag_2, \mathcal{C}), (Ag_1, \mathcal{B})]$, all the arguments introduced by the same agent are non-contradictory, and
- the argument $\mathcal{C}$ is not a subargument of any argument proposed by the agent Ag_2 in the sequence $[(Ag_1, \mathcal{A}_1), . \quad , (Ag_1, \mathcal{A}_n), (Ag_2, \mathcal{B})]$.

When these conditions are meet, $[(Ag_1, \mathcal{A}_1), \quad , (Ag_1, \mathcal{A}_n), (Ag_2, \mathcal{C}), (Ag_1, \mathcal{B})]$ can be added to the pool of arguments denoting that the argument $\mathcal{C}$ has been rebutted by the argument $\mathcal{B}$. The case where Ag_1 is the opponent and Ag_2 is the proponent is analogous. ■

Having said that, it is easy to see that the proponent wins the debate if and only if after the dispute there is at least one sequence of odd length in the argument pool (*i.e.* there exists an undefeated argumentation line).

2.4 Modelling a protocol

We have defined a dialectical framework for modelling negotiation protocols. The choices made in the aspects discussed in section 2.2 characterizes different protocols. We would like to show as an example how this dialectical framework describes an actual protocol that was reported in [SG95b].

This protocol is modeled by choosing the second alternative in the comparison criteria discussion, that is, the agents must always have a common comparison criterion. For the knowledge background the alternative selected should be the first, where the agents share the same strict knowledge. This protocol makes the assumption that the agents can trust each other completely (*i.e.*, the first alternative in the mutual trust discussion).

3 Conclusions

In this paper we have proposed a dialectical framework capable of modelling a family of negotiation protocols. These negotiation protocols have the common property of being based on a formal theory. All these protocols were obtained combining the alternatives identified for each relevant aspect of the framework (*e.g.*, comparison criteria, knowledge background, etc.). At the present state of affairs, we avoided any kind of comparison between protocols.

Despite of this, we feel that this line of research should be deepened, bearing in mind that many of the promises made in the multi-agent systems community relies on the development of a good negotiation framework.

References

[Gar97] GARCÍA, A. J., *"La Programación en Lógica Rebatible: su definición teórica y computacional"*, MSc Thesis, Universidad Nacional del Sur, Bahía Blanca, 1997.

[GS99] García, A. J., and Simari, G. R., *"Strong and default negation in defeasible logic programming"*, in proceedings, Fourth Dutch-German Workshop on Nonmonotonic Reasoning Techniques and their Application (DGNMR), 1999.

[GSC98] García, A. J., Simari, G. R., and Chesñevar, C. I., *"An Argumentative Framework for Reasoning with Inconsistent and Incomplete Information"*, in proceedings, Workshop on Practical Reasoning and Rationality, 13th biennial European Conference on Artificial Intelligence (ECAI), 13–19, 1998.

[McC80] McCarthy, J., *'Circumscription – A Form of Non-monotonic Reasoning"*, Artificial Intelligence, **13**:27–39, 1980.

[Pol87] Pollock, J. L., *"Defeasible Reasoning"*, Cognitive Science, **11**:481–518, 1987.

[Poo85] Poole, D. L., *"On the Comparison of Theories: Preferring the Most Specific Explanation"*, in proceedings, Ninth International Joint Conference on Artificial Intelligence (IJCAI), 144–147, 1985.

[Rei80] Reiter, R., *"A Logic for Default Reasoning"*, Artificial Intelligence, **13**:81–132, 1980.

[SCG94] Simari, G. R., Chesñevar, C. I., and García, A. J., *"The Role of Dialectics in Defeasible Argumentation"*, in proceedings, XIV Conferencia Internacional del la Sociedad Chilena para Ciencias de la Computación, Chile, 1994.

[SG95a] Simari, G. R., and García, A. J., *"A Knowledge Representation Language for Defeasible Argumentation"*, in proceedings, XXI Conferencia Latinoamericana de Informática, Brasil, 1995.

