GDL as a Unifying Domain Description Language for Declarative Automated Negotiation

JAAMAS Track

Dave de Jonge
IIIA-CSIC
Barcelona, Spain
davedejonge@iiia.csic.es

Dongmo Zhang
Western Sydney University
Sydney, Australia
d.zhang@westernsydney.edu.au

ABSTRACT

We show that Game Description Language (GDL) can be used to describe some of the most commonly used test-beds in the automated negotiations literature, namely Genius and Colored Trails. This opens up an entirely new, *declarative*, approach to automated negotiation, in which a single algorithm can negotiate over a very broad class of different negotiation domains. We formally prove that the set of possible agreements of any negotiation domain from Genius (either linear or non-linear) can be modeled as a set of strategies over a deterministic extensive-form game that can be described efficiently in GDL. Furthermore, we show experimentally that, given only this GDL description, we can explore the agreement space efficiently using entirely generic domain-independent algorithms. In addition, we show that the same also holds for negotiation domains in the Colored Trails framework. This means we have the basic ingredients to implement a single negotiating agent that is capable of negotiating over many different kinds of negotiation domains, including Genius and Colored Trails.

KEYWORDS

Automated Negotiation; Game Description Language; Non-zero-sum Games; Extensive-form Games; General Game Playing; Monte Carlo Tree Search

ACM Reference Format:
Dave de Jonge and Dongmo Zhang. 2022. GDL as a Unifying Domain Description Language for Declarative Automated Negotiation: JAAMAS Track. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

One of the main goals of Artificial Intelligence is to develop *general intelligence*. That is, to develop algorithms that are capable of reasoning about virtually any kind of problem, rather than just one specific problem. For this reason, the field of General Game Playing (GGP) [3] studies algorithms that are able to play games of which the rules are only known at run-time. Such a 'declarative' approach requires a machine-readable language to express the rules of many different games. For this purpose the Game Description Language (GDL) [12] was developed.

Recently, we have proposed to apply the declarative approach also to the field of automated negotiation [6], using GDL to specify negotiation domains. Our idea was that two agents would be playing a non-zero-sum game described in GDL, while having the opportunity to negotiate binding agreements with each other about which moves each would make. This idea was further developed in [7] and [8], which introduced an algorithm for declarative automated negotiation, called Monte Carlo Negotiation Search (MCNS). It was based on Monte Carlo Tree Search (MCTS) [10], but extended with a negotiation algorithm that allowed the player to negotiate with its opponent. MCNS is entirely generic in the sense that it can be applied to any game expressible in GDL, and this has been tested on three different games. Although this work did shed light on the feasibility of the declarative approach for automated negotiation, it did not demonstrate to what extent it is applicable to more traditional negotiation domains that do not involve games. Therefore, in this paper we take this idea a step further and show that indeed it can also be applied to two of the most commonly used test-beds in the field of automated negotiation, namely Genius [11] and Colored Trails [2, 4]. Although the domains in Genius are not directly related to games, we show that they can nevertheless be described efficiently in GDL.

We present the following results:

- We classify negotiation domains into two broad classes, namely *Cartesian domains* and *strategic domains*.
- We define two notions of equivalence (*isomorphism* and *weak isomorphism*) between negotiation domains.
- We formally prove that every Cartesian domain (which includes all domains in Genius) is weakly isomorphic to some strategic negotiation domain.
- We show that, thanks to this weak isomorphism, any domain from the Genius framework (either linear or non-linear) can be described efficiently in GDL.
- We show that the Colored Trails domains can also be described efficiently in GDL.
- We experimentally show that the the Genius domains and the Colored Trails domains can be explored efficiently by completely generic algorithms that only take GDL descriptions as their input.

For more details we refer to the full version of this paper [9].

2 FORMAL RESULTS

Definition 2.1. A (bilateral) **Negotiation Domain** is a tuple $\langle \Omega, c, U_1, U_2 \rangle$ where Ω is the set of **deals**, c is the **conflict outcome**, U_1 and U_2 are two **utility functions** (one for each agent) which are maps from $\Omega \cup \{c\}$ to $\mathbb{R}$. The utility values $U_i(c)$ are called the **reservation values**.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

We distinguish between two types of negotiation domains, namely *Cartesian* domains and *strategic* domains. All domains in the Genius framework are Cartesian, while Colored Trails and the game of Diplomacy [5] are examples of strategic domains.