[SG95b] Simari, G. R., and García, A. J., *"Protocolos para arbitrar debates entre agentes inteligentes"*, XXIV Jornadas Argentinas de Informática e Investigación Operativa, 1995.

[SL92] Simari, G. R., and Loui, R. P., *"A Mathematical Treatment of Defeasible Reasoning and its Implementation"*, Artificial Intelligence, **53**:125–157, 1992.

[Sta99] Stankevicius, A. G., *"Visualización e Interpretación de Programas Lógicos Rebatibles"*, BSc Thesis, Universidad Nacional del Sur, Bahía Blanca, 1999.

UN MODELO PARA EL TRATAMIENTO DE SISTEMAS DINÁMICOS BASADO EN LA SATISFACCIÓN DE RESTRICCIONES

R. Forradellas[1], F. Ibañez,[1] R. Berlanga[2]

RESÚMEN

En numerosas aplicaciones industriales complejas de plaificación y scheduling, resulta frecuente encontrar casos donde un problema ya resuelto debe ser reconsiderado a causa de una ligera modificación en la instancia de dicho problema. Estas modificaciones se originan generalmente a partir de sucesos externos que implican un cambio de creencias y en consecuencia el conjunto de soluciones obtenido para el problema resuelto ha de modificarse.

Estos casos son referidos generalmente como *problemas dinámicos*, frente a los problemas *estáticos*. En los primeros, el conjunto de soluciones puede ser ligeramente modificado, mientras que en los segundos, el conjunto de soluciones es fijo e inalterable.

El tipo de problemas que nos preocupa se refieren a problemas modelados a través de restricciones, concretamente, restricciones lineales sobre variables de dominio finito. Estos tipos de problemas son estáticos, cuando las soluciones obtenidas no son reconsideradas ante el cambio de la instancia del problema. Los casos dinámicos antes expuestos son resueltos iniciando de nuevo el proceso de resolución con la instancia modificada como si fuese un problema diferente.

Un resolvedor de problemas que reconsidere las soluciones obtenidas en un problema anterior ante un cambio ligero de su instancia lo denominaremos dinámico, frente a la denominación de estático antes utilizada. Así pues, un Sistema Dinámico de Restricciones (SDR) será aquel que considere las soluciones obtenidas para resolver la instancia modificada. Al contrario de los sistemas estáticos, un SDR plantea las modificaciones de las instancias como un único problema.

En este trabajo definiremos un modelo de SDR e identificaremos el tipo de transiciones permitidas en el mismo, y discutiremos como abordar la resolución dinámica del SDR desde diferentes aproximaciones. Además, se propondrán varios métodos para el manejo dinámico de un sistema de restricciones. Finalmente, discutiremos brevemente que opción de las analizadas es la más adecuada para los problemas que estamos abordando.

1.- MODELO DEL SISTEMA DINÁMICO CON RESTRICCIONES –SDR-

En general el costo de la búsqueda de una solución o todas las soluciones, en un Sistema con Restricciones, requiere de una cantidad de tiempo bastante elevada, más aún, impredecible. En muchos casos el problema se acentúa cuando, además, se imponen restricciones de optimización sobre el Sistema, como sucede en las aplicaciones de planificación industrial.

Por otro lado, es muy frecuente encontrar en aplicaciones de planificación, en los que, una vez ya representado y resuelto el problema basado en restricciones, algunos valores de variables o de restricciones del mismo problema

---

[1]LISI/IdeI – UNSJ {*kike, fibanez*}*iinfo.unsj.edu.ar*

[2]Dpto. Informática – UJI berlanga@inf.uji.es

Parte de las investigaciones presentadas en este trabajo fueron realizadas en el marco del Proyecto CICITCA (UNSJ) "*Resolución de Problemas de Asignación de Recursos aplicando tecnicas de Inteligencia Artificial*"