Definition 2.2. A negotiation domain $\langle \Omega, c, U_1, U_2 \rangle$ is called a **Cartesian negotiation domain** if its agreement space Ω is the Cartesian product of a finite number of finite sets $\Omega = I_1 \times I_2 \cdots \times I_n$, which are called the **issues** of the domain. A **linear negotiation domain** is a Cartesian negotiation domain for which its utility functions U_i satisfy: $U_i(a_1, a_2, \ldots a_n) = \sum_{j=1}^{n} d_{i,j}(a_j)$ where each a_j is an element of I_j and each $d_{i,j}$ is a map from I_j to $\mathbb{R}$.

The idea of a strategic negotiation domain, is that it represents a negotiation between two players that are playing some game G, but that may negotiate which joint strategies they will play. Of course, this only makes sense if the game is a non-zero-sum game, so that the players may indeed mutually benefit from making binding agreements.

Definition 2.3. Let G be an extensive-form game. Then a **strategic negotiation domain** over G is a negotiation domain where Ω is some set of joint strategies of G, and for each $\omega \in \Omega$ the utility values $U_i(\omega)$ are given by the utility values of G that the players would obtain if they played according to the joint strategy ω.

A *branch negotiation domain* is a special case of a strategic negotiation domain, in which the joint strategies that the players may agree upon completely fix every single move from the beginning of the game until the end. So, any agreement between the players completely determines the outcome of the game.

Definition 2.4. For any extensive-form game G its corresponding **branch negotiation domain** $\mathcal{B}(G)$ is the strategic negotiation domain over G for which the possible agreements are exactly the *branches* of G (sequences of moves that start at the initial state of G and end at some terminal state of G) .

Definition 2.5. Two negotiation domains are **weakly isomorphic** if there exists a one-to-one mapping between all the individually rational outcomes of both domains that preserves the utility values (up to a linear transformation).

We can now state our main theorems. Their proofs can be found in the full version of this paper [9].

Theorem 2.6. *For any Cartesian negotiation domain C there exists a game G_C such that $\mathcal{B}(G_C)$ is weakly isomorphic to C.*

Theorem 2.6 is important because it says that any Cartesian negotiation domain C can essentially be described by a GDL description of the game G_C.

Theorem 2.7. *Let C be a linear negotiation domain with n issues, and for which the largest issue has size m. Then, a description of C in the format of Genius can be converted to a GDL description of G_C in $O(mn)$ time and this GDL description will have $O(mn)$ size.*

Theorem 2.8. *Let C be a non-linear negotiation domain from the Genius framework with n issues, and with k constraints. Then, a description of C in the format of Genius can be converted to a GDL description of G_C in $O(nk)$ time and this GDL description will have $O(nk)$ size.*

Theorem 2.9. *Suppose we have an instance of Colored Trails with a grid of size $m \times m$, and with c colors. Then it can be described in GDL with a description of size $O(m^2 + c)$.*

3 EXPERIMENTAL RESULTS

The theorems above show that it is possible to efficiently generate GDL descriptions of Genius and Colored Trails instances. The next question we aim to answer, is whether those GDL descriptions can also be *parsed* efficiently by a negotiation algorithm. Specifically, we have performed a number of experiments to answer the following two questions: Given only the GDL description of a game G, *how quickly can an agent detect its reservation value for the negotiation domain $\mathcal{B}(G)$?* and *how quickly can an agent discover and evaluate the possible agreements in $\mathcal{B}(G)$?*

Detecting the reservation values of $\mathcal{B}(G)$ amounts to determining the subgame-perfect equilibrium of G. To do this, we implemented a Score-Bounded MCTS algorithm [1]. To answer the second question we implemented a simple depth-first search algorithm that iterates over all branches of G. We repeated these experiments for three types of domains: linear Genius domains, non-linear Genius domains, and Colored Trails domains.

We have converted each of the 24 linear Genius domains that were used for ANAC 2012 into GDL, and we observed that *in all cases the reservation value could be found in less than a millisecond*, and that *even in the largest domain (the Energy domain with 390,625 possible deals) we were able to find and evaluate all possible deals in just over 3 seconds.*

We have repeated these experiments with the non-linear domains used for ANAC 2014. However, since the sizes were too large for exhaustive exploration (between 10^{10} and 10^{50} deals) we measured the time required to evaluate 1 million deals. We observed that *in all cases the reservation value could be found in less than 10 milliseconds*, and that *in most cases it took between 15 and 60 seconds to evaluate a million deals.* Furthermore, we observed that this time increases with the number of issues in the domain.

Finally, we performed the same experiments on the Colored Trails game. We randomly generated 30 instances of this game, each consisting of a 6×6, 7×7, or 8×8 grid, with 4 different colors, and each player having between 7 and 10 chips. The maximum number of rounds was set to 40. The initial squares of the two players were set at the top-left and bottom-right squares respectively, while their goal squares were set at the center of the grid. We observed that *finding the reservation value took between 3 and 11 milliseconds*, with very large variance between the various instances, which does not seem to have any correlation with the size of the instances. Furthermore, we observed that *it took between 40 and 46 seconds to evaluate a million deals.*

ACKNOWLEDGMENTS

This work was funded by the E.U. Horizon 2020 Research and Innovation Programme project LOGISTAR (Grant Agreement No. 769142), and by a Juan de la Cierva research grant from the Spanish Ministry of Science and Innovation (IJC2018-036443-I).

REFERENCES

[1] Tristan Cazenave and Abdallah Saffidine. 2010. Score Bounded Monte-Carlo Tree Search. In *Computers and Games (Lecture Notes in Computer Science, Vol. 6515)*. Springer, 93–104.

[2] Y. Gal, B. Grosz, S. Kraus, A. Pfeffer, and S. Shieber. 2010. Agent Decision-Making in Open-Mixed Networks. *Artificial Intelligence* (2010). http://dx.doi.org/10.1016/j.artint.2010.09.002

[3] M. Genesereth, N. Love, and B. Pell. 2005. General game playing: Overview of the AAAI competition. *AI Magazine* 26(2) (2005), 62–72.

[4] Barbara J. Grosz, Sarit Kraus, Shavit Talman, Boaz Stossel, and Moti Havlin. 2004. The Influence of Social Dependencies on Decision-Making: Initial Investigations with a New Game. In *3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 19-23 August 2004, New York, NY, USA*. IEEE Computer Society, 782–789. https://doi.org/10.1109/AAMAS.2004.10097

[5] Dave de Jonge and Carles Sierra. 2017. D-Brane: a Diplomacy Playing Agent for Automated Negotiations Research. *Applied Intelligence* 47, 1 (2017), 158–177. https://doi.org/10.1007/s10489-017-0919-y

[6] Dave de Jonge and Dongmo Zhang. 2016. Using GDL to Represent Domain Knowledge for Automated Negotiations. In *Autonomous Agents and Multiagent Systems: AAMAS 2016 Workshops, Visionary Papers, Singapore, Singapore, May 9-10, 2016, Revised Selected Papers*, Nardine Osman and Carles Sierra (Eds.). Springer International Publishing, Cham, 134–153.

[7] Dave de Jonge and Dongmo Zhang. 2017. Automated Negotiations for General Game Playing. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund Durfee (Eds.). ACM, 371–379.

[8] Dave de Jonge and Dongmo Zhang. 2020. Strategic negotiations for extensive-form games. *Autonomous Agents and Multi-Agent Systems* 34, 1 (Apr 2020). https://doi.org/10.1007/s10458-019-09424-y

[9] Dave de Jonge and Dongmo Zhang. 2021. GDL as a unifying domain description language for declarative automated negotiation. *Auton. Agents Multi Agent Syst.* 35, 1 (2021), 13. https://doi.org/10.1007/s10458-020-09491-6

[10] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-carlo Planning. In *Proceedings of the 17th European Conference on Machine Learning* (Berlin, Germany) *(ECML'06)*. Springer-Verlag, Berlin, Heidelberg, 282–293. https://doi.org/10.1007/11871842_29

[11] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. 2014. Genius: An Integrated Environment for Supporting the Design of Generic Automated Negotiators. *Computational Intelligence* 30, 1 (2014), 48–70. https://doi.org/10.1111/j.1467-8640.2012.00463.x

[12] Nathaniel Love, Michael Genesereth, and Timothy Hinrichs. 2006. *General Game Playing: Game Description Language Specification.* Technical Report LG-2006-01. Stanford University, Stanford, CA. http://logic.stanford.edu/reports/LG-2006-01.pdf